PENINGKATAN PERFORMA DAN PENERAPAN KINERJA ANDROID UNTUK DETEKSI MALWARE MENGGUNAKAN ALGORITMA RANDOM FOREST

SKRIPSI SARJANA INFORMATIKA

Oleh

Muhammad Rauzan Fadhila

217064516085

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS NASIONAL

2024

PENINGKATAN PERFORMA DAN PENERAPAN KINERJA ANDROID UNTUK DETEKSI MALWARE MENGGUNAKAN ALGORITMA RANDOM FOREST

SKRIPSI SARJANA

Karya ilmiah sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer dari Fakultas Teknologi Komunikasi dan Informatika

Oleh

Muhammad Rauzan Fadhila

217064516085

PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS NASIONAL

2024

KATA PENGANTAR

Puji Syukur penulis panjatkan atas kehadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul "PEN PENINGKATAN PERFORMA DAN PENERAPAN KINERJA ANDROID UNTUK DETEKSI MALWARE MENGGUNAKAN ALGORITMA RANDOM FOREST". Adapun tujuan dari penyusunan skripsi ini ialah untuk memenuhi salah satu syarat memperoleh gelar Sarjana Komputer di Fakultas Teknologi Komunikasi dan Informatika Universitas Nasional Jakarta.

Dalam penyusunan skripsi ini penulis banyak menerima bantuan, bimbingan serta dukungan dari berbagai pihak, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Maka dalam kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu dalam penyusunan skripsi ini, terutama kepada yang terhormat:

1. Bapak Dr. Drs. El Amry Bermawi Putera, M.A. Selaku Rektor Universitas Nasional
2. Bapak Dr. Agung Triayudi, S.Kom.,M.Kom Selaku Dekan Fakultas Teknologi Komunikasi dan Informatika Universitas Nasional
3. Ibu Ir. Endah Tri Esti Handayani, MMSI Selaku Wakil Dekan Fakultas Teknologi Komunikasi dan Informatika Universitas Nasional
4. Ibu Ratih Titi Komalasari, ST.,MM., MMSI Selaku Ketua Program Studi Informatika Universitas Nasional
5. Ibu Dr. Andrianingsih, S.Kom.,MMSI.. Selaku Dosen Pembimbing yang telah mengorbankan waktu, pikiran dan tenaga untuk membimbing serta memberikan saran dalam menyelesaikan skripsi.
6. Ibu Rima Tamara Aldisa, S.Kom., M.Kom. Selaku Sekretaris Program Studi Informatika Universitas Nasional

7. Para Dosen dan Seluruh Staff akademik Fakultas Teknologi Komunikasi dan Informatika Universitas Nasional yang telah memberikan bekal ilmu yang bermanfaat
8. Para Pimpinan Instansi beserta Mentor yang telah mengizinkan saya untuk melakukan kegiatan penelitian dan memberikan ilmu yang bermanfaat
9. Kedua orangtua tercinta dan saya sayangi yang senantiasa mencurahkan segenap kasih sayang yang tiada henti-hentinya, doa, motivasi, nasehat, serta kesabaran yang begitu besar.

Penulis mengakui bahwa skripsi ini memiliki kekurangan dalam berbagai aspek, termasuk materi, isi, dan teknik penyajian. Hal ini disebabkan oleh keterbatasan pengetahuan dan kemampuan penulis. Oleh karena itu, penulis sangat menghargai kritik dan saran yang bersifat membangun dari semua pihak, dan berharap agar kontribusi tersebut dapat meningkatkan kualitas skripsi ini ke arah yang lebih baik.

Jakarta, XX XX XXXX

Muhammad Rauzan Fadhila

ABSTRAK

DAFTAR ISI

DAFTAR GAMBAR

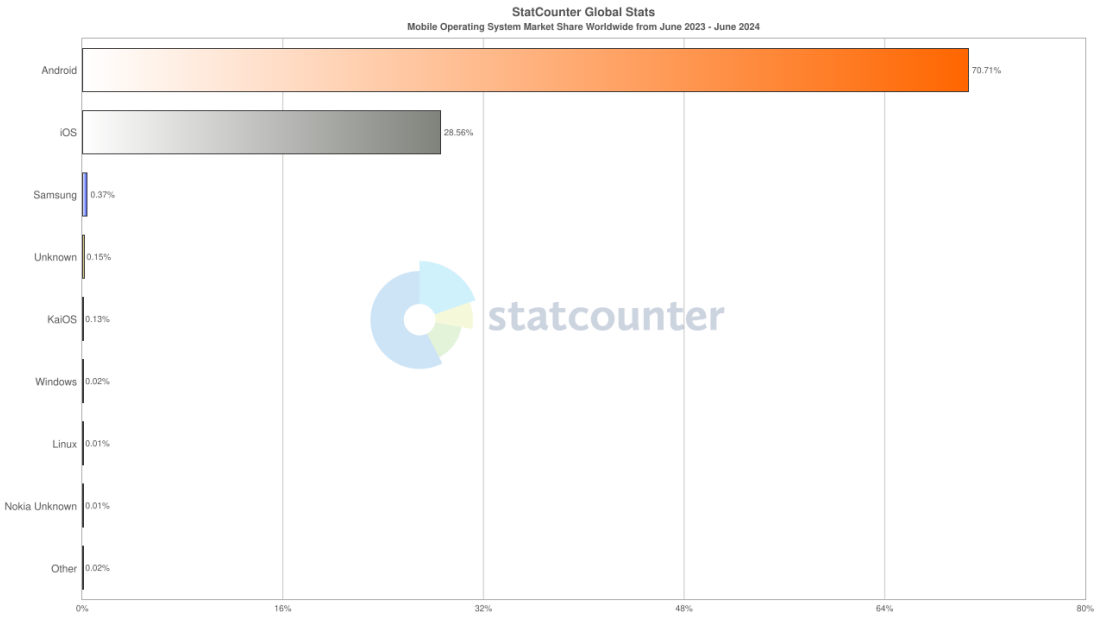
DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Penggunaan perangkat mobile berbasis Android terus meningkat pesat, menjadikannya target utama bagi berbagai jenis serangan siber, khususnya malware. Menurut analisis Statcounter(2024)[9], Android menguasai lebih dari 70% pangsa pasar perangkat seluler global, sehingga rentan terhadap ancaman malware yang disebarkan melalui aplikasi berbahaya seperti: Trojan, Ransomware, dan Spyware. Serangan malware pada perangkat Android berpotensi membahayakan privasi pengguna, memberikan akses tidak sah ke perangkat, dan mencuri data pribadi. Strategi baru yang lebih mudah beradaptasi diperlukan karena sistem deteksi malware tradisional yang mengandalkan teknik deteksi berbasis tanda tangan mengalami kesulitan dalam mengidentifikasi serangan baru atau *Zero-Day.* (Chitayae et al., 2023)[4]



Gambar 1.1 *Market share held by mobile operating systems in 2023-2024*

Oleh karena itu, penggunaan *Machine learning* dapat mengenali pola aktivitas malware dan membedakannya dari aplikasi yang aman, algoritma ini sangat menjanjikan untuk memecahkan masalah mengenai pola aktivitas malware(Thorat et al., 2024). Salah satu algoritma yang dapat digunakan yaitu Random Forest, yang dapat mendeteksi malware dengan akurasi tinggi dan dapat mengelola data yang sangat besar dan kompleks(Rafrastara et al., 2023). Dalam upaya untuk meningkatkan ketepatan dan efektivitas sistem pendeteksi malware, penelitian ini berfokus pada penerapan dan peningkatan kinerja algoritma Random Forest dalam mengidentifikasi *malware* pada Android. Informasi yang digunakan dalam penelitian ini dikumpulkan dari berbagai Android yang `aman dan bebas malware dan tersedia sebagai dataset terbuka pada Android Permision Dataset. Dengan optimalisasi algoritma ini, diharapkan sistem deteksi malware dapat memberikan perlindungan lebih baik bagi pengguna Android tanpa mengorbankan kinerja perangkat.

