PENGEMBANGAN DAN EVALUASI METODE STACKING BERBASIS ENSEM UNTUK DETEKSI MALWARE DENGAN PEN AN AUGMENTASI DATA PADA DATASET TIDAK SEIMBANG

SKRIPSI

Diajukan Sebagai Syarat Untuk Menyelesaikan Pendidikan
Program Sarjana (S-1)
Pada Program Studi Rekayasa Sistem Komputer

Oleh :

FERRO AUDI PAJRIN
NIM : 2102010013

PROGRAM STUDI REKAYASA SISTEM KOMPUTER
FAKULTAS ILMU TEKNIK
UNIVERSITAS BINA INSAN
2025

HALAMAN PENGESAHAN SKRIPSI

PENGEMBANGAN DAN EVALUASI METODE STACKING BERBASIS ENSEMBLE UNTUK DETEKSI MALWARE DENGAN PENDEKATAN AUGMENTASI DATA PADA DATASET TIDAK SEIMBANG

Oleh :

FERRO AUDI PAJRIN

NIM : 2102010013

Lubuklinggau, 24 Januari 2025

Pembimbing I	Pembimbing II
Dr. Muhamad Akbar, S.T., M.IT.	Deni Nurdiansyah, M.Kom.

Mengesahkan,

Dekan Fakultas Ilmu Teknik

Universitas Bina Insan,

Dr.Rudi Kurniawan, S.T., M.Kom.

HALAMAN PERSETUJUAN TIM PENGUJI

Pada hari Jumat tanggal 24 bulan Januari tahun. 2025 telah dilaksanakan sidang Skripsi oleh Program Studi Rekayasa Sistem Komputer Fakultas Ilmu Teknik Universitas Bina Insan.

Nama	: Ferro Audi Pajrin
NIM	: 2102010013
Judul Skripsi	: Pengembangan Dan Evaluasi Metode Stacking Berbasis Ensemble Untuk Deteksi Malware Dengan Pendekatan Augmentasi Data Pada Dataset Tidak Seimbang

Komisi penguji

1. Ketua	: Dr. Muhamad Akbar, S.T., M.IT.	( )
2. Sekretaris	: Deni Nurdiansyah, M.Kom.	( )
3. Anggota	: Novi Lestari, M.Kom	( )

Mengetahui,

Kepala Program Studi Rekayasa Sistem Komputer

Universitas Bina Insan

Armanto, M.Kom.

HALAMAN M[illegible] DAN PERSEMBAHAN

MOTTO :

- **Hidup adalah perjalanan, bukan perlombaan. Nikmati setiap langkahnya.**
- **Kesederhanaan bukan tanda kelemahan, melainkan kekuatan untuk fokus pada yang utama**
- **Jika tidak ada jalan, buatlah jalan**

Karya tulis ini Ku persembahkan Untuk :

- **Allah SWT – Sumber segala kekuatan, ilmu, dan petunjuk yang telah membimbing setiap langkah perjalanan ini. Segala puji hanya bagi-Mu, Ya Allah.**
- **Keluarga tercinta – Ayah, Ibu, dan adik-adik, yang selalu memberikan cinta tanpa syarat, doa yang tiada henti, serta dukungan yang menjadi fondasi bagi setiap keberhasilan saya.**
- **Sahabat dan teman seperjuangan – Rekan-rekan yang telah berbagi semangat, inspirasi, dan kebersamaan selama perjalanan akademik ini. Kehadiran kalian sangat berarti dalam menghadapi setiap tantangan.**
- **Para pembimbing dan dosen – Sosok yang dengan dedikasi tinggi memberikan ilmu, arahan, dan motivasi, sehingga saya dapat menyelesaikan karya ini dengan baik.**
- **Almamater – Tempat saya bertumbuh, belajar, dan memperkaya diri dengan ilmu pengetahuan serta pengalaman yang akan selalu saya kenang.**

HALAMAN PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama Mahasiswa/I : Ferro Audi Pajrin

Nim : 2102010013

Program Studi : Rekayasa Sistem Komputer

Fakultas : Ilmu Teknik

Menyatakan dengan sesungguhnya bahwa penelitian dan penulisan skripsi yang saya susun sebagai persyaratan untuk memperoleh gelar Sarjana (S-1) Universitas Bina Insan, Merupakan hasil kerja saya sendiri dan tidak menyuruh orang lain yang mengerjakannya. Ada pun bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain dan telah saya tuliskan sumbernya secara jelas sesuai dengan norma, kaidah dan etika penulisan ilmiah.

Jika dikemudian hari ternyata terbukti bahwa penelitian dan tugas akhir ini bukan hasil kerja saya sendiri atau plagiat dalam bagian-bagian tertentu, maka saya bersedia sikenakan sanksi sesuai dengan peraturan perundangan yang berlaku.

Lubuklinggau, 24 Januari 2025

Penulis,

Materai

10.000

Ferro Audi Pajrin

NIM 2102010013

ABSTRACT

With the increasing n malware attacks on Android devices, traditional signature-based dete hods have shown limitations in detecting new malware. This study aims to develop a stacking ensemble method combined with SMOTE data augmentation to enhance malware detection accuracy on imbalanced datasets. The TUANDROMD dataset, consisting of 4,465 malware and goodware samples, was used in this research. Several machine learning algorithms, including Random Forest, SVM, and Extra Trees, were employed as base classifiers and combined using Logistic Regression as the meta-classifier.

The results indicate that the stacking ensemble method with hyperparameter tuning achieved 99.58% accuracy, 100% precision, 99.16% recall, 99.58% F1-score, and 99.95% AUC. Additionally, the SMOTE data augmentation technique successfully improved the representation of the minority class, leading to better overall model performance. These findings demonstrate that the proposed method provides an effective solution to enhance Android device security against malware threats.

Keywords:

Malware Detection, Stacking Method, Ensemble Learning, SMOTE, Hyperparameter Tuning

Abstrak

Seiring dengan meningkatnya serangan malware pada perangkat Android, metode deteksi tradisional berbasis tanda tangan mulai menunjukkan keterbatasan dalam mendeteksi malware baru. Penelitian ini bertujuan mengembangkan metode stacking berbasis ensemble dengan teknik augmentasi data menggunakan SMOTE untuk meningkatkan akurasi deteksi malware pada dataset tidak seimbang. Dataset TUANDROMD digunakan sebagai data penelitian, yang terdiri dari 4.465 sampel malware dan goodware. Beberapa algoritma pembelajaran mesin digunakan sebagai *base classifiers*, yaitu Random Forest, SVM, dan Extra Trees, yang kemudian digabungkan menggunakan Logistic Regression sebagai meta-classifier.

Hasil penelitian menunjukkan bahwa metode stacking ensemble dengan tuning hyperparameter menghasilkan akurasi sebesar 99,58%, presisi 100%, recall 99,16%, F1-score 99,58%, dan AUC 99,95%. Selain itu, teknik augmentasi data SMOTE berhasil meningkatkan representasi kelas minoritas sehingga memperbaiki performa model secara keseluruhan. Hasil ini menunjukkan bahwa metode yang dikembangkan mampu memberikan solusi yang efektif untuk meningkatkan keamanan perangkat Android terhadap serangan malware.

Kata Kunci:

Deteksi Malware, Metode Stacking, Ensemble Learning, SMOTE, Hyperparameter Tuning

KATA PENGANTAR

Alhamdulillah, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas limpahan rahmat, karunia, dan kemudahan yang diberikan, sehingga skripsi ini dapat diselesaikan dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan program pendidikan Sarjana (S1) pada Program Studi Rekayasa Sistem Komputer, Fakultas Teknik, Universitas Bina Insan Lubuk Linggau. Solawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, beserta keluarga, para sahabat, dan seluruh umatnya hingga akhir zaman.

Dalam penyusunan skripsi ini, penulis telah berusaha dengan maksimal untuk menyajikan karya yang memenuhi standar akademik, baik dari segi substansi maupun penyajian. Namun, penulis menyadari bahwa karya ini masih memiliki keterbatasan yang dapat memengaruhi kesempurnaannya. Oleh karena itu, penulis dengan senang hati menerima kritik dan saran yang bersifat membangun, guna memperbaiki dan meningkatkan kualitas Skripsi ini di masa mendatang.

Untuk selanjutnya penulis mengucapkan terimaksih kepada pihak-pihak yang teklah membantu dalam menyelesaikan Skripsi ini yaitu :

1) Allah SWT, yang telah memberikan segala nikmat dan karunia-Nya higga saya dapat menyelesaikan Skripsi ini.
2) Bapak Dr. H. Sardiyo, M.M. selaku Rektor Universitas Bina Insan.
3) Bapak Dr. Muhammad Akbar, S.T, M.IT. Selaku Wakil Rektor I Universitas Bina Insan Lubuklinggau.
4) Bapak Wakhid Mukhlis, S.Pd., M.Pd.,M.M. Selaku Wakil Rektor II Universitas Bina Insan Lubuklinggau.
5) Dr. Rudi Kurniawan, S.T, M.Kom. selaku Dekan Fakultas Ilmu Teknik Universitas Bina insan Lubuklinggau.
6) Armanto, M.Kom. selaku Ketua Kaprodi Rekayasa Sistem Komputer Universitas Bina Insan.
7) Bapak Dr. Muhammad Akbar, S.T, M.IT. Selaku Dosen Pembimbing I yang telah memberikan bimbingan dan arahan dalam penulisan Skripsi ini.

8) Bapak Deni Nurdiansyah, M.Kom. selaku Dosen Pembimbing II yang telah memberikan bimbingan dan arahan dalam penulisan Skripsi ini.
9) Dosen penguji Bapak N[illegible]i, M.Kom. Terimakasih atas arahannya.
10) Bapak/Ibu Seluruh D[illegible]n Karyawan Universitas Bina Insan Lubuklinggau yang telah banyak memberikan ilmu pengetahuan dan bimbingan kepada penulis.
11) Kedua Orang Tua dan Keluarga besar yang telah memberikan doa dan dukungan baik moril dan material dalam penulisan Skripsi ini.
12) Teman-teman seperjuangan terimakasih atas kebersamaan dan motivasinya.

Akhir kata semoga penelitian ini dapat bermanfaat untuk penelitian selanjutnya.

Lubuklinggau, 24 Januari 2025

Penulis

DAFTAR RIWAYAT HIDUP

Biodata

Nama	: Ferro Audi Pajrin
Tempat / Tanggal lahir	: Muara Kati Baru I, 28 Mei 2003
Jenis Kelamin	: Laki-laki
Agama	: Islam
Alamat	: Sp. PT. Sawit GSSL Dusun IV Desa Muara Kati Baru I

Pendidikan

- SD : SDN 1 Muara Kati Baru I
- SMP/MTS Sederajat : SMPN 1 Muara Kati Baru I
- SMA/ MAN/SMK Sederajat : SMAN 2 Muara Beliti

Pengalaman Organisasi dan Pelatihan (opsional)

1. Wakil Ketua Divisi E-Fun Universitas Bina Insan Lubuklinggau tahun 2021
2. Anggota HMP Rekayasa Sistem Komputer tahun 2021

Prestasi Akademik dan Non-Akademik (opsional)

NO	Prestasi Akademik dan Non-Akademik	Tahun
1.	3th Indonesian Student & Lecturer Competition (ISLC) 2023	2023

DAFTAR ISI

Halaman

DAFTAR TABEL

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang Penelitian

Seiring dengan pertumbuhan penggunaan smartphone yang terus meningkat, pada tahun 2023 diperkirakan pengiriman smartphone secara global mencapai 1,48 miliar perangkat. Pada tahun 2021, jumlah pengguna smartphone diproyeksikan mencapai 6,4 miliar, dan angka ini kemungkinan akan meningkat menjadi 7,5 miliar pada tahun 2026 [1]. Android, sebagai sistem operasi mobile dengan pangsa pasar 73%, menjadi pilihan utama berkat fleksibilitasnya yang mendukung berbagai inovasi teknologi, seperti GPS, Wi-Fi, kamera, dan aplikasi berbasis internet. Namun, popularitas ini juga menjadi sasaran utama serangan siber, seperti *malware* yang dirancang untuk mengeksploitasi data sensitif pengguna dan mengganggu perangkat mereka [2].

Malware, atau *malicious software*, adalah perangkat lunak berbahaya yang dirancang untuk mencuri data, mengganggu sistem, atau menyusupi perangkat tanpa izin pengguna. Malware Android hadir dalam berbagai bentuk, termasuk ransomware, spyware, dan Trojan horse. *Malware* sering kali memanfaatkan kerentanan pada aplikasi atau sistem operasi *Android* untuk menjalankan serangannya [3]. Ancaman ini terus berkembang seiring dengan meningkatnya jumlah pengguna Android dan kompleksitas malware yang muncul.

Metode tradisional dalam deteksi *malware*, seperti berbasis tanda tangan (*signature-based*), memiliki keterbatasan dalam mendeteksi malware yang belum dikenal ataupun yang telah dimodifikasi. Pendekatan ini cenderung kurang adaptif terhadap evolusi malware yang terus berkembang. Oleh karena itu, diperlukan metode deteksi yang lebih canggih, adaptif, dan akurat. *Machine learning*, khususnya pendekatan berbasis *ensemble* seperti *stacking*, telah terbukti sebagai solusi yang menjanjikan untuk meningkatkan akurasi deteksi malware [4].

Beberapa penelitian yang berkaitan dengan deteksi malware Android antara lain penelitian yang dilakukan oleh [5] berjudul *"Enhancing Android Malware Detection Through Ensemble Stacking Classifiers and Regularization-Based*

Feature Selection" yang mengembangkan metode deteksi malware berbasis ensemble stacking dengan s fitur berbasis regularisasi. Penelitian ini menunjukkan akurasi ting n deteksi malware, namun tidak mengimplementasikan teknik a i data, yang merupakan tantangan utama dalam deteksi malware yang tidak seimbang. Penelitian selanjutnya dilakukan oleh [4] yang berjudul *"Deteksi Malware menggunakan Metode Stacking Berbasis Ensemble"* menggunakan meta-classifier Logistic Regression dalam pendekatan stacking ensemble. Walaupun hasilnya menunjukkan akurasi yang cukup tinggi, penelitian ini juga tidak membahas teknik augmentasi data, yang penting untuk penanganan ketidakseimbangan data dalam dataset. Penelitian lain yang dilakukan oleh [6] yang berjudul *"Android Ransomware Detection Using Supervised Machine Learning Techniques Based on Traffic Analysis"* yang berfokus pada deteksi ransomware Android menggunakan algoritma supervised learning dan augmentasi data (SMOTE), namun penelitian ini tidak menggunakan metode stacking ensemble.

Berdasarkan kajian penelitian sebelumnya, penelitian ini bertujuan untuk mengatasi tantangan dataset yang tidak seimbang melalui penerapan teknik augmentasi data. Penelitian ini akan mengembangkan sistem deteksi malware yang lebih akurat, adaptif, dan tangguh menggunakan metode stacking berbasis ensemble [4]. Pendekatan ini dipilih karena dapat meningkatkan performa algoritma data mining, khususnya jika dibandingkan dengan penggunaan single algorithm secara konvensional [4]. Selain itu penelitian ini akan mengintegrasikan optimasi algoritma dengan *Cross-validatio*n untuk memperkirakan tingkat kesalahan uji dengan cara memisahkan sebagian data pelatihan untuk divalidasi [7], proses *tuning hyperparameternya* akan dilakukan menggunakan *GridSearchCV* sebab dapat meningkatkan performa model secara signifikan dalam mendeteksi *malware* [8], untuk mengatasi ketidak seimbangan dataset akan diterapkan metode SMOTE (*Synthetic Minority Over-sampling Technique*) yang digunakan untuk menangani ketidak seimbangan kelas dalam data yang tidak peka terhadap kasus-kasus dalam kelas minoritas yang mungkin memiliki nilai prediktif yang penting [9]. Penelitian ini akan menggunakan dataset TUANDROMD, berisi

4.465 sampel malware dan goodware berbasis izin dan API Android yang telah diproeses sebelumnya oleh Te[illegible]niversity, untuk memastikan kinerja model dalam mendeteksi malware [illegible]Android. Penelitian ini akan menguji generalisasi model terhadap va[illegible]dalam dataset yang sama melalui teknik validasi dan optimasi algoritma. Berdasarkan pendekatan ini, peneliti merancang sebuah studi dengan judul: *"Pengembangan dan Evaluasi Metode Stacking Berbasis Ensemble untuk Deteksi Malware Dengan Pendekatan Augmentasi Data pada Dataset Tidak Seimbang"*.

1.2 Identifikasi Masalah

Penelitian ini berfokus pada tantangan dalam mendeteksi malware Android yang semakin canggih. Masalah utama yang dihadapi adalah:

a. Metode deteksi malware tradisional, seperti yang berbasis tanda tangan (signature-based), memiliki keterbatasan dalam menghadapi evolusi malware yang semakin kompleks dan adaptif. Di sisi lain, kurangnya penerapan teknik optimasi algoritma serta tuning hyperparameter dalam proses pengembangan model deteksi menyebabkan performanya belum mencapai tingkat yang diharapkan
b. Ketidakseimbangan jumlah antara data malware dan goodware dalam dataset merupakan tantangan signifikan dalam pengembangan model deteksi. Kondisi ini menyebabkan model cenderung bias terhadap kelas mayoritas (goodware), sehingga mengurangi kemampuan dalam mendeteksi malware yang termasuk dalam kelas minoritas secara akurat

1.3 Rumusan Masalah

Berdasarkan Latar Belakang masalah yang telah dipaparkan sebelumnya, maka Rumusan maslah penelitian ini sebagai berikut :

a. Bagaimana mengembangkan metode deteksi malware berbasis stacking yang mampu meningkatkan akurasi dengan mengintegrasikan optimasi algoritma dan tuning hyperparameter?

b. Bagaimana augmentasi data menggunakan teknik SMOTE dapat mengatasi ketidakseimbangan data[illegible] meningkatkan kemampuan model dalam mendeteksi malware sec[illegible]efektif?

1.4 Batasan Masalah

Supaya pembahasan ini dilakukan lebih terarah, maka perlu ditentukan batas permasalahan. Adapun Batasan pada penelitian ini adalah:

a **Dataset yang Digunakan**: Penelitian ini hanya menggunakan dataset *TUANDROMD*, yang berfokus pada *malware* Android terbaru. Dataset ini berisi 4465 contoh dan 241 atribut dengan kategori target (*malware vs goodware*).

b **Fokus pada Metode *Ensemble Stacking*:** Penelitian ini hanya berfokus pada deteksi *malware* Android dengan menggunakan teknik *stacking* berbasis *ensemble,* tanpa menguji metode deteksi lain seperti deep learning atau teknik lainnya.

c ***Optimasi Algoritma***: Optimasi algoritma dilakukan menggunakan teknik validasi silang (cross-validation) untuk meminimalkan risiko overfitting.

d **Augmentasi Data dengan *SMOTE*:** Augmentasi data dilakukan dengan teknik *Synthetic Minority Over-sampling Technique (SMOTE)* untuk mengatasi ketidakseimbangan data antara kelas malware dan goodware.

e ***Tuning Hyperparameter***: Pengaturan dan *tuning hyperparameter* menggunakan *Grid Search* ini dilakukan untuk memaksimalkan kinerja model deteksi *malware*.

f **Fokus pada Deteksi *Malware Android*:** Penelitian ini difokuskan pada deteksi malware yang menyerang perangkat Android, mengingat platform ini populer dan rentan terhadap ancaman malware yang terus muncul.

1.5 Tujuan dan Manfaat Penelitian

a) Tujuan Penelitian

1) Mengembangkan m eksi malware berbasis stacking yang lebih akurat dengan me sikan optimasi algoritma dan tuning hyperparameter untuk meningkatkan performa model deteksi.
2) Mengeksplorasi efektivitas teknik augmentasi data menggunakan SMOTE dalam mengatasi ketidakseimbangan dataset, sehingga meningkatkan kemampuan model dalam mendeteksi malware secara lebih andal dan adaptif.

b) Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1) Manfaat Akademis: Penelitian ini akan memperkaya literatur tentang deteksi malware menggunakan metode stacking ensemble yang dioptimalkan, serta eksplorasi teknik augmentasi data SMOTE untuk meningkatkan akurasi deteksi pada dataset tidak seimbang.
2) Manfaat Praktis: Hasil penelitian ini diharapkan dapat memberikan solusi untuk meningkatkan performa sistem deteksi malware pada perangkat Android, dengan mengoptimalkan algoritma dan memanfaatkan teknik augmentasi data untuk mengatasi ketidakseimbangan dataset.
3) Manfaat Sosial: Penelitian ini berkontribusi pada peningkatan keamanan dunia maya, melindungi data pribadi pengguna, dan mengurangi risiko kerugian akibat serangan malware, sehingga meningkatkan rasa aman bagi masyarakat digital.

1.6 Sistematika Penulisan

Untuk memberikan gambaran secara garis besar dari laporan skripsi, berikut akan diuraikan secara singkat sistematika penulisannya, yang terdiri 5 bab dan masing-masing sub-bab dengan relevan terhadap permasalahan yang dibahas. Secara sistematis, isi dari masing-masing bab tersebut adalah sebagai berikut:

BAB I : PENDAHULUAN

Dalam bab ini berisikan latar belakang, identifikasi masalah, perumusan masalah, batasan masalah, tujuan dan manfaat penelitian, dan sistematika penelitian.

BAB II : KAJIAN PUSTAKA

Dalam bab ini berisikan literatur, penelitian relevan, dan kerangka berpikir.

BAB III : METODOLOGI PENELITIAN

Dalam bab ini berisikan analisa sistem, teknik pemilihan informan, dan tempat & waktu penelitian.

BAB IV : HASIL PENELITIAN DAN PERANCANGAN

Dalam bab ini berisikan gambaran umum, hasil dan pembahasan.

BAB V : KESIMPULAN DAN SARAN

Dalam bab ini berisikan tentang kesimpulan dari seluruh penelitian.

BAB II
TIN N PUSTAKA

2.1 Literatur

2.1.1 Definisi Malware

Malware adalah istilah yang berasal dari gabungan kata *malicious* dan *software*, yang berarti perangkat lunak berbahaya. Program ini dirancang untuk merusak sistem, mencuri data, atau mengumpulkan informasi penting, bahkan mengambil alih akses ke suatu sistem tanpa izin. Penyebarannya bisa melalui berbagai cara, seperti email phishing, teknik rekayasa sosial, hingga menggunakan aplikasi downloader. Biasanya, malware digunakan untuk mencuri data sensitif, mengumpulkan informasi seperti kata sandi dan alamat email, serta menyebarkan spam[10]. Jenis *malware* pun beragam, contohnya *adware*, *Trojan*, *ransomware*, dan lainnya[10]. Cara kerja *malware* adalah dengan masuk ke dalam suatu sistem, seperti sistem komputer ataupun handphone sudah ada, melalui berbagai aplikasi dalam sistem tersebut atau melalui pengiriman data dari perangkat yang sudah terinfeksi *virus*[11].