Pada salah satu riset yang pernah dilakukan sebelumnya, Penelitian ini menggunakan analisis statis terhadap file Windows Portable Executable (PE) untuk menilai seberapa baik algoritma Random Forest dan Support Vector Machine (SVM) dalam mendeteksi malware. Algoritma Random Forest menunjukkan akurasi yang mengesankan sebesar 98,53%, sementara SVM mencatat akurasi sedikit lebih rendah yaitu 97,14%, menurut studi yang menggunakan dataset file PE terkait malware dan file aman. Berdasarkan temuan ini, Random Forest merupakan pilihan yang lebih baik untuk deteksi malware dalam konteks penelitian ini karena lebih berhasil dalam mengidentifikasi file PE sebagai aman atau jahat. (Ismail et al., 2024).

Pada penelitian sebelumnya juga menunjukkan bahwa setelah menggunakan pendekatan feature selection, algoritma K-Nearest Neighbor (KNN) mampu mengklasifikasikan malware dan program jinak pada perangkat Android dengan akurasi 77%. Dengan tidak adanya seleksi fitur, akurasi yang dicapai hanya 44%. Studi ini juga menemukan bahwa 80% data harus digunakan untuk pelatihan dan 20% untuk pengujian dalam hal pelatihan dan pengujian. Selain itu, penelitian ini menyoroti pentingnya preprocessing dataset untuk meningkatkan kinerja model dan menyarankan

investigasi tambahan untuk mengkategorikan malware selain aplikasi Android. (Chitayae et al., 2023).

Terdapat sejumlah permasalahan yang signifikan dalam penerapan machine learning untuk mengidentifikasi Malware pada Android, pada penelitian "Peningkatan Performa dan Penerapan Algoritma Random Forest untuk Deteksi Malware di Android." Salah satu tantangan terbesar adalah kinerja algoritma di lingkungan perangkat mobile yang terbatas, seperti CPU, memori, dan daya baterai. Meskipun algoritma Random Forest memiliki kemampuan klasifikasi yang kuat, penerapannya dalam mendeteksi malware secara real-time di Android dapat menyebabkan penurunan performa akibat keterbatasan sumber daya tersebut.(Lakshmanarao & Shashi, 2022)

keberagaman dan kualitas data malware juga merupakan masalah yang signifikan. Algoritma mengalami kesulitan untuk mengidentifikasi keberagaman malware baru karena sejumlah besar dataset yang digunakan untuk pelatihan model tidak memiliki keterwakilan yang memadai. Overfitting, yaitu ketika model cocok dengan data pelatihan dengan sangat baik sehingga kinerjanya menurun Ketika dihadapkan dengan data baru. Teknik Random Forest sering menghadapi masalah ini karena mereka memiliki sejumlah besar parameter yang perlu diatur secara optimal. Waktu pendeteksian yang lambat adalah salah satu masalah tambahan. Deteksi malware yang cepat terhambat oleh waktu komputasi Random Forest yang relatif lebih lama karena banyaknya pohon keputusan yang harus diproses. Selain itu, model ini mengalami kesulitan dalam mengidentifikasi malware yang menyembunyikan aktivitas berbahaya karena taktik penyamaran yang digunakan oleh pembuatnya.

Sebagai solusi, penelitian ini akan mengurangi jumlah fitur yang tidak berguna dan memangkas pohon keputusan untuk mempercepat proses prediksi, membuat algoritma Random Forest lebih efektif di lingkungan Android. Selain itu, malware yang menggunakan taktik penyamaran dapat ditemukan dengan menggunakan pendekatan ekstraksi fitur berbasis perilaku. Metode ini berkonsentrasi pada pemeriksaan perilaku aplikasi setelah instalasi, termasuk interaksi dengan API dan pola izin. Validasi K-Fold akan digunakan untuk mengurangi overfitting dan memastikan model berkinerja baik

pada dataset malicious website yang lebih besar dan beragam. Disarankan juga agar batch processing atau deteksi berbasis cloud digunakan untuk mempersingkat waktu deteksi ketika analisis ekstensif dilakukan di server dengan sumber daya yang lebih besar. Diharapkan metode ini akan sangat meningkatkan kecepatan, akurasi, dan efisiensi pemanfaatan sumber daya untuk deteksi malware di aplikasi Android.

1.2 Rumusan Masalah

1. Bagaimana cara menangani permasalahan overfitting pada model Random Forest yang digunakan untuk deteksi malware Android?
2. Bagaimana meminimalkan waktu deteksi malware tanpa mengurangi akurasi prediksi ?
3. Bagaimana meningkatkan kinerja algoritma Random Forest dalam mendeteksi malware pada aplikasi Android tanpa mengorbankan performa perangkat?

1.3 Tujuan Penelitian

1. Mengembangkan metode atau teknik optimalisasi model Random Forest untuk mencegah overfitting, sehingga meningkatkan generalisasi model terhadap data baru dalam proses deteksi malware pada aplikasi Android.
2. Merancang dan mengimplementasikan pendekatan yang efektif untuk mempercepat proses deteksi malware pada Android menggunakan algoritma Random Forest tanpa mengorbankan tingkat akurasi prediksi.
3. Mengembangkan strategi optimalisasi algoritma Random Forest yang efisien dalam mendeteksi malware pada Android, dengan mempertahankan konsumsi sumber daya perangkat (CPU, memori, dan baterai) tetap minimal.

1.4 Manfaat

1. Memberikan kontribusi terhadap pengembangan ilmu pengetahuan di bidang keamanan siber, khususnya pada pengembangan algoritma Random Forest untuk deteksi malware pada platform Android.

2. Menyediakan referensi baru bagi peneliti lain yang tertarik untuk meningkatkan efisiensi dan akurasi algoritma pembelajaran mesin dalam mendeteksi ancaman keamanan digital.
3. Menghasilkan model deteksi malware berbasis Random Forest yang lebih andal dan efisien, sehingga dapat diimplementasikan pada sistem keamanan aplikasi Android.
4. Meningkatkan kemampuan perangkat Android dalam mendeteksi malware tanpa mengorbankan performa perangkat, seperti konsumsi baterai, memori, atau prosesor.
5. Membantu pengembang aplikasi keamanan dalam meningkatkan kualitas produk mereka, khususnya pada efisiensi dan akurasi deteksi ancaman malware.

1.5 Batasan Masalah

1 Penelitian ini hanya menggunakan dataset malware yang tersedia secara terbuka, yaitu *Android Permission Dataset*.

2 Penelitian ini hanya berfokus pada optimasi dan peningkatan kinerja algoritma Random Forest.

3 Penelitian ini terbatas pada perangkat berbasis Android, dan tidak menguji algoritma di platform mobile lain seperti iOS, yang mungkin memiliki karakteristik sistem dan keamanan berbeda.