Malware dapat dibagi menjadi beberapa jenis berdasarkan cara kerjanya dan karakteristiknya [11]. Berikut adalah jenis-jenis *malware:*

a) *Logic Bomb* **:**Malware yang memiliki dua bagian inti, yaitu konten dan pemicu. Jenis malware ini akan aktif jika pemicunya dijalankan. Mirip dengan *time bomb, Logic Bomb* dapat aktif selama jangka waktu tertentu.

b) *Trojan Horse* : Program yang terlihat tidak berbahaya tetapi diam-diam menjalankan tugas berbahaya. Misalnya, saat pengguna ingin masuk (login) ke sebuah situs web dengan memasukkan nama pengguna dan kata sandi, tetapi situs web tersebut telah dipasang program pencuri kata sandi. *Trojan Horse* berperan dengan menampilkan pesan kesalahan yang menyatakan bahwa pengguna salah memasukkan nama pengguna atau kata sandi. Namun, di balik proses tersebut, *Trojan Horse* telah mencuri informasi autentikasi pengguna.

c) *Back Door* : Mekanisme yang melewati proses pemeriksaan standar keamanan, seperti autenti

d) *Virus* : Malware yang a perbanyak dirinya dengan menginfeksi program lain yang sedang

e) *Worm* : Malware yang mirip dengan virus karena dapat menyalin dirinya sendiri. Yang membedakannya adalah worm mampu menyebar dari satu sistem ke sistem lainnya melalui jaringan tanpa memerlukan program yang bisa dieksekusi terlebih dahulu.

f) *Rabbit* : Program yang menggunakan semua sumber daya dalam suatu sistem. Contohnya adalah *Fork Bomb,* yang terus-menerus membuat proses baru secara berulang tanpa batas sehingga membuat sistem menjadi lambat.

g) *Spyware* : Program yang mengambil informasi dari komputer dan mengirimkannya kepada pihak lain.

h) *Adware* : Program yang mirip dengan *Spyware,* tetapi fokus utamanya adalah untuk menampilkan iklan pemasaran.

i) *Zombies* : Istilah untuk komputer yang telah diserang atau diretas tanpa sepengetahuan pengguna. Biasanya aktivitas yang dilakukan adalah menyebarkan email spam.

Salah satu kasus *malware* yang diungkap dalam artikel di *infokomputer.grid.id* menyebutkan bahwa sejak Mei 2019, sejumlah aplikasi Android di Play Store terinfeksi malware jenis baru bernama *Joker*. *Malware* ini berhasil menginfeksi perangkat Android di berbagai negara, termasuk Indonesia, dan menyebabkan kerugian bagi para pemilik perangkat yang terinfeksi. Joker mengakses informasi sensitif seperti data kartu SIM dan melakukan komunikasi dengan server yang menggunakan layanan AWS, serta menargetkan negara tertentu berdasarkan kode MCC *(Mobile Country Code)* perangkat yang terinfeksi. Peningkatan jumlah aplikasi yang terinfeksi ini menunjukkan bahwa *malware Joker* merupakan ancaman serius yang dapat mempengaruhi pengguna Android di seluruh dunia [12].

2.1.2 Sistem Operasi Android

Smartphone adalah sa perangkat mobile yang terus berkembang dan selalu menarik perhatian t. Di antara berbagai sistem operasi yang tersedia, Android menjadi ng populer. Hal ini disebabkan oleh kemudahan penggunaannya serta beragam fitur unggulan yang ditawarkan, sehingga diminati oleh banyak orang di seluruh dunia. Keunggulan Android meliputi antarmuka pengguna *(user interface)* yang intuitif dan pengalaman pengguna *(user experience)* yang nyaman. Selain itu, Android bersifat *open source,* mendukung multitasking, dan memiliki ekosistem aplikasi yang luas untuk memudahkan berbagai aktivitas pengguna sehari-hari [12].

Namun, di balik berbagai kemudahan yang ditawarkan, Android dikenal memiliki kelemahan dalam hal keamanan. Celah keamanan ini kerap dimanfaatkan oleh oknum tidak bertanggung jawab untuk membuat dan menyisipkan malware ke dalam aplikasi di platform Android. Contoh kasusnya adalah malware yang sengaja dirancang untuk mengeksploitasi kelemahan sistem, yang pada akhirnya bisa merugikan pengguna[12].

2.1.3 Teknik Deteksi *Malware*

2.1.3.1 *Machine Learning* untuk Deteksi *Malware*

Machine Learning adalah sekumpulan teknik yang memungkinkan komputer belajar dari data, sehingga dapat menganalisis dan membuat prediksi layaknya manusia yang memiliki kecerdasan [13]. Dalam deteksi *malware*, *Machine Learning* digunakan untuk mengenali pola dan karakteristik dari sampel *malware* yang sudah diketahui, lalu mengklasifikasikan apakah sampel baru termasuk *malware* atau bukan. Teknik ini banyak diterapkan dalam pendeteksian *malware* melalui pendekatan *analisis statis, dinamis, ataupun hybrid* [14].

Pendekatan tradisional untuk mendeteksi *malware* biasanya mengandalkan tanda tangan digital (*signature-base*) atau metode *heuristik* dari sampel malware yang sudah ada. Sayangnya, metode ini memiliki keterbatasan karena kurang efektif dalam mendeteksi *malware* baru atau *malware* yang telah

mengalami modifikasi [14]. *Machine Learning* memungkinkan komputer untuk mempelajari pola dari data lalu dan menggunakan pola tersebut untuk menyelesaikan masalah ata at prediksi. Algoritma dalam *Machine Learning* beragam dan dapat okkan berdasarkan metode pembelajaran, jenis tugas yang dilakukan, atau tingkat kedalaman proses pembelajarannya. Secara umum, terdapat tiga strategi utama dalam *Machine Learning*, yaitu *Supervised Learning, Unsupervised Learning, dan Semi-Supervised Learning* [15].

2.1.3.2 Metode *Ensemble*

Salah satu pendekatan *Machine Learning* yang terbukti efektif dalam mendeteksi *malware* adalah metode *Ensemble*. Teknik ini menggabungkan beberapa model *Machine Learning* untuk meningkatkan akurasi dan kinerja deteksi. Pendekatan *Ensemble* bekerja dengan mengumpulkan hasil dari berbagai model, seperti *decision tree, random forest, atau support vector machine (SVM),* lalu membuat keputusan berdasarkan suara mayoritas atau bobot tertentu pada setiap model. Dengan cara ini, kelemahan dari satu model dapat tertutupi oleh kelebihan model lainnya, sehingga kemampuan deteksi secara keseluruhan menjadi lebih baik[14].

Beberapa metode klasifikasi yang sering digunakan dalam *Machine Learning* antara lain *bagging, boosting, dan stacking* [16]

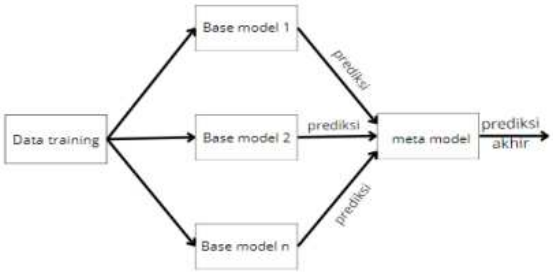
a. Metode *Bagging* adalah metode yang menggabungkan beberapa model pembelajaran untuk meningkatkan performa secara keseluruhan, seperti pada *algoritma Random Forest*. Keputusan akhir dalam klasifikasi diperoleh dari gabungan hasil prediksi sejumlah pohon keputusan yang telah dilatih sebelumnya, lalu digabungkan menjadi satu model akhir [17].
b. *Boosting* adalah teknik pembelajaran mesin yang mampu mengubah pembelajar lemah menjadi pengklasifikasi yang kuat. Ini adalah jenis *meta-algoritma ensemble* yang digunakan untuk mengurangi *bias* dan *varians* [18].

c. *Stacking* adalah teknik ensemble yang memadukan berbagai algoritma data mining untuk membang[illegible]el baru yang lebih kuat dan akurat [4].

2.1.3.3 Metode Stacking Be[illegible]semble

Tidak seperti bagging dan boosting, stacking mengombinasikan berbagai jenis classifier yang berbeda. Dalam metode ini, classifier individual disebut base learner, sementara classifier yang berfungsi menggabungkan hasil dari base learner disebut meta learner atau generalizer [16].

Stacking adalah teknik ensemble learning yang memanfaatkan berbagai algoritma data mining untuk menciptakan model baru yang lebih kuat dan akurat [4]. Dalam stacking, kombinasi berbagai model berbeda melibatkan penggunaan meta-learner. Proses dimulai dengan melatih learner pertama menggunakan dataset asli untuk menghasilkan dataset baru. Dataset baru tersebut kemudian digunakan sebagai input untuk melatih learner di tingkat berikutnya. Output dari learner pertama menjadi fitur input bagi learner kedua, sementara label aslinya tetap dipertahankan sebagai label pada dataset baru. Learner pertama biasanya dibangun menggunakan beragam algoritma pembelajaran yang berbeda [19]. Seperti pada gambar 2.1 dibawah ini :



Gambar 2.1. Tahapan Metode *Stacking*

Langkah-langkah pada metode *stacking* yaitu :

a. Dataset dibagi menjadi dua bagian.
b. Bagian pertama digunakan untuk melatih beberapa base-learner.
c. Base-learner kemudian membuat prediksi pada data di bagian kedua.
d. Hasil prediksi tersebut digunakan sebagai input untuk melatih learner di tahap berikutnya [20].

Berikut adalah berbagai algoritma yang akan diterapkan sebagai base classifier pada level 0 di anta

a. *Random Forest*

Ini adalah algoritma p ran mesin di mana sekumpulan pohon keputusan *(forest of decision trees)* merepresentasikan instance independen dari setiap pohon. Algoritma ini dapat digunakan untuk operasi klasifikasi maupun regresi. Selama proses pelatihan RF, sampel dan pemilihan fitur dilakukan secara acak. Dengan cara ini, *overfitting* dapat dicegah[21].

b. *Support Vector Machine (SVM)*

Support Vector Machine (SVM) adalah algoritma yang digunakan untuk klasifikasi data menjadi dua kelompok. Dengan kata lain, SVM adalah algoritma klasifikasi biner, yang hanya bisa membedakan antara dua kelas, misalnya positif dan negatif. Prinsip kerjanya adalah dengan mencari sebuah hyperplane atau garis pemisah yang dapat membedakan kedua kelas tersebut. Fungsi yang digunakan untuk mendefinisikan garis pemisah ini adalah $wx + b = 0$. Kelas positif berada di sisi di mana nilai $wx + b \geq 1$, sementara kelas negatif ada di sisi dengan nilai $wx + b \leq 1$[22].

c. *Extraa Trees*

Extra Tree adalah metode pembelajaran *ensemble*. Satu-satunya perbedaan dari *Random Forest* terletak pada cara pohon keputusan dibuat. Dalam metode ini, setiap pohon keputusan dibuat berdasarkan hubungan ketergantungan dari sampel data[23]. Algoritma ini menggunakan metode rata-rata (*averaging*) untuk meningkatkan akurasi dan mencegah *overfitting* [23].

Untuk Meta classifernya akan diterapkan pada level 1 dalam penelitian ini yaitu *Logisctick Regression* Ini adalah model pembelajaran terawasi yang didasarkan pada statistik linear. Logika kerjanya adalah memodelkan probabilitas keluaran berdasarkan masukan. Dalam hal ini, algoritma tidak melakukan klasifikasi secara langsung. Model ini memilih nilai *cut-off* selama operasi. Dalam klasifikasi biner, masukan dengan probabilitas kurang dari nilai

cut-off dapat diklasifikasikan ke dalam satu kelas, sedangkan masukan dengan probabilitas lebih besar ke k nya [23].

2.1.4 Augmentasi Data unt et Tidak Seimbang

Pendekatan augmentasi data dengan teknik oversampling bertujuan untuk membuat sampel tambahan dari kelas yang lebih sedikit. Teknik ini dipilih karena tidak mengurangi ukuran dataset dan dapat memberikan hasil yang lebih baik dalam mengatasi ketidakseimbangan antara kelas, terutama ketika jumlah sampel dari kelas minoritas sangat sedikit [9]. Teknik oversampling meliputi berbagai metode, seperti Random Over Sampling (ROS), Synthetic Minority Oversampling Technique (SMOTE), Borderline SMOTE, k-Means SMOTE, Support Vector Machine SMOTE (SVM-SMOTE), dan Adaptive Synthetic (ADASYN) [9]. Dalam penelitian ini, model SMOTE digunakan karena efektif dalam mengatasi ketidakseimbangan kelas, mengurangi overfitting, dan menghasilkan akurasi yang baik [9].

SMOTE (Synthetic Minority Over-sampling Technique) adalah metode oversampling yang digunakan untuk mengatasi ketidak- seimbangan kelas dalam data. Ketidakseimbangan ini terjadi ketika jumlah kelas mayoritas jauh lebih banyak daripada kelas minoritas. Jika masalah ini tidak diatasi, model bisa jadi terlalu fokus pada kelas mayoritas, sehingga tidak sensitif terhadap kelas minoritas, padahal kelas minoritas mungkin mengandung informasi penting. Akibatnya, meskipun akurasi model tinggi karena dominasi kelas mayoritas, prediksi untuk kelas minoritas akan sangat rendah [9].

Teknik ini bertujuan untuk menambah jumlah sampel dalam kelas minoritas dengan menciptakan sampel sintetis berdasarkan data yang sudah ada. Rumus dari teknik SMOTE dapat dilihat pada gambar 2.2 di bawah ini:

$$Xsyn = Xi + (Xknn - Xi) \times \delta \quad (4)$$

Dimana:

$Xsyn$	:	D s
Xi	:	D kan direplikasi
$Xknn$	:	Data yang memiliki jarak terdekat dengan data Xi
δ	:	Nilai random antara 0 dan 1

Gambar 2.2 Rumus Teknik *SMOTE*

SMOTE bekerja dengan memilih sebuah sampel dari kelas minoritas, sebut saja Xi, yang akan diperbanyak. Kemudian, algoritma mencari K-Nearest Neighbors (tetangga terdekat) dari Xi dalam kelas minoritas. Nilai K ditentukan secara acak terlebih dahulu. Setelah itu, perbedaan antara Xi dan salah satu tetangga terdekat $Xknn$ dihitung, kemudian sebuah nilai acak (δ) antara 0 dan 1 dihasilkan dan dikalikan dengan perbedaan tersebut. Langkah terakhir adalah menambahkan hasil perkalian tersebut pada Xi untuk menghasilkan sampel sintetis $Xsyn$ [9].

Menurut penelitian yang dilakukan oleh Turnip et al. (2023), Synthetic Minority Over-Sampling Technique (SMOTE) diterapkan untuk mengatasi masalah ketidakseimbangan kelas dalam dataset malware. Dataset yang digunakan berisi data APK dari Virus Share dengan total 13.076 sampel, mencakup berbagai jenis malware. SMOTE diterapkan untuk meningkatkan jumlah sampel dari kelas minoritas, sehingga model machine learning dapat mempelajari pola dari jenis malware yang kurang terwakili. Hasil eksperimen menunjukkan bahwa akurasi terbaik yang tercapai adalah 92,26%, dengan menggunakan kombinasi SMOTE, yang memungkinkan pengklasifikasian sampel APK yang terdiri dari 13 kelas malware [24].

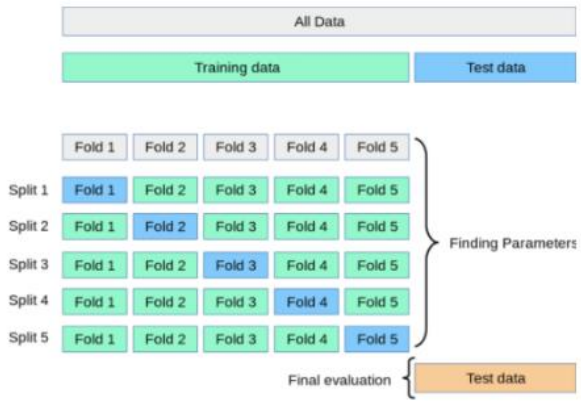
2.1.5 Optimasi Model

2.1.5.1 Optimasi Algoritm

KBBI, Optimasi be kata "optimal," yang berarti sesuatu yang paling baik, paling tinggi, g menguntungkan. Berdasarkan definisi tersebut, optimasi berarti upaya atau cara yang dilakukan untuk memperoleh hasil yang terbaik dari sebelumnya[21]. Pada penelitian ini *Cross-validatio*n digunakan untuk memperkirakan tingkat kesalahan uji dengan cara memisahkan sebagian data pelatihan untuk divalidasi. Dalam hal ini, pendekatan yang digunakan adalah k-fold cross-validation, di mana data dibagi menjadi k subset atau fold dengan ukuran hampir sama. Setiap fold secara bergantian digunakan sebagai data validasi, sementara fold lainnya digunakan untuk melatih model. Proses ini menghasilkan k perkiraan tingkat kesalahan uji, yang kemudian dirata-rata untuk memperoleh hasil akhir [7]. Sebagai contoh:

1) Jika k=5, maka data akan dibagi menjadi 5 subset. Model akan dilatih menggunakan 4 subset dan divalidasi pada subset yang tersisa. Proses ini diulang sebanyak 5 kali, sehingga setiap subset akan digunakan sebagai data validasi satu kali
2) Keuntungan dari k-fold CV adalah kompromi yang baik antara bias dan varians dibandingkan metode lain seperti *validation set* atau LOOCV (*Leave-One-Out Cross-Validation*).

Penerapan validasi silang 5-fold yang akan digunakan dapat dilihat pada gambar 2.3 di bawah ini :



Gambar 2.3. *Cross Validation 5-fold* **Sumber**[16]

2.1.5.2 Optimasi Model d*Tuning Hyperparameter*

GridSearchCV ada de untuk *tuning hyperparameter* yang memungkinkan penggun mengeksplorasi berbagai kombinasi hyperparameter yang telah ditentukan [8]. prinsip dasarnya adalah mencoba semua nilai parameter yang mungkin. Misalkan ada k parameter yang perlu dioptimalkan dalam algoritma ML yang relevan. Dalam kasus ini, grid berukuran k×k akan ditentukan dan kombinasi setiap parameter diproses secara terpisah. Jika jumlah k parameter besar, pencarian akan lebih sulit, oleh karena itu algoritma ini cocok digunakan pada jumlah *hiperparameter* yang lebih kecil[23].

Menurut penelitian yang dilakukan oleh[8] Dengan menggunakan GridSearchCV untuk melakukan tuning hyperparameter pada algoritma Random Forest dalam mendeteksi malware, hasil eksperimen menunjukkan bahwa parameter terbaik yang diperoleh adalah:

a) ***criterion*** = *"entropy"*
b) ***max_depth*** = 128
c) ***max_features*** = "log2"
d) ***min_samples_split*** = 2
e) ***n_estimators*** = 400

Penggunaan parameter terbaik ini memberikan peningkatan kinerja pada model, dengan recall yang naik sebesar 0,37%, serta akurasi dan F1-score yang meningkat masing-masing sebesar 0,19%. Hasil ini mengindikasikan bahwa *tuning hyperparameter* menggunakan *GridSearchCV* dapat secara signifikan meningkatkan performa model dalam mendeteksi *malware* [8].

2.1.6 Evaluasi Performa Model

Evaluasi dilakukan untuk mengukur efektivitas sistem dalam mendeteksi malware di platform Android dengan teknologi machine learning. Penilaian ini meliputi pengukuran akurasi, presisi, recall, dan F1-score untuk memastikan

kesesuaian dengan desain awal serta mengidentifikasi kemungkinan kesalahan, seperti yang terlihat pada gam bawah ini. [25].

		Sebenarnya	
		Positif	**Negatif**
Prediksi	**Positif**	**TN** (True Positive)	**FN** (False Positive)
	Negatif	**FP** (False Negative)	**TN** (True Negative)

Gambar 2.4 Confusion Matrix **Sumber** [14].

Confusion matrix adalah komponen dasar untuk menghitung akurasi, presisi, dan recall. Akurasi mengukur perbandingan antara prediksi yang benar yang diklasifikasikan oleh sistem. Rumus dasar untuk pengukuran dalam penelitian ini adalah sebagai berikut:

a. Accuracy adalah nilai yang digunakan untuk mengukur seberapa tepat sistem dalam mengklasifikasikan data dengan benar, yang dihitung menggunakan rumus pada formula 1 dibawah ini.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

b. Precision (presisi) adalah nilai yang mengukur jumlah data positif yang diklasifikasikan dengan benar, dibagi dengan total data yang diklasifikasikan sebagai positif, yang dihitung menggunakan rumus pada formula 2 dibawah ini.

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

c. Recall adalah nilai yang digunakan untuk mengukur persentase data kategori positif yang berhasil diklasifikasikan dengan benar oleh sistem, yang dihitung menggunakan rumus pada formula 3 dibawah ini.

$$\text{Recall} = \frac{TP}{TP+FN}$$

d. F1-Score adalah nilai rata-rata harmonik dari presisi dan recall. Nilai terbaik F1-Score adalah 1.0, sementara nilai terburuknya adalah 0. Jika F1-Score menunjukkan nilai yang baik, ini mengindikasikan bahwa metode klasifikasi yang digunakan memiliki presisi dan recall yang seimbang, yang dihitung menggunakan rumus pada formula 4 dibawah ini.

$$\text{F1} - \text{score} = \frac{2 \text{ x Precision x Recall}}{\text{Precision+Recall}}$$

e. Area Under the Curve (AUC) adalah ukuran dari area di bawah kurva ROC. Semakin besar area tersebut, semakin baik model klasifikasi yang diusulkan. ROC merupakan gambaran grafis dari hubungan antara sensitivitas dan spesifisitas. Nilai AUC ini menunjukkan seberapa baik model dalam membedakan antara kelas yang berbeda [26].