4 Penelitian ini difokuskan pada perangkat Android dengan sumber daya terbatas (CPU, memori, daya baterai), sehingga hasilnya mungkin tidak dapat diimplementasikan secara langsung pada perangkat Android dengan performa tinggi atau perangkat non-mobile seperti server.

5 Penelitian ini dilakukan dalam lingkungan simulasi dan tidak menguji penerapan deteksi malware dalam skala besar atau di lingkungan produksi yang sebenarnya, yang mungkin menghadapi tantangan tambahan seperti lalu lintas data besar dan berbagai kondisi jaringan.

BAB II

TINJAUAN PUSTAKA

2.1 Studi Literatur

No	Judul &Penulis	Permasalahan & Solusi	Algoritma	Hasil & Kekurangan
1.	Mitigating the Risks of Malware Attacks with Deep Learning Techniques - (Alnajim et al., 2023)	Perkembangan malware yang terus meningkat, dapat menimbulkan bahaya besar bagi individu, bisnis, dan aset digital di seluruh dunia, merupakan fokus utama penelitian ini. Karena evolusi malware baru yang begitu cepat, pendekatan deteksi malware konvensional, seperti metode berbasis tanda tangan, sudah tidak lagi memadai. Selain itu, klasifikasi malware yang canggih dan akurat diperlukan karena metode yang ada saat ini tidak dapat menangani kompleksitas metodologi pembuatan malware baru. -	Convolutional Neural Network (CNN)	Model yang diusulkan berkinerja sangat baik pada dataset tolok ukur Malimg, dengan akurasi 98,14%. Ketika diuji pada dataset BIG 2015, model ini mencapai akurasi yang lebih baik lagi yaitu 98,95%, melampaui metode-metode mutakhir sebelumnya. Dengan akurasi, recall, dan skor F1 yang sangat baik, model ini secara efektif mengatasi tantangan klasifikasi malware, yang mengindikasikan potensinya sebagai metode yang dapat diandalkan untuk penggunaan di dunia nyata dalam sistem keamanan siber.

		Para peneliti mengusulkan pendekatan berbasis pembelajaran mendalam baru yang memanfaatkan proses perhatian ganda dan jaringan saraf konvolusi (CNN) untuk klasifikasi malware. Dengan mengubah biner malware menjadi gambar dan memanfaatkan pendekatan perhatian spasial, metodologi yang diusulkan secara signifikan meningkatkan ekstraksi fitur dengan berkonsentrasi pada wilayah yang relevan dari input. Model CNN dengan perhatian ganda ini meningkatkan ketepatan dan efektivitas deteksi dan klasifikasi malware ketika digunakan bersama dengan arsitektur MobileNetV1.		- Terdapat sejumlah masalah dengan penelitian klasifikasi malware yang menggunakan dataset Malimg. Meskipun memiliki akurasi pelatihan yang baik, model yang disarankan berkinerja buruk selama validasi, mengindikasikan masalah generalisasi. Keterbatasan penelitian ini termasuk penekanan eksklusifnya pada dataset Malimg, yang mungkin membatasi seberapa luas hasilnya dapat diterapkan pada dataset lain atau situasi dunia nyata. Selain itu, terdapat ketidakseimbangan kelas yang mencolok dalam dataset, yang dapat berdampak pada seberapa baik kinerja

				model pada jenis malware yang kurang umum. Validasi eksternal pada berbagai dataset masih kurang, yang sangat penting untuk menentukan seberapa tangguh model yang disarankan. Terakhir, kurangnya pendanaan dari luar dapat membatasi sumber daya yang tersedia untuk pengujian dan validasi menyeluruh dari prosedur yang digunakan.
2.	Innovative Approach to Android Malware Detection: Prioritizing Critical Features Using Rough Set Theory -	Keterbatasan analisis statis dalam mengidentifikasi malware di Android dibahas dalam penelitian ini karena analisis ini tidak dapat merekam peristiwa runtime seperti interaksi jaringan dan kebocoran data. Selain itu, karena pembuat malware	Support Vector Machines (SVM), K-Nearest Neighbor, Random Forest, dan Logistic	Dengan akurasi deteksi 97% yang menggunakan keempat kategori fitur, model yang diusulkan berkinerja lebih baik daripada teknik deteksi canggih sebelumnya. Menggabungkan banyak kategori

	(Gupta et al., 2024)	menggunakan teknik penyamaran, analisis statis tidak dapat secara akurat menggambarkan maksud dasar kode, yang berakibat pada terlewatnya aktivitas berbahaya. - Dengan mencampurkan beberapa jenis fitur untuk deteksi dan menggunakan teori himpunan kasar, metode yang diusulkan meningkatkan akurasi sekaligus memprioritaskan karakteristik utama. Ada empat langkah yang membentuk metodologi ini: pra-pemrosesan data, pemeringkatan fitur, deteksi, dan penghitungan reduksi himpunan awal. Penelitian di masa depan bertujuan untuk mengintegrasikan teknik analisis dinamis dan mengembangkan model berbasis klien-server untuk	Regressio n	karakteristik secara signifikan meningkatkan akurasi deteksi, menunjukkan bahwa metode yang diusulkan mampu mengidentifikasi secara akurat aplikasi Android yang berbahaya atau tidak. - Sifat statis dari penelitian malware pada Android memiliki keterbatasan, termasuk ketidakmampuan untuk merekam perilaku runtime aplikasi, termasuk konektivitas jaringan dan kebocoran data. Selain itu, pengembang virus dapat menggunakan teknik penyamaran yang mencegah analisis statis untuk mengungkapkan tujuan sebenarnya dari kode tersebut. Model yang

		deteksi ponsel pintar secara real-time.		disarankan tidak dapat dipasang di ponsel cerdas untuk deteksi waktu nyata karena ini juga merupakan model di luar perangkat.
3	Swarm Optimization and Machine Learning for Android Malware Detection - (Jhansi et al., 2022)	Penelitian ini mengatasi tantangan dalam mendeteksi malware Android secara akurat dengan memeriksa panggilan antarmuka pemrograman aplikasi (API), yang menawarkan data penting untuk identifikasi malware. Kesulitan membedakan antara malware dan program jinak serta kebutuhan akan teknik identifikasi yang cepat membutuhkan sistem pendeteksian malware yang lebih baik yang dapat secara efektif mengelola ruang fitur yang besar. - Cara mengatasi masalah ini dengan	Linear Regression (LR), Decision Tree (DT), Random Forest(RF) ,K–Nearest Neighbor(KNN) dan Support Vector Machine(SVM).	Hasil eksperimen menunjukkan peningkatan yang signifikan dalam akurasi deteksi malware, mencapai 98,87% hanya dengan tujuh karakteristik yang dipilih di antara seratus atribut permintaan API. Optimalisasi data sebesar 93% menunjukkan bahwa pendekatan yang diusulkan secara efektif mengurangi ruang fitur sambil mempertahankan kinerja klasifikasi yang baik. - Temuan ini kurang dapat diterapkan pada