2.2 Penelitian Terdahulu yang Relevan

Berikut adalah beberapa penelitian relevan yang dijadikan referensi dalam penyusunan skripsi ini:

1) Penelitian pertama berjudul "*Enhancing Android Malware Detection Through Ensemble Stacking Classifiers and Regularization-Based Feature Selection*" oleh Elisa Bahara Soritua dan Ditdit Nugraha Utama. Penelitian ini bertujuan untuk mengembangkan metode deteksi *malware Android* berbasis *ensemble stacking* dengan memanfaatkan seleksi fitur berbasis *regularisasi (Lasso, Ridge, Elastic Net).* Penelitian ini memiliki kesamaan dengan penelitian saya dalam hal membahas deteksi *malware,* menggunakan pendekatan *stacking ensemble,* optimasi algoritma melalui *tuning hyperparameter*, penggunaan *meta-classifier Logistic Regression,* serta algoritma seperti *Random Forest, KNN, SVM, Logistic Regression,* dan *Bernoulli Naive Bayes.* Perbedaannya terletak pada tidak digunakannya *SMOTE,* dan dataset yang digunakan adalah *Drebin*-215.

Hasil penelitian ini menunjukkan pendekatan *stacking ensemble* dengan regularisasi fitur mer an akurasi tinggi sebesar 99,17% dalam deteksi *malware Andr*

2) Penelitian kedua berjudul "*Deteksi Malware* menggunakan Metode *Stacking Berbasis Ensemble"* oleh Fauzi Adi Rafrastara, Catur Supriyanto, Cinantya Paramita, dan Yani Parti Astuti. Penelitian ini menerapkan metode *stacking* berbasis *ensemble* untuk meningkatkan akurasi klasifikasi *file malware* menggunakan dataset publik (API *Call*). Penelitian ini memiliki persamaan dengan penelitian saya dalam hal membahas deteksi *malware,* menggunakan pendekatan *stacking ensemble, meta-classifier Logistic Regression,* serta algoritma seperti *Neural Network, Random Forest*, KNN, dan *Logistic Regression.* Perbedaannya terletak pada tidak digunakannya *SMOTE* (menggunakan *under-sampling*), dataset yang digunakan adalah *VxHeaven* dan Virus Total (bukan dataset Android). Hasil penelitian menunjukkan bahwa metode *stacking* dengan *Logistic Regression* sebagai *meta-classifier* memberikan akurasi tinggi sebesar 98,7%, lebih unggul dibandingkan algoritma individu[4].

3) Penelitian ketiga berjudul "Explainable Classification Model for Android Malware Analysis Using API and Permission-Based Features" oleh Aslam et al. Penelitian ini berfokus pada penggunaan pembelajaran mesin untuk mendeteksi malware pada perangkat Android. Dengan memanfaatkan data terbaru dari dataset TUANDROMD, aplikasi Android diklasifikasikan sebagai malware atau bukan berdasarkan fitur izin dan API. Untuk mengatasi masalah ketidakseimbangan data, beberapa teknik seperti RandomOverSampler, SMOTETomek, dan RandomUnderSampler diterapkan. Dari hasil eksperimen, model Extra Tree yang menggunakan Random OverSampler berhasil mencapai tingkat akurasi tertinggi, yaitu 99,53%, dengan waktu prediksi yang sangat cepat, hanya 0,0198 detik.

Selain itu, penelitian ini juga menggunakan teknik Explainable Artificial Intelligence (EAI) u[illegible]menjelaskan bagaimana model membuat keputusan, di mana[illegible]perti "Receive_Boot_Completed" dan "Kill_Background_Pro[illegible]menjadi penentu utama untuk mendeteksi malware. Studi ini menawarkan solusi yang cepat, akurat, dan transparan untuk membantu mengidentifikasi ancaman malware, meskipun masih ada ruang untuk perbaikan, terutama dalam jumlah fitur dan kemampuan klasifikasi jenis malware [27].

4) Penelitian kedua berjudul Jurnal ini berjudul ***"Random Feature Selection Using Random Subspace Logistic Regression"*** oleh Nuttanan Wichitaksorn et al. Penelitian ini mengusulkan metode *random subspace logistic regression* untuk seleksi fitur secara acak melalui simulasi bootstrap, dengan penerapan pada *standard logistic regression* dan *lasso logistic regression*. Metode ini bertujuan mengatasi masalah beban komputasi tinggi pada regresi logistik konvensional, khususnya pada dataset berdimensi tinggi. Evaluasi dilakukan menggunakan data simulasi serta dataset besar dari UCI Machine Learning Repository dan Kaggle. Hasil penelitian menunjukkan bahwa metode ini mampu mengurangi waktu komputasi secara signifikan sekaligus meningkatkan akurasi prediksi dibandingkan metode regresi logistik standar, dengan akurasi terbaik mencapai **98,43%** pada dataset **TUANDROMD**. Penelitian ini dapat menjadi referensi penting dalam penelitian saya, terutama dalam konteks penggunaan dataset TUANDROMD serta pendekatan seleksi fitur dan optimasi algoritma. Meskipun tidak menggunakan metode *stacking ensemble* seperti penelitian saya, pendekatan random subspace ini dapat dipertimbangkan sebagai metode alternatif dalam meningkatkan performa model klasifikasi[28]

5) Penelitian ketiga berjudul “Outsmarting Android Malware with Cutting-Edge Feature Engineering and Machine Learning Techniques” Jurnal ini membahas deteksi malware Android menggunakan teknik machine learning dengan fokus pada preprocessing data, pemilihan fitur, dan algoritma seperti

Random Forest (RF), Support Vector Classifier (SVC), serta K-Nearest Neighbors (KNN). D[illegible]i Drebin dan TUANDROMD digunakan sebagai benchmark, [illegible] preprocessing meliputi penghapusan duplikasi, imputasi nil[illegible]serta normalisasi data menggunakan Min-Max. Fitur penting seperti permissions, API calls, dan intents dipilih menggunakan Gain-Ratio dan Chi-Squared Test. Hasil penelitian menunjukkan akurasi tinggi, di mana RF dan SVC mencapai 98,9% dalam mendeteksi malware. Meskipun menawarkan framework deteksi yang kuat, metode ini masih terbatas pada dataset tertentu dan bergantung pada feature engineering manual[29]

6) Penelitian Keempat berjudul "Jurnal ini berjudul *"***An effective deep learning scheme for android malware detection leveraging performance metrics and computational resources***"*. Penelitian ini mengusulkan penggunaan model deep learning berbasis **Deep Neural Decision Forest (DNDF)** dan **Deep Belief Network (DBN)** untuk mendeteksi malware Android. Dua dataset digunakan dalam evaluasi, yaitu **Drebin Dataset** untuk membandingkan dengan studi sebelumnya dan **TUANDROMD Dataset (2021)** untuk mendeteksi ancaman terbaru dengan teknik obfuscation dan morphing yang lebih canggih. Fokus utama penelitian ini adalah membandingkan performa DNDF dengan teknik machine learning lainnya, serta menghitung waktu eksekusi dan konsumsi sumber daya komputasi. Hasil eksperimen menunjukkan bahwa model DNDF mencapai akurasi 99%, sensitivitas 1, dan AUC sebesar 0,98%. Hasil ini setara atau bahkan lebih baik dibandingkan dengan metode berbasis machine learning lainnya dan beberapa antivirus komersial. Studi ini relevan sebagai pembanding dalam penelitian saya, khususnya dalam penggunaan dataset **TUANDROMD**, meskipun perbedaan utamanya adalah pendekatan deep learning dan tidak digunakannya metode stacking ensemble atau augmentasi data seperti SMOTE[30]

7) Penelitian keempat berjudul **"**Komparasi Performansi Algoritma *Naive Bayes* dan *Logistic Regression* pada *Malware Android"* oleh Putra Wijaya

dan Santoso. Penelitian ini membandingkan performa algoritma *Naive Bayes* dan *Logistic ion* dalam mendeteksi *malware Android* dengan menggunakan *ermission* dan *Intent* sebagai dataset. Kesamaan dengan pe ya adalah penggunaan algoritma *Naive Bayes dan Logistic Regression* yang dioptimasi menggunakan *Cross Validation.* Perbedaannya terletak pada hasil akurasi yang tidak setinggi penelitian lainnya serta hanya menggunakan fitur *Permission* dan Intent sebagai dataset. Hasil penelitian menunjukkan perbandingan tingkat akurasi antara kedua algoritma[31].

8) Penelitian kelima berjudul "*Android Ransomware Detection Using Supervised Machine Learning Techniques Based on Traffic Analysis*" oleh M. Abdur Rahman, Md. Abdur Rouf, Shamim Ripon, dan Md. Shamsul Alam. Penelitian ini bertujuan untuk mendeteksi *ransomware Android* menggunakan *supervised learning* dengan analisis *traffic* jaringan. Kesamaan dengan penelitian saya adalah membahas deteksi *malware ransomware,* menggunakan teknik augmentasi data *(SMOTE),* dan algoritma *supervised learning* seperti *Random Forest.* Perbedaannya adalah penelitian ini tidak menggunakan *ensemble* atau *stacking ensemble,* fokus pada *ransomware* berbasis jaringan, dan tidak membahas *hyperparameter tuning* secara detail. Hasil penelitian menunjukkan bahwa penggunaan *SMOTE* meningkatkan akurasi hingga 94%, dengan model *Decision Tree* menghasilkan performa terbaik[6].

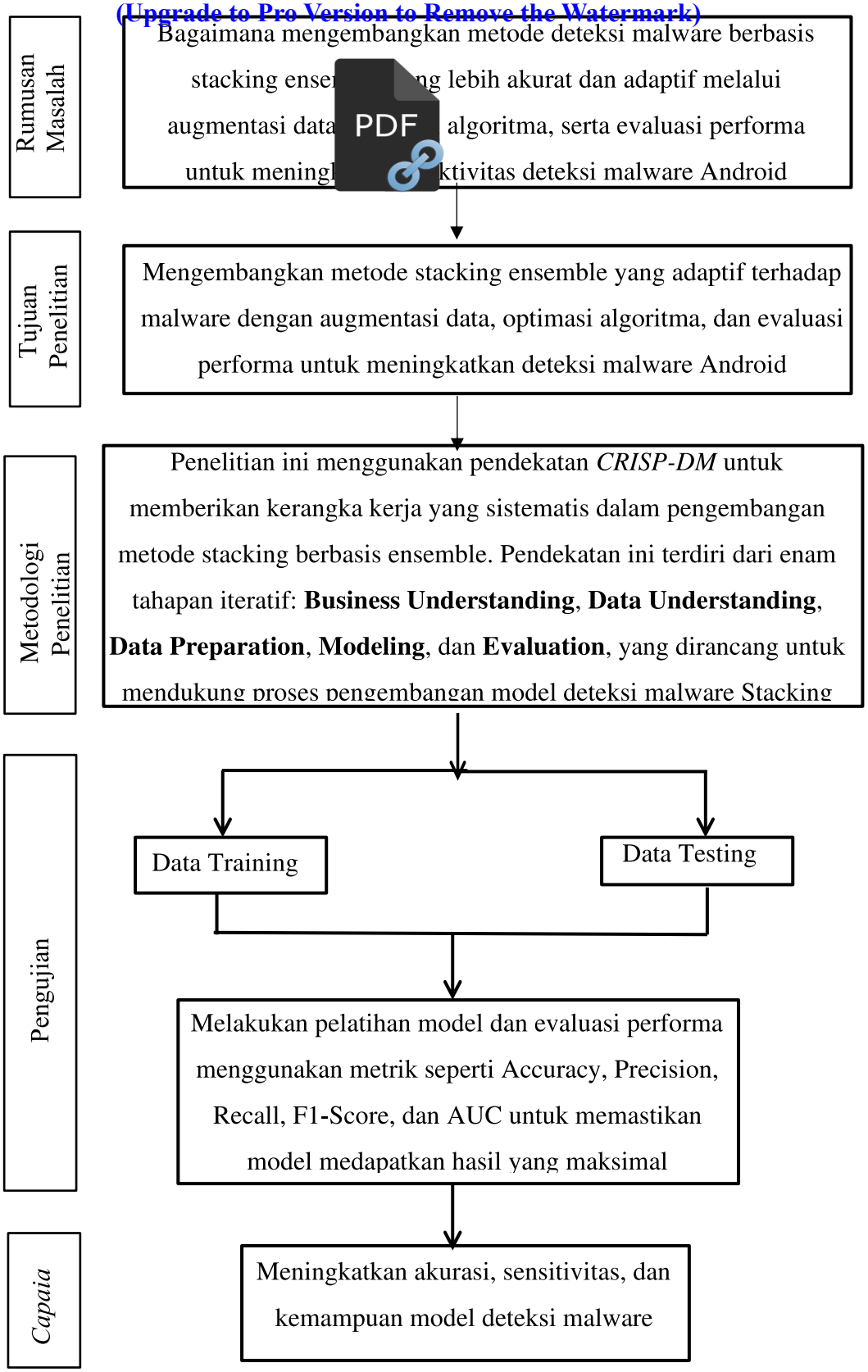
9) Penelitian keenam berjudul "*Enhanced Detection of Obfuscated Malware in Memory Dumps: A Machine Learning Approach for Advanced Cybersecurity*" oleh Md. Alamgir Hossain dan Md. Saiful Islam. Penelitian ini membahas deteksi *malware obfuscated* berbasis *memory dumps* dengan menggunakan SMOTE dan algoritma *ensemble* seperti *Gradient Boosting, Random Forest,* dan *Bagging*. Kesamaan dengan

penelitian saya adalah membahas deteksi malware, menggunakan metode augmentasi data *(SMO* n algoritma *ensemble.* Perbedaannya adalah penelitian ini tidak m an metode *stacking ensemble* atau *meta-classifier,* fokus pada *dumps*. Hasil penelitian menunjukkan bahwa penerapan *SMOTE* meningkatkan akurasi hingga 100% pada semua kategori *malware* [32].

10) Penelitian ketujuh berjudul "*Malware Detection and Classification in Android Application Using Simhash-Based Feature Extraction and Machine Learning*" oleh Wafaa Al-Kahla, Eyad Taqieddin, Ahmed S. Shatnawi, dan Rami Al-Ouran. Penelitian ini menggunakan *Simhash* untuk ekstraksi fitur dan *tuning hyperparameter Logistic Regression* menggunakan *Grid Search.* Kesamaan dengan penelitian saya adalah membahas deteksi *malware,* menggunakan algoritma *machine learning,* dan *tuning hyperparameter* untuk model tertentu *(Logistic Regression).* Perbedaannya adalah penelitian ini tidak menggunakan *metode ensemble* atau *stacking,* tidak ada pendekatan augmentasi data. Hasil penelitian menunjukkan bahwa *Grid Search pada Logistic Regression* menghasilkan performa terbaik, namun fokus pada efisiensi ekstraksi fitur menggunakan Simhash [3].

2.3 Kerangka Berpikir

Kerangka berpikir adalah alat yang membantu peneliti dalam merencanakan dan menyusun argumen terkait arah asumsi yang akan diambil. Dalam penelitian kuantitatif, kerangka berpikir biasanya berfokus pada penerimaan atau penolakan hipotesis. Sementara dalam penelitian naratif atau deskriptif, peneliti memulai dari data yang ada dan menggunakan teori sebagai landasan penjelasan. Hasil akhirnya adalah pembaruan pada suatu pernyataan atau hipotesis, seperti yang ditunjukkan pada gambar 2.5 di bawah ini [33].

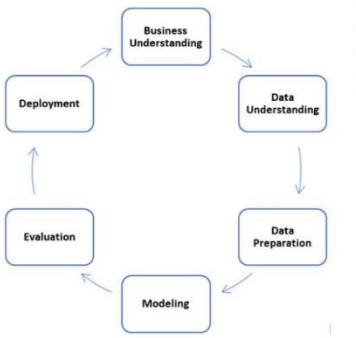


Gambar 2.5. Kerangka berpikir [33]

BAB III
METODOLOGI PENELITIAN

3.1 Metode Penelitian

Metodologi penelitian ini menggunakan pendekatan *Cross-Industry Standard Process for Data Mining* (CRISP-DM), yang terdiri dari enam tahap iteratif: *Business Understanding, Data Understanding, Data Preparation, Modeling, dan Evaluation*. Proses ini dipilih untuk memberikan kerangka kerja yang terstruktur dan mendalam dalam pengembangan metode deteksi malware berbasis stacking ensemble seperti pada gambar dibawah 3.1 dibawah ini [34].



Gambar 3.1. *Model CRISP-DM* [34]

Berikut adalah tahapan penelitian sesuai model CRISP-DM:

a. Business Understanding

   Pada fase pemahaman bisnis, penelitian ini bertujuan untuk mengembangkan *metode stacking berbasis ensemble* untuk deteksi *malware* Android dengan mengoptimalkan algoritma agar akurasi model meningkat dan *overfitting* dapat dihindari. Selain itu, dilakukan *tuning hyperparameter* menggunakan teknik *GridSearchCV* untuk menemukan parameter optimal, serta menggunakan teknik augmentasi data seperti *SMOTE* untuk menangani ketidakseimbangan kelas antara *malware* dan *goodmalware*

b. Data Understanding

Pada fase pemaham[illegible] dataset yang digunakan berasal dari *Tuandromd* yang be[illegible] *malware* dan *goodmalware*, di mana dilakukan analisis dist[illegible]as untuk memahami ketidakseimbangan antara kedua kelas tersebut. Analisis lebih lanjut dilakukan untuk mendeteksi nilai hilang dan anomali pada data, yang kemudian diatasi dengan metode *imputasi* yang sesuai. Selain itu, analisis korelasi antar fitur juga dilakukan untuk memilih fitur yang relevan dalam proses deteksi *malware*, menggunakan visualisasi heatmap untuk mempermudah pemahaman hubungan antar fitur.

c. Data Preparation

Pada fase pengolahan data, beberapa langkah penting diambil untuk menyiapkan data agar siap digunakan dalam model. Pertama, nilai hilang pada dataset ditangani dengan imputasi, menggunakan metode seperti rata-rata atau median sesuai kebutuhan. Selanjutnya, dilakukan *feature scaling* dengan *MinMaxScale*r untuk memastikan setiap fitur berada dalam rentang yang sama. Untuk mengatasi ketidakseimbangan kelas dalam dataset, teknik augmentasi data seperti *SMOTE* diterapkan untuk menghasilkan data sintetis yang membantu memperbaiki representasi kelas minoritas (*malware*).

d. Modeling

Pada fase pemodelan, dilakukan pelatihan model menggunakan beberapa base classifiers pada level 0 seperti *Random Forest*, *SVM*, *KNN*. Setiap base classifier dilatih menggunakan *cross-validation* untuk mengevaluasi performa model dan menghindari overfitting. Selain itu, dilakukan tuning hyperparameter untuk masing-masing model menggunakan GridSearchCV, guna mengoptimalkan kinerja. Setelah model base classifiers terbentuk, *stacking ensemble* diterapkan untuk menggabungkan hasil klasifikasi menggunakan *meta-classifier Logistic Regression* pada level 1 untuk menghasilkan prediksi akhir yang lebih akurat.

e. Evalution

Pada fase evaluasi, performa model diukur menggunakan teknik Confusion Matrix untu hitung metrik seperti akurasi, presisi, recall, dan F1-score. Metrik i untuk mengevaluasi sejauh mana model dapat mengklasifikasik re dan goodmalware dengan benar, serta memberikan gambaran tentang keseimbangan antara hasil positif dan negatif yang dihasilkan oleh model. Selain itu, Area Under Curve (AUC) dapat digunakan untuk menilai sejauh mana model dapat membedakan antara kelas malware dan goodmalware. Teknik ini memastikan bahwa model tidak hanya efektif dalam mendeteksi malware, tetapi juga memberikan wawasan yang mendalam mengenai performa keseluruhan model.

3.2 Metode Pengumpulan Data

Dalam penelitian ini menggunakan *dataset TUANDROMD* dari *Tezpur University* yang tersedia di UCI Machine Learning Repository. Dataset ini berisi 4.465 aplikasi Android**,** yang terdiri dari 3.565 aplikasi berbahaya (*malware*) dan 899 aplikasi aman (*goodware*)**.** Untuk membantu membedakan antara aplikasi berbahaya dan aman, dataset ini mencakup 178 fitur berbasis izin dan 186 fitur berbasis API, dengan total 242 atribut biner. Selain itu, aplikasi dalam dataset ini terbagi ke dalam 135 kategori dan mencakup 71 keluarga *malware*. Meskipun kaya akan informasi, dataset ini menghadapi masalah ketidakseimbangan data karena jumlah malware jauh lebih banyak dibandingkan aplikasi aman. Dataset ini digunakan untuk melatih model deteksi malware pada perangkat Android, memberikan wawasan penting untuk mengidentifikasi ancaman keamanan secara akurat.

3.3 Metode Analisa

Metode analisis berfokus pada evaluasi hasil model menggunakan teknik stacking berbasis ensemble. Tahapan analisis dilakukan untuk mengevaluasi kinerja model dan efektivitas pendekatan stacking dalam deteksi malware pada dataset yang tidak seimbang:

a. **Evaluasi Model:** Menggunakan beberapa metrik evaluasi seperti *akurasi*, *presisi*, *recall*, *F1-score*, dan *Area Under Curve (AUC)* untuk mengukur seberapa baik model dalam mendeteksi malware pada dataset yang tidak seimbang.

b. **Analisis Performa Base Classifiers:** Mengevaluasi kontribusi setiap algoritma dasar yang digunakan dalam stacking untuk memahami peran mereka dalam meningkatkan performa model.

c. **Evaluasi Hasil Stacking Ensemble:** Menganalisis performa meta-classifier (Logistic Regression) pada level 1 untuk mengukur kemampuan stacking ensemble dalam mengintegrasikan kekuatan base classifiers secara efektif.