		mengidentifikasi aspek yang paling penting dari permintaan API menggunakan encoder otomatis bersama dengan tiga teknik optimasi swarm: Ant Lion Optimization (ALO), Cuckoo Search Optimization (CSO), dan Firefly Optimization (FO). Mereka kemudian menggunakan beberapa pengklasifikasi pembelajaran mesin yang terkenal, termasuk Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest (RF), Decision Tree (DT), dan Linear Regression (LR), untuk mengevaluasi teknik-teknik yang terinspirasi dari alam ini. Pengklasifikasi saraf tiruan hibrida (ANC) juga diperkenalkan untuk meningkatkan akurasi		varian malware yang lebih rumit atau unik karena ketergantungannya pada kumpulan data tertentu. Selain itu, ketahanan model terhadap serangan lawan - risiko serius dalam deteksi malware kontemporer - tidak diperiksa dalam penelitian ini. Karena Pengklasifikasi Syaraf Tiruan (ANC) hanya dapat dievaluasi dalam kombinasi tertentu dengan algoritme pengoptimalan (ALO, CSO, dan FO), kinerjanya dengan pendekatan alternatif belum diselidiki. Kemampuan swarm optimization untuk mengurangi dimensi fitur secara signifikan bergantung pada pengaturan manusia, yang mungkin

		klasifikasi malware Android.		berdampak pada hasilnya. Selain itu, efektivitas komputasi dan waktu reaksi metode ini pada perangkat dengan sumber daya terbatas masih dipertanyakan karena belum dievaluasi dalam pengaturan waktu nyata.
4	Similarity-Based Hybrid Malware Detection Model Using API Calls -(Alhashmi et al., 2023)	Penelitian ini membahas tantangan dalam mendeteksi malware yang menggunakan taktik penipuan untuk menghindari deteksi menggunakan metode standar. Pertumbuhan malware yang cepat, terutama yang dapat menyembunyikan aktivitas destruktifnya, membuat teknologi pendeteksi malware saat ini sulit untuk mempertahankan keakuratannya. Tingginya prevalensi positif palsu	Hybrid Similarity-Based API Malware Detection Model (HAPI-MDM)	Model HAPI-MDM berkinerja lebih baik daripada model pendeteksian malware lainnya, dengan akurasi keseluruhan 97,91%. Model ini juga memiliki tingkat positif palsu dan negatif palsu terendah, yang mengindikasikan bahwa dia dapat mendeteksi malware yang dikenal maupun yang sulit dipahami. Menggabungkan analisis berbasis

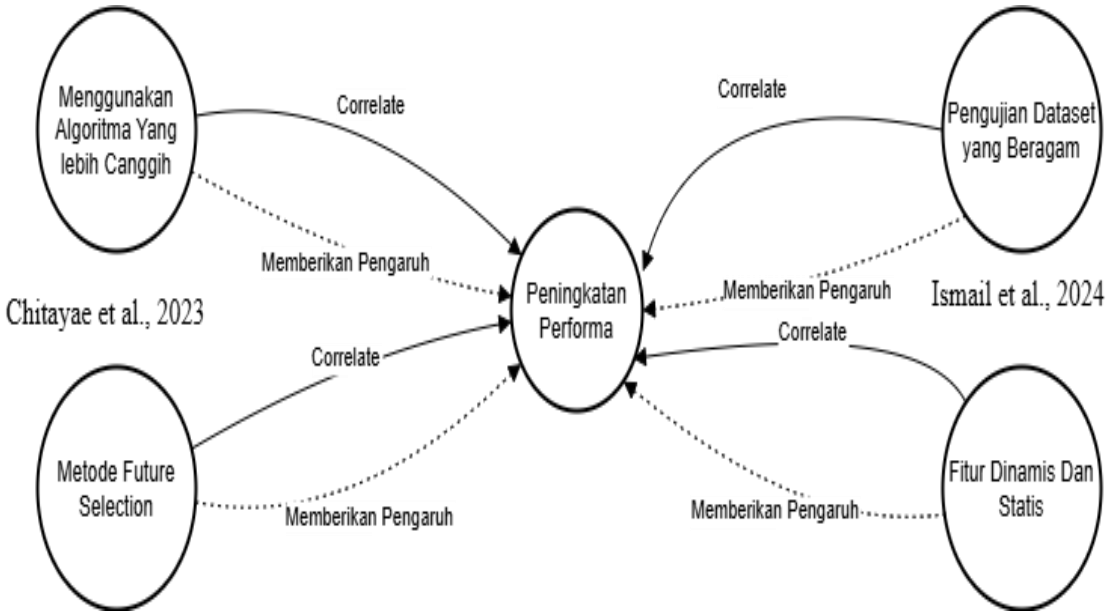
		dan negatif palsu mengurangi efektivitas tindakan keamanan siber. - Solusi yang direkomendasikan adalah Model Deteksi Malware Hibrida Berbasis Kemiripan (HAPI-MDM), yang menggabungkan analisis statis dan dinamis dari panggilan API. Pendekatan ini memanfaatkan teknik pembelajaran mesin, termasuk XGBoost sebagai pengekstrak fitur dan Jaringan Syaraf Tiruan (JST) untuk klasifikasi. Teknik ini meningkatkan akurasi deteksi dan membantu menemukan malware yang telah disembunyikan dengan memanfaatkan karakteristik berbasis kemiripan pada analisis statis dan dinamis.		kemiripan dengan properti dinamis dan statis secara signifikan meningkatkan akurasi identifikasinya. - Meliputi biaya komputasi yang terkait dengan analisis dinamis, yang mungkin berdampak pada skalabilitas saat mengelola jumlah sampel yang tinggi. Selain itu, kualitas dan relevansi fitur yang diekstraksi memiliki dampak yang signifikan terhadap kinerja model, dan malware yang menggunakan pengaburan yang kompleks dapat berdampak pada kemampuan deteksi model. Seiring dengan munculnya jenis malware baru dan teknik penyamaran,

				model ini mungkin juga perlu diperbarui atau dilatih ulang secara berkala agar berhasil.
5	Advanced Android Malware Detection through Deep Learning Optimization - (Alhussen, 2024)	Tantangan yang dihadirkan oleh ancaman malware yang terus berkembang dan keterbatasan sistem pendeteksi malware yang ada - terutama terkait skalabilitas dan generalisasi model - keduanya dibahas dalam makalah ini. Studi ini menyoroti masalah termasuk kumpulan data yang tidak seimbang yang dapat menyebabkan model yang bias mendukung kelas mayoritas. Dengan mengurangi sensitivitas dan membuat generalisasi yang buruk pada kelas minoritas, model-model ini dapat membuat pengukuran kinerja menjadi tidak akurat dan meningkatkan risiko negatif palsu.	Long Short-Term Memory (LSTM) dan Neural Network (NN)	Tingkat akurasi, presisi, dan recall yang sangat baik diperoleh oleh model LSTM dan NN, menurut temuan penelitian; kinerja model LSTM setara dengan model NN, yang mencapai akurasi 0,99 untuk kedua kelas. Dengan menggunakan pendekatan evaluasi yang ketat dan validasi silang untuk memastikan keakuratan dan ketangguhan penilaian model, penelitian ini menunjukkan seberapa efektif model yang diusulkan membedakan antara aplikasi berbahaya dan tidak berbahaya. -

		- Untuk mengatasi masalah ini, penelitian ini memanfaatkan teknik pembelajaran mendalam yang mutakhir, khususnya model Long Short-Term Memory (LSTM) dan Neural Network (NN), untuk meningkatkan kemampuan deteksi malware. Selain itu, untuk mengatasi ketidakseimbangan kumpulan data, penelitian ini menggunakan SMOTE untuk penyeimbangan kelas dan Tensor Processing Unit (TPU) untuk mempercepat proses pelatihan. Penelitian ini menekankan pentingnya persiapan data yang cermat, penyesuaian hyperparameter, dan potensi integrasi arsitektur deep learning yang canggih, seperti mekanisme perhatian dan		peneletian ini terdapat kelemahan yaitu tidak melakukan tuning hyperparameter, yang dapat mempengaruhi akurasi model dan juga tidak melakukan balancing kelas dapat menyebabkan model yang bias, yang mengarah pada sensitivitas yang rendah dan generalisasi yang buruk terhadap kelas minoritas

		model hibrida, untuk meningkatkan kinerja dan kemampuan beradaptasi model.		

2.2 Research Positioning



Gambar 2.2 Research Gap

2.3 Landasan Teori

2.3.1 Cyber Security

Penggunaan metodologi canggih yang memastikan setiap detail dari lingkungan komputer dan perangkat yang terhubung terlindungi dari penyalahgunaan,penghancuran,perubahan, dan invasi merupakan definisi dari Cyber Security menurut (Mamidi & Reddy Mamidi, 2024).Hal ini memungkinkan organisasi untuk menghadapi ancaman baru, menciptakan teknologi canggih, mempersiapkan tenaga kerja yang terampil, dan menjaga aset serta data yang penting dari *Malware*.

2.3.1.1 *Malware*

Malware didefinisikan sebagai perangkat lunak atau program yang ditulis dengan maksud berbahaya, yang berasal dari istilah 'mal' untuk '*malicious*(berbahaya)' dan '*software*' untuk 'perangkat lunak'. Ini adalah jenis serangan siber yang signifikan dapat mengganggu aktivitas sehari-hari pengguna dan dapat mengambil berbagai bentuk serangan, termasuk yang dikodekan secara cerdas untuk mengubah bentuk dan perilakunya, sehingga tidak terdeteksi oleh sistem antivirus tradisional yang mengandalkan tanda tangan malware yang sudah ada.(P et al., n.d. 2023)

2.3.2 Mobile

Menurut (Saeed, 2024), Aplikasi pada perangkat seluler yang menawarkan efisiensi dan kenyamanan untuk berbagai tugas sehari-hari, mulai dari bisnis hingga rekreasi, disebut sebagai seluler. Karena aplikasi seluler mengelola informasi keuangan, kesehatan, dan informasi pribadi pengguna, maka sangat penting untuk melindungi informasi sensitif dari akses yang tidak diinginkan.

2.3.2.1 Android

Android adalah sistem operasi seluler yang telah berkembang pesat dalam popularitas, khususnya di lingkungan seluler, dan dicirikan oleh arsitektur dan mekanisme keamanannya (Thorat et al., 2024). Sistem operasi ini mendukung berbagai macam aplikasi dan telah menjadi platform yang penting bagi para

pengembang dan pengguna. Namun, kebangkitan Android juga menyebabkan peningkatan malware yang menargetkan sistem operasi ini, sehingga mendorong penelitian ekstensif ke dalam metode pendeteksian, terutama yang didasarkan pada teknik machine learning.

2.3.3 *Machine Learning*

Dalam penelitian ini, *machine learning* (ML) digunakan sebagai teknik untuk mendeteksi malware di aplikasi Android dengan merumuskan tugas sebagai masalah klasifikasi biner, di mana aplikasi jinak diklasifikasikan sebagai sampel negatif dan aplikasi berbahaya sebagai sampel positif. Penelitian ini menekankan pentingnya berbagai teknik pra-pemrosesan data, pengurangan dimensi, dan pemilihan pengklasifikasi yang sesuai, seperti *support vector machines* (SVM) dan random forests (RF), untuk meningkatkan akurasi dan kemampuan menjelaskan proses deteksi malware (Palma et al., 2024).Selain itu, penelitian ini menyoroti peran pemilihan fitur dalam mengidentifikasi karakteristik yang paling relevan untuk mengklasifikasikan aplikasi, terutama berfokus pada izin sebagai indikator signifikan keberadaan malware.

2.3.4 Random Forest

Random Forest didefinisikan sebagai algoritma yang digunakan dalam data mining untuk membuat model prediksi. Algoritma ini beroperasi dengan memanfaatkan beberapa Pohon Keputusan, masing-masing dibangun dari sampel acak dari data asli. Prediksi dari setiap Decision Tree kemudian digabungkan untuk menghasilkan hasil akhir. Random Forest secara khusus digunakan untuk tugas klasifikasi, memprediksi kelas data, dan terkenal dengan ketangguhannya terhadap noise pada data karena penggunaan beberapa Decision Tree, yang juga membuatnya lebih mudah untuk dipahami.(Alnajim et al., 2023)

2.3.5 WebSite

Menurut (Chishti et al., 2024), Peran Website dalam kehidupan sehari-hari, khususnya dalam e-commerce, yang memungkinkan bisnis berinteraksi dengan konsumen, berekspansi ke area baru, dan menjadi lebih kompetitif, menunjukkan pentingnya situs web. Selain itu, situs web adalah sumber daya berharga untuk komunikasi, pengumpulan informasi, pembelajaran, dan rekreasi.

BAB III

METODOLOGI PENELITIAN

3.1 Penentuan Objek Penelitian

Penelitian ini mengkaji sebuah aplikasi Android yang dievaluasi untuk deteksi malware menggunakan teknik Random Forest. Data dari dataset tertentu, yaitu Android Permission Dataset, termasuk dalam aplikasi yang diperiksa. Karena dataset ini mencakup berbagai aplikasi yang telah diklasifikasikan sebagai "aman" atau "berisiko," sehingga dapat menjadi bahan uji untuk melihat efektivitas model dalam membedakan aplikasi berbahaya dari aplikasi yang aman.

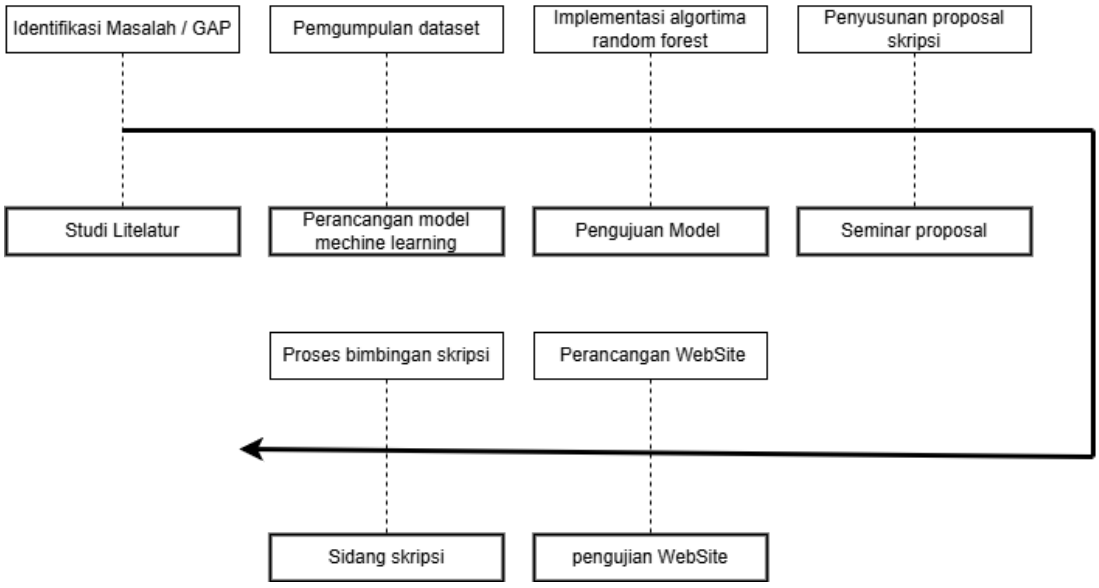
Faktor-faktor berikut mendorong pemilihan dataset ini sebagai topik studi:

1. Representasi Malware yang Bervariasi: Berbagai izin aplikasi termasuk dalam dataset untuk membantu model dalam mengidentifikasi pola aktivitas malware.
2. Ketersediaan fitur yang mendukung Deteksi Malware: Algoritma Random Forest dapat dilatih dan diuji untuk mengidentifikasi pola terkait malware menggunakan Android Permission Dataset, yang menyediakan berbagai data relevan, termasuk izin aplikasi.

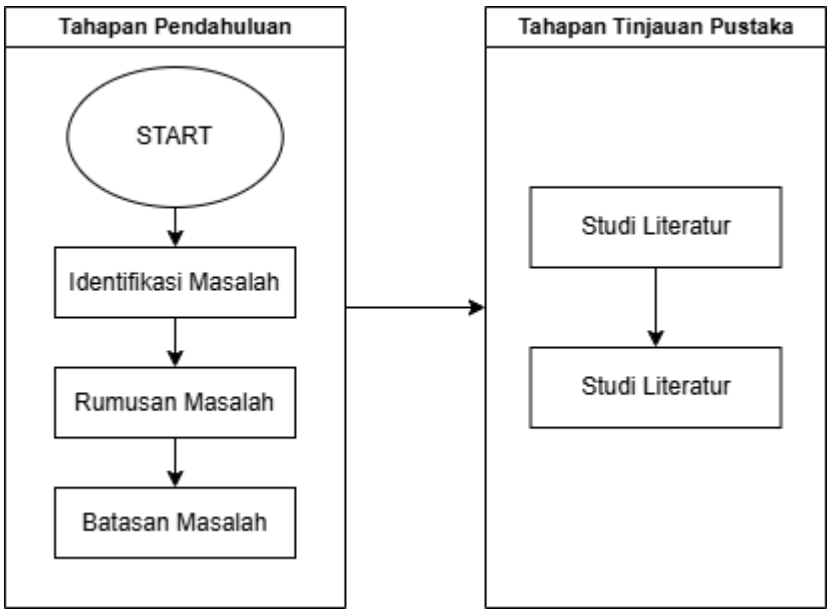
Tujuan studi untuk meningkatkan kinerja dan akurasi algoritma Random Forest dalam deteksi malware pada perangkat Android didukung oleh penentuan topik ini.

3.2Waktu Penelitian

Roadmap



3.3 Flowchart



Berikut ialah penjelasan mengenai setiap tahapan pada desain penelitian ini :

3.3.1 Tahapan Pendahuluan

- **Identifikasi Masalah :** Mengidentifikasi kebutuhan mendasar untuk mendeteksi malware secara efektif dan efisien. Masalah yang mungkin terjadi adalah meningkatnya ancaman keamanan dari malware terhadap perangkat pengguna, dan kurangnya sistem otomatis untuk deteksi.
- **Rumusan Masalah :** Menjabarkan pernyataan masalah yang lebih spesifik,
- **Batasan Masalah :** Menetapkan batasan penelitian seperti jenis malware yang dianalisis, teknik machine learning yang digunakan, dan batasan dataset yang akan digunakan. Hal ini dilakukan agar penelitian tidak meluas dan lebih fokus pada masalah inti.

3.3.2 Tahapan Tinjauan Pustaka

- **Studi Literatur**: Melakukan tinjauan terhadap penelitian sebelumnya terkait deteksi malware, teknik klasifikasi dalam machine learning, serta model-model dan algoritma yang umum digunakan dalam keamanan siber. Studi literatur ini bertujuan untuk mengidentifikasi metode terbaik serta celah atau perbaikan yang dapat dilakukan dalam pendekatan yang akan diambil.
- **Hasil Tinjauan**: Berdasarkan studi literatur, diidentifikasi algoritma yang paling relevan untuk mendeteksi malware, seperti algoritma Random Forest, metode preprocessing data dan teknik seleksi fitur yang sesuai untuk data keamanan siber akan dicatat.

3.4 Dataset

Android Permission Dataset menyediakan rincian tentang izin yang diminta oleh aplikasi, termasuk akses ke kamera, kontak, lokasi, dan kemampuan lain yang sering dieksploitasi oleh malware untuk tujuan jahat. Karena dapat menggambarkan pola perilaku aplikasi yang aman dan berbahaya, dataset ini dipilih untuk membantu algoritma mengidentifikasi izin unik dari setiap jenis aplikasi. Dataset ini digunakan