3.4 Tempat dan Waktu Penelitian

3.4.1 Tempat Penelitian

Penelitian ini dilakukan di rumah penulis sendiri yaitu di Sp. PT. Sawit GSSl Dusun IV Desa Muara Kati Baru I

3.4.2 Waktu Penelitian

Estimasi pelaksanaan penelitian akan dimulai pada bulan September 2024, seperti yang terperinci dalam Tabel 3.1 Rencana ini mencakup jadwal kerja yang telah disusun untuk proyek penelitian yang diharapkan dimulai pada tanggal tersebut :

Tabel 3.1 Waktu Penelitian

Kegiatan	Waktu Penelitian 2024				
	Sep	Oktober	November	Desember	Januari
Analisa Permasalahan Penelitian					
Review jurnal terkait					
Pengajuan Judul					
Pengumpulan Data					
Bimbingan Proposal					
Pembuatan sistem					
Ujian Proposal					
Pengujian Sistem					
Bimbingan Skripsi					
Ujian Skripsi					
Revisi Skripsi					

3.5 Alat Dan Bahan

a. Alat

Adapun alat yang digu  am penelitian ini sebagai berikut :

1) Perangkat Keras
   a) Laptop Acer Aspire ES 14
   b) Printer Cannon MG2570S
   c) Handphone Vivo Y15S
2) Perangkat Lunak
   a) Microsoft Word
   b) Google Coolab

b. Bahan

Adapun Bahan yang digunakan dalam Penelitian ini sebagai berikut :

1) Tinta Printer
2) Kertas A4 70g

3.6 Metode Pengujian dan Pengolahan Data

a. Metode Pengujian

Dalam tahap pengujian, metode stacking berbasis ensemble diterapkan dalam kerangka kerja CRISP-DM untuk menjelaskan langkah-langkah yang terstruktur. Proses ini mencakup pembagian dataset, pengolahan data, pelatihan model dengan stacking berbasis ensemble, tuning hyperparameter, hingga evaluasi performa model menggunakan metrik yang relevan. CRISP-DM digunakan sebagai panduan kerangka alur, memastikan setiap tahapan mendukung pengembangan metode deteksi malware yang efektif, termasuk pada kondisi dataset yang tidak seimbang.

Adapun tahapan-tahapannya sebagai berikut:

1) Pembagian Dataset untuk Pelatihan dan Pengujian:
   a) Dataset Tuandromd akan displit menjadi 80 % training dan 10% validasi dan 10 % testing Dataset terdiri dari dua kelas utama: *malware* dan *goodmalware.*

b) *Cross-validation:* Dataset dibagi menjadi lima bagian (folds), di mana empat bagi[illegible]nakan untuk pelatihan dan satu bagian untuk pengujian. Prose[illegible]ang sebanyak lima kali, sehingga setiap bagian akan berp[illegible]gai data uji satu kali.

2) Pengujian Model dengan Stacking Ensemble:
   a) Level 0 (Base Classifiers): Model dilatih menggunakan beberapa algoritma dasar, seperti *Random Forest*, *Catboost*, *AdaBoost*, *SVM*, *Logistic Regression*, *KNN*.
   b) Stacking pada level 0 : Prediksi dari setiap base classifier pada Level 0 digabungkan menjadi fitur baru sebagai input untuk meta-classifier.
   c) Level 1 (Meta-classifier): Logistic Regression digunakan sebagai meta-classifier untuk mengolah input dari Level 0 dan menghasilkan prediksi akhir yang lebih akurat.

3) *Tuning Hyperparameter* dengan *GridSearchCV*

   Setelah model dilatih, parameter dari setiap base classifier dioptimalkan menggunakan *GridSearchCV* untuk mendapatkan kombinasi parameter terbaik. Teknik ini berjalan seiring dengan proses cross-validation, memastikan setiap iterasi menghasilkan model yang lebih baik.

4) Evaluasi Model:

   Model diuji menggunakan confusion metrik[14] berikut:
   a) Accuracy adalah nilai yang digunakan untuk mengukur seberapa tepat sistem dalam mengklasifikasikan data dengan benar, yang dihitung menggunakan rumus pada formula 1 dibawah ini.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{FP+FNTP+TN}}$$

b) Precision (presisi) adalah nilai yang mengukur jumlah data positif yang diklasifikasikan dengan benar, dibagi dengan total data yang diklasifikasikan sebagai positif, yang dihitung menggunakan rumus pada formula 2 dibawah ini.

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

c) Recall adalah nilai yang digunakan untuk mengukur persentase data kategori positif yang berhasil diklasifikasikan dengan benar oleh sistem, yang dihitung menggunakan rumus pada formula 3 dibawah ini.

$$\text{Recall} = \frac{TP}{TP+FN}$$

d) F1-Score adalah nilai rata-rata harmonik dari presisi dan recall. Nilai terbaik F1-Score adalah 1.0, sementara nilai terburuknya adalah 0. Jika F1-Score menunjukkan nilai yang baik, ini mengindikasikan bahwa metode klasifikasi yang digunakan memiliki presisi dan recall yang seimbang, yang dihitung menggunakan rumus pada formula 4 dibawah ini.

$$F1 - \text{score} = \frac{2 \text{ x Precision x Recall}}{\text{Precision+Recall}}$$

e) Area Under the Curve (AUC) adalah ukuran dari area di bawah kurva ROC. Semakin besar area tersebut, semakin baik model klasifikasi yang diusulkan. ROC merupakan gambaran grafis dari hubungan antara sensitivitas dan spesifisitas. Nilai AUC ini

menunjukkan seberapa baik model dalam membedakan antara kelas yang berbeda [26

***b.* Pengolahan Data**

Proses pengolahan data mengikuti tahapan dalam metodologi *CRISP-DM* untuk menyiapkan dataset agar siap digunakan untuk pelatihan dan pengujian model. Langkah-langkah yang dilakukan adalah sebagai berikut:

1) Dataset:

   Dataset *TUANDROMD* terdiri dari 4.465 contoh (3.565 *malware* dan 899 *goodmalware*) dengan 241 atribut berbasis izin dan API.

2) Prepocessing

   a) Pembersihan Data:

      (1) Penanganan Nilai Hilang: Nilai hilang dalam dataset diidentifikasi dan diisi menggunakan teknik imputasi seperti rata-rata atau median.

      (2) Penyaringan Data: Data yang tidak relevan atau duplikat dihapus dari dataset.

   b) Feature Engineering:

      (1) Seleksi Fitur: Fitur yang relevan untuk deteksi malware dipilih, dan fitur yang tidak relevan dihapus.

      (2) *Feature Scaling: MinMaxScaler* digunakan untuk menormalkan fitur agar berada dalam rentang yang sama, memastikan fitur memiliki bobot yang setara dalam model.

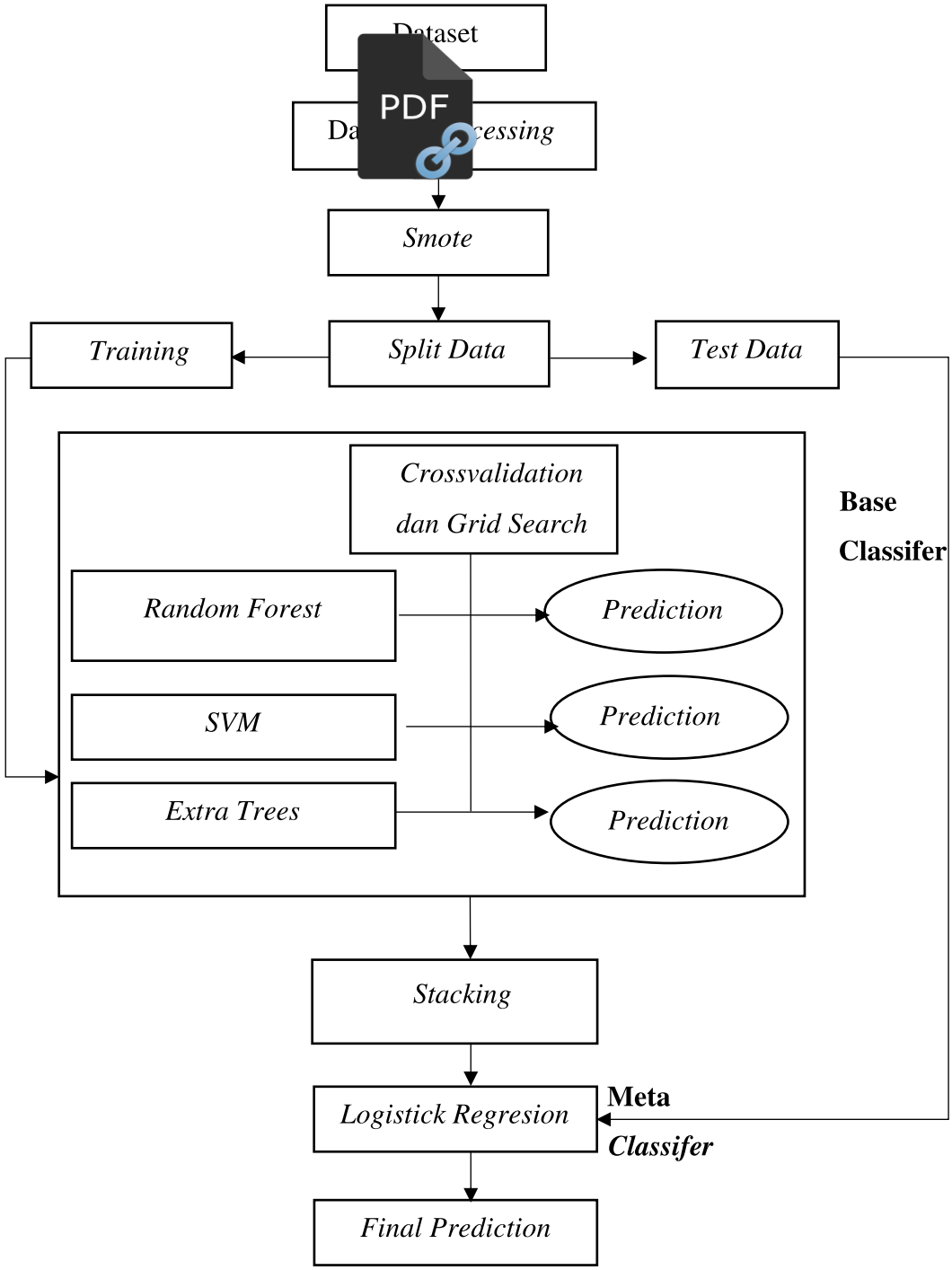
3) Augmentasi Data:

   SMOTE (Synthetic Minority Over-sampling Technique): SMOTE diterapkan untuk mengatasi ketidakseimbangan kelas dalam dataset dengan mensintesis sampel baru untuk kelas malware, sehingga distribusi data menjadi lebih seimbang.

4) Persiapan Dataset:

Dataset akan displi  i data latih dan data uji kemudian crossvalidation akan n pada data training berdasarkan skema *5-fold cross-validation*, seperti yang dijelaskan di metode pengujian.

Alur diagram tahapan modeling system pada penelitian ini dapat dilihat di gambar 3.2 dibawah ini.



Gambar 3.2. Tahapan Modeling

BAB IV
HASIL PEN[illegible]AN DAN PEMBAHASAN

4.1 Gambaran Umum (Temp[illegible]tian)

Penelitian ini dilakukan sepenuhnya di rumah penulis dengan memanfaatkan perangkat keras dan perangkat lunak yang mendukung proses analisis data. Penulis menggunakan laptop Acer Aspire ES 14 untuk menjalankan program dan analisis, serta Google Colab sebagai platform utama untuk pengolahan data. Dataset yang digunakan, yaitu TUANDROMD, diakses secara online dari repositori yang disediakan oleh Tezpur University. Dataset ini terdiri dari total 4.465 sampel, yang mencakup 3.565 malware dan 899 goodware.