untuk melatih dan menguji model Random Forest, yang membantu mengidentifikasi program berbahaya dengan lebih akurat dengan mendeteksi pola izin yang tidak biasa.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 476 entries, 0 to 475
Data columns (total 17 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              476 non-null    int64
 1   text            476 non-null    object
 2   relations       476 non-null    object
 3   diagnosis       475 non-null    object
 4   solutions       476 non-null    object
 5   id_1            476 non-null    int64
 6   label_1         476 non-null    object
 7   start_offset_1  476 non-null    int64
 8   end_offset_1    476 non-null    int64
 9   id_2            476 non-null    int64
 10  label_2         476 non-null    object
 11  start_offset_2  476 non-null    int64
 12  end_offset_2    476 non-null    int64
 13  id_3            329 non-null    float64
 14  label_3         329 non-null    object
 15  start_offset_3  329 non-null    float64
 16  end_offset_3    329 non-null    float64
dtypes: float64(3), int64(7), object(7)
memory usage: 63.3+ KB
```

Berikut adalah penjelasan tentang Dataset yang digunakan :

3.4.1 Penjelasan Dataset

Dataset yang Anda berikan adalah sebuah DataFrame dari library Pandas di Python, yang berisi 476 entri (baris) dan 17 kolom. Berikut adalah rincian dari setiap kolom yang ada dalam dataset ini:

3.4.2 Struktur Umum

- Jumlah Entri: 476
- Jumlah Kolom**: 17
- Tipe Data: Terdapat tiga tipe data utama: `int64`, `float64`, dan `object`.

3.4.3 Rincian Kolom

1. ‘id’:

- Tipe: `int64`
- Deskripsi: ID unik untuk setiap entri dalam dataset.

2. ‘text’:

- Tipe: `object`
- Deskripsi: Teks atau konten yang mungkin berisi informasi terkait diagnosis atau solusi.

3. ‘relations’:

- Tipe: `object`
- Deskripsi: Informasi mengenai hubungan antara entri-entri dalam dataset.

4. ‘diagnosis’:

- Tipe: `object`
- Deskripsi: Diagnosis yang terkait dengan entri tersebut. Terdapat satu nilai null pada kolom ini (475 non-null).

5. ‘solutions’:

- Tipe: `object`
- Deskripsi: Solusi yang diusulkan untuk diagnosis yang diberikan.

6. ‘id_1’:

- Tipe: `int64`
- Deskripsi: ID untuk entitas pertama dalam relasi.

7. ‘label_1’:

- Tipe: `object`
- Deskripsi: Label atau kategori untuk entitas pertama.

8. ‘start_offset_1’ dan ‘end_offset_1’:

- Tipe: `int64`
- Deskripsi: Offset untuk menentukan posisi awal dan akhir dari label pertama dalam teks.

9. ‘id_2’:

- Tipe: `int64`
- Deskripsi: ID untuk entitas kedua dalam relasi.

10. ‘label_2’:

- Tipe: `object`
- Deskripsi: Label atau kategori untuk entitas kedua.

11. ‘start_offset_2’ dan ‘end_offset_2’:

- Tipe: `int64`
- Deskripsi: Offset untuk menentukan posisi awal dan akhir dari label kedua dalam teks.

12. ‘id_3’:

- Tipe: `float64` (329 non-null)
- Deskripsi: ID untuk entitas ketiga dalam relasi, dengan beberapa nilai null (329 non-null).

13. ‘label_3’:

- Tipe: `object` (329 non-null)
- Deskripsi: Label atau kategori untuk entitas ketiga, juga dengan beberapa nilai null.

14. ‘start_offset_3’ dan ‘end_offset_3’:

- Tipe: `float64` (329 non-null)
- Deskripsi: Offset untuk menentukan posisi awal dan akhir dari label ketiga dalam teks, dengan beberapa nilai null.

3.4.4 Analisis Data

- Ada beberapa kolom yang memiliki nilai null, khususnya pada kolom yang berkaitan dengan entitas ketiga (`id_3`, `label_3`, `start_offset_3`, dan `end_offset_3`), yang menunjukkan bahwa tidak semua entri memiliki informasi lengkap mengenai entitas ini.
- Offset digunakan untuk menunjukkan posisi label dalam teks, yang bisa berguna untuk analisis lebih lanjut seperti ekstraksi informasi atau pemrosesan bahasa alami (NLP).

3.5 Story Board

(Gambaran Hasil Akhir Dekstop)

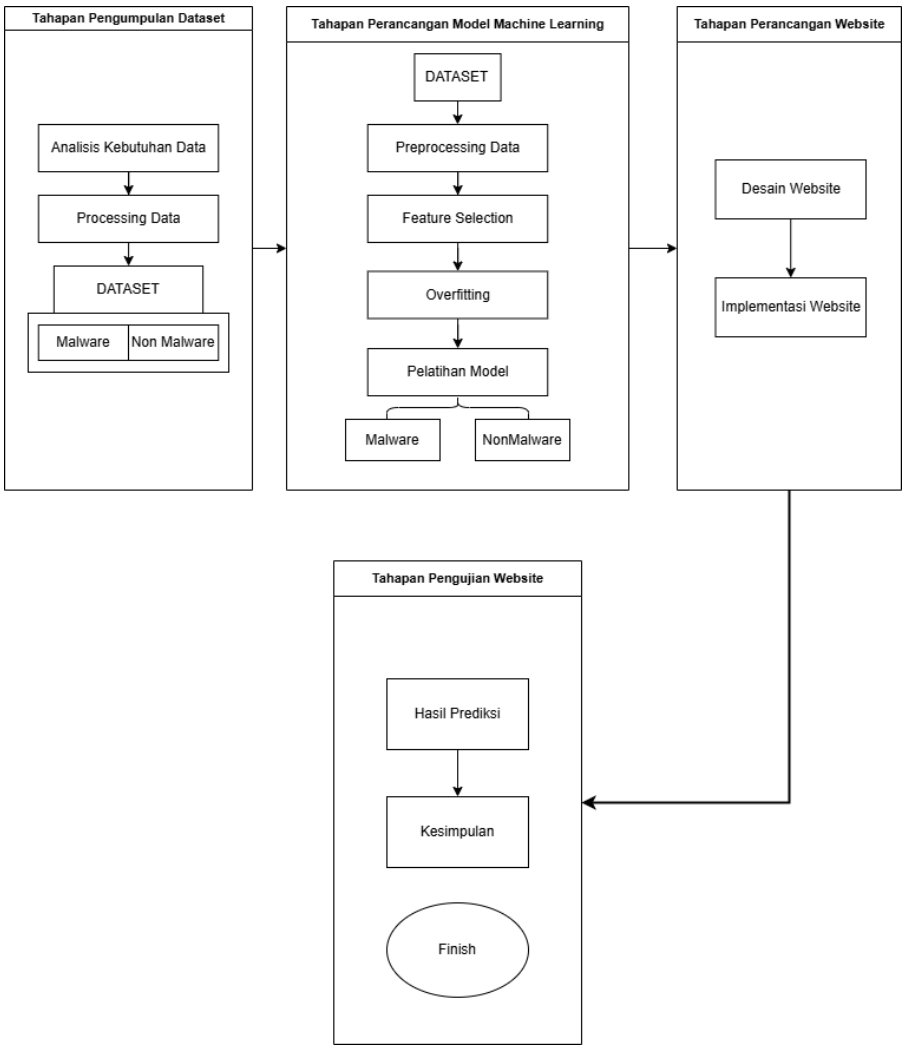
3.6Teknis Pengumpulan Data & Sumber Data

Untuk mengumpulkan data untuk studi ini, Android Permission Dataset harus diunduh dari sumber terbuka yang terpercaya. Dataset ini berisi kumpulan izin aplikasi Android yang telah dibagi ke dalam kategori "aman" dan "berisiko" sesuai dengan status keamanannya. Informasi ini diambil langsung dari dataset yang telah terstruktur sebelumnya dan kemudian diproses untuk memenuhi kebutuhan penelitian. Sebelum data digunakan dalam pengembangan dan pengujian algoritma Random Forest, data tersebut harus melalui prosedur verifikasi dan persiapan, termasuk segmentasi data, normalisasi, dan penghapusan atribut atau izin yang tidak perlu.

Android Permission Dataset, yaitu dataset terbuka yang berisi informasi tentang izin aplikasi Android yang dikategorikan sebagai "aman" dan "berisiko" (menandakan adanya malware), menjadi sumber data untuk studi ini. Metode Random Forest dilatih

dan diuji menggunakan dataset ini, yang mengandung rincian tentang izin yang dicari oleh aplikasi Android. Komputer dapat mempelajari karakteristik ini dan meningkatkan akurasi deteksi berkat signifikansi data dalam memetakan pola izin yang umum ditemukan dalam program malware. Model ini juga dapat berlatih mengidentifikasi berbagai risiko malware di platform Android berkat beragam aplikasi yang terdapat dalam dataset ini.

3.7 Desain Penelitian



Berikut ialah penjelasan mengenai setiap tahapan pada desain penelitian ini :

3.7.1 Tahapan Pengumpulan Dataset

- **Analisis Kebutuhan Data**: Menganalisis kebutuhan data yang diperlukan, seperti jenis fitur (misalnya, ciri khas file malware, pola perilaku aplikasi) dan label (malware atau non-malware). Tahap ini juga memastikan bahwa dataset yang dikumpulkan memenuhi kebutuhan penelitian.
- **Processing Data**: Melakukan proses pembersihan data, seperti menghapus data duplikat, menangani data yang hilang, serta menyelaraskan data untuk memastikan konsistensi. Data yang tidak relevan atau tidak lengkap dapat merusak akurasi model.
- **Dataset Malware dan Non-Malware**: Mengklasifikasikan dataset menjadi dua kategori: malware dan non-malware. Proses ini melibatkan pengelompokan berdasarkan ciri khas dari file atau aplikasi yang termasuk malware maupun yang tidak, misalnya berdasarkan kode biner, pola komunikasi jaringan, atau pola file executable.

3.7.2 Tahapan Perancangan Model Machine Learning

- **Dataset**: Menggunakan dataset yang telah dikategorikan dan diproses pada tahap sebelumnya sebagai data latih untuk model machine learning.
- **Preprocessing Data**: Tahapan ini melibatkan praproses seperti normalisasi, standar data, dan encoding (jika ada data berbentuk teks atau kategori). Hal ini bertujuan untuk meningkatkan performa model dalam memahami dan mengklasifikasikan data.
- **Feature Selection**: Memilih fitur-fitur yang paling relevan untuk membantu model memahami pola malware. Fitur yang tidak relevan atau redundan dapat dihilangkan untuk meningkatkan efisiensi dan akurasi model.
- **Overfitting**: Melakukan teknik seperti regularisasi, cross-validation, dan pengaturan hyperparameter untuk mencegah overfitting, yaitu ketika model "menghafal" data latih dan tidak mampu memprediksi data baru dengan baik.

- **Pelatihan Model**: Melatih model menggunakan algoritma machine learning terpilih. Model ini akan dilatih untuk mengenali pola-pola yang membedakan malware dari non-malware dengan menggunakan dataset yang ada.
- **Pengujian Model**: Setelah pelatihan, model diuji menggunakan data uji untuk mengukur akurasi, presisi, recall, dan metrik lainnya. Model yang memenuhi kriteria akurasi yang ditetapkan akan digunakan lebih lanjut.

3.7.3 Tahapan Perancangan Website

- **Desain Website**: Mendesain website yang user-friendly, di mana pengguna bisa mengunggah file atau data untuk dideteksi apakah termasuk malware atau non-malware. Desain ini memperhatikan tata letak, navigasi, dan tampilan yang intuitif bagi pengguna.
- **Implementasi Website**: Membuat website berdasarkan desain yang telah dirancang. Tahap ini meliputi pengembangan frontend (antarmuka pengguna) dan backend (pemrosesan dan penyimpanan data). Website juga akan diintegrasikan dengan model machine learning yang telah dilatih untuk melakukan prediksi.

3.7.4 Tahapan Pengujian Website

- **Integrasi Model ke Website**: Model machine learning diintegrasikan ke dalam sistem backend website, sehingga website dapat memanfaatkan model untuk melakukan klasifikasi malware dan non-malware.
- **Hasil Prediksi**: Menyediakan halaman atau area di website yang menampilkan hasil prediksi, apakah file atau data yang diunggah termasuk dalam kategori malware atau non-malware. Hasil ini juga bisa dilengkapi dengan informasi tambahan seperti tingkat keyakinan model.
- **Kesimpulan**: Meringkas hasil penelitian dengan menarik kesimpulan berdasarkan kinerja model machine learning dan feedback dari pengujian

website. Kesimpulan ini dapat mencakup efektivitas model, akurasi, dan area untuk pengembangan lebih lanjut.

- **Finish**: Menandakan bahwa seluruh tahapan penelitian telah selesai, dari identifikasi masalah hingga implementasi solusi.

BAB IV

HASIL DAN PEMBAHASAN

BAB V

KESIMPULAN DAN SARAN

DAFTAR PUSTAKA

Alhashmi, A. A., Darem, A. A., Alashjaee, A. M., Alanazi, S. M., Alkhaldi, T. M., Ebad, S. A., Ghaleb, F. A., & Almadani, A. M. (2023). Similarity-Based Hybrid Malware Detection Model Using API Calls. *Mathematics*, *11*(13). https://doi.org/10.3390/math11132944

Alhussen, A. (2024). Advanced Android Malware Detection through Deep Learning Optimization. *Engineering, Technology and Applied Science Research*, *14*(3), 14552–14557. https://doi.org/10.48084/etasr.7443

Alnajim, A. M., Habib, S., Islam, M., Albelaihi, R., & Alabdulatif, A. (2023). Mitigating the Risks of Malware Attacks with Deep Learning Techniques. *Electronics (Switzerland)*, *12*(14). https://doi.org/10.3390/electronics12143166

Chishti, S. A., Ardekani, I., & Varastehpour, S. (2024). AI-Enhanced Personality Identification of Websites. *Information (Switzerland)*, *15*(10). https://doi.org/10.3390/info15100623

Chitayae, N., Muhammad, A. H., Kalimantan, U., & Yogyakarta, A. (2023). Identifikasi Malware pada Android menggunakan Algoritma K-Nearest Neighbor. *JOURNAL OF INFORMATION TECHNOLOGY*, *3*(2).

Gupta, R., Sharma, K., & Garg, R. K. (2024). Innovative Approach to Android Malware Detection: Prioritizing Critical Features Using Rough Set Theory. *Electronics (Switzerland)*, *13*(3). https://doi.org/10.3390/electronics13030482

Ismail, H., Utomo, R. G., & Bawono, M. W. A. (2024). Comparison of Support Vector Machine and Random Forest Method on Static Analysis Windows Portable Executable (PE) Malware Detection. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, *8*(1), 154. https://doi.org/10.30865/mib.v8i1.7110

Jhansi, K. S., Varma, P. R. K., & Chakravarty, S. (2022). Swarm Optimization and Machine Learning for Android Malware Detection. *Computers, Materials and Continua*, *73*(3), 6327–6345. https://doi.org/10.32604/cmc.2022.030878

Lakshmanarao, A., & Shashi, M. (2022). Android Malware Detection using Multilayer Autoencoder and Random Forest. *International Journal of Engineering Trends and Technology*, *70*(11), 249–257. https://doi.org/10.14445/22315381/IJETT-V70I11P227

Mamidi, S., & Reddy Mamidi, S. (2024). *Future Trends in AI Driven Cyber Security*. https://www.researchgate.net/publication/383915013

P, S. S., Tiwari, A., & Chaudhari, N. S. (n.d.). *Obfuscated Memory Malware Detection*.

Palma, C., Ferreira, A., & Figueiredo, M. (2024). Explainable Machine Learning for Malware Detection on Android Applications †. *Information (Switzerland)*, *15*(1). https://doi.org/10.3390/info15010025

Rafrastara, F. A., Supriyanto, C., Paramita, C., Astuti, Y. P., & Ahmed, F. (2023). *Performance Improvement of Random Forest Algorithm for Malware Detection on Imbalanced Dataset using Random Under-Sampling Method*. *8*(2). https://orangedatamining.com/

Saeed, S. (2024). Usable Privacy and Security in Mobile Applications: Perception of Mobile End Users in Saudi Arabia. *Big Data and Cognitive Computing*, *8*(11), 162. https://doi.org/10.3390/bdcc8110162

Thorat, P., Rathod, M., & Shinde, S. (2024). MALWARE DETECTION IN ANDROID. *ANDROID Article in International Research Journal of Modernization in Engineering Technology and Science*. www.irjmets.com