Penelitian ini mengikuti kerangka kerja CRISP-DM (Cross-Industry Standard Process for Data Mining). Pendekatan ini mencakup enam tahapan utama: memahami kebutuhan bisnis, memahami data, mempersiapkan data, membangun model, mengevaluasi model, dan menerapkan hasil. Setiap tahapan dirancang untuk mendukung pengembangan model deteksi malware berbasis stacking ensemble

4.2 Hasil Penelitian

Penelitian ini bertujuan mengembangkan metode deteksi malware berbasis *stacking ensemble* dengan pendekatan augmentasi data menggunakan SMOTE pada dataset tidak seimbang. Sistem yang dibangun mengintegrasikan beberapa algoritma pembelajaran mesin sebagai *base classifiers*, seperti Random Forest, Support Vector Machine (SVM), dan Extra Trees. Meta-classifier yang digunakan adalah Logistic Regression. Teknik augmentasi data diterapkan menggunakan SMOTE untuk menyeimbangkan distribusi data antara kelas malware dan goodware. Sistem ini diuji menggunakan validasi silang 5-fold dan dioptimalkan dengan *GridSearchCV* untuk memperoleh kombinasi parameter terbaik. Tahapan sistem dimulai dari pengumpulan data, preprocessing, augmentasi data, pelatihan model, hingga evaluasi performa model menggunakan metrik akurasi, presisi, recall, F1-score, dan AUC.

4.2.1 Dataset Penelitian

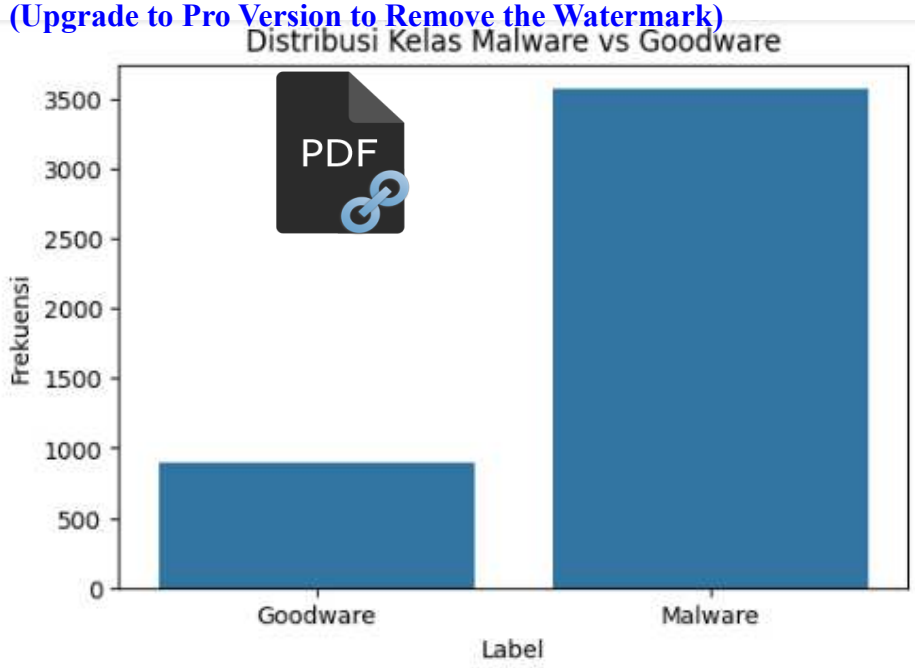
a. Dataset

Data yang digunakan te[PDF]es oleh Tezpur University 1-214: Fitur berbasis izin 215-241: F[PDF]asis API, Dataset TUNADROMD berisi 4465 contoh dan 241 atribut. Atribut target untuk klasifikasi adalah kategori (malware vs goodware). Isi dataset nya dapat dilihat pada gambar 4.1 disana ada 2 kelas yaiut malware dan good malware serta terdapat juga data nan (Not a Number) atau nilai yang hilang atau tidak valid dan akan diatasi dengan fitur numerik dengan median setelah itu akan dilakukan heatmap matrix korelasi dengan memilih hanya kolom yang bertipe data numerik seperti float64 dan int64 sehingga terdapat lah 31 buah seperti pada gambar 4.2 selanjutnya akan dilakukan distribusi label seperti pada gambar 4.3 untuk melihat jumlah sample malware dan good malware nya.

TelephonyManager;->getSimOperatorName	Landroid/telephony/TelephonyManager;->getSimCountryIso	Landroid/telephony/TelephonyManager;->getSimSerialNumber	Lorg/apache/http/impl/client/DefaultHttpClient;->execute	Label
0.0	0.0	0.0	1.0	malware
0.0	1.0	0.0	0.0	malware
0.0	0.0	0.0	0.0	malware
0.0	1.0	0.0	0.0	malware
0.0	0.0	0.0	0.0	malware
0.0	0.0	0.0	0.0	goodware
0.0	0.0	0.0	0.0	goodware
0.0	1.0	0.0	0.0	goodware
0.0	0.0	0.0	0.0	goodware
1.0	1.0	0.0	1.0	goodware
NaN	NaN	NaN	NaN	NaN

Gambar 4.1 Informasi Dataset

Gambar 4.2. Visualisasi Heatmap



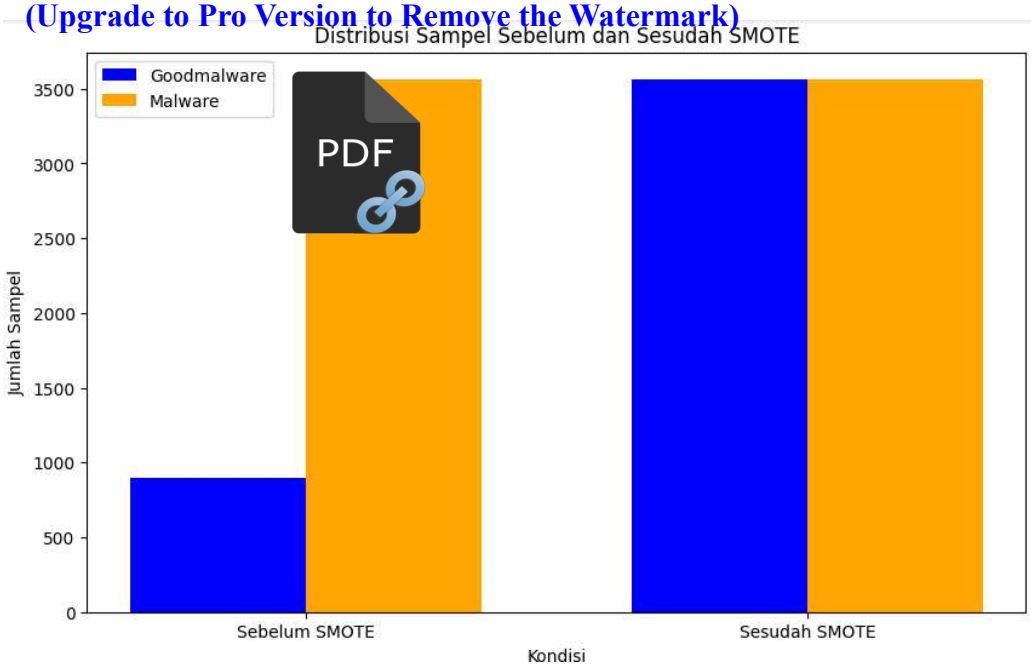

```
Distribusi Label:
Label
malware     3565
goodware     899
Name: count, dtype: int64
Persentase:
Label
malware     79.843225
goodware    20.134378
Name: count, dtype: float64
```

Gambar 4.3 Distribusi Dataset

b. Processing

Tahapan preprocessing meliputi:

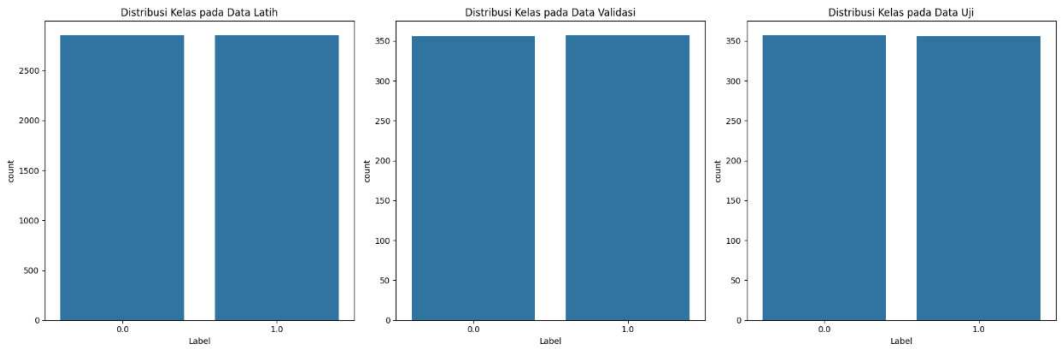
1) **Pembersihan Data:** Menghapus data duplikat dan menangani nilai yang hilang menggunakan imputasi rata-rata.
2) **Feature Scaling:** Menggunakan *MinMaxScaler* untuk memastikan semua fitur berada dalam rentang 0 hingga 1.
3) **Augmentasi Data:** Diterapkan teknik SMOTE untuk menghasilkan sampel sintetis dari kelas minoritas (goodware). Jumlah sample sebelum smote label 0 berjumlah 899 dan label 1.0 berjumlah 3565. Setelah dilakukan smote label 0 berjumlah 3565 dan label 1.0 berjumah 3566 seperti pada gambar 4.4 dibawah ini.



Gambar 4.4 Distribusi Sample Sebelum dan Sesudah SMOTE

4.2.2 Split Data

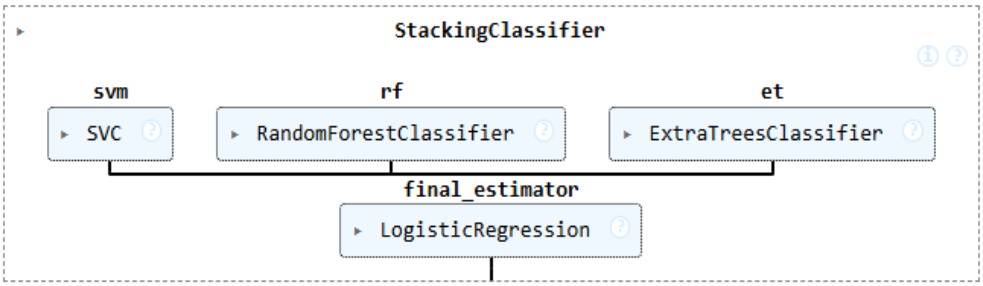
Setelah dilakukan Smote maka akan dilakukan pembagian Dataset atau split data menjadi 3 buah data yaitu data train berjumlah 5.704 digunakan untuk melatih model dan menyesuaikan parameter model, data validasi berjumlah 713 data digunakan untuk tuning hyperparameter dan mencegah overfitting selama pelatihan dan data test berjumlah 714 data digunakan untuk menguji kinerja akhir model setelah pelatihan selesai berikut diagram split data seperti pada gambar 4.5 dibawah ini.



Gambar 4.5 Visualisasi Split Dataset

4.2.3 Pengujian Model

a. Model Base Classifier d king Base classifiers yang digunakan adalah Random Forest, Suppor Machine (SVM), dan Extra Trees. Model stacking menggunakan Regression sebagai meta-classifier untuk menggabungkan prediksi dari base classifiers. Proses ini dirancang untuk mengoptimalkan performa dengan memanfaatkan kekuatan masing-masing algoritma. Visualisasi terkait ditampilkan pada Gambar 4.6



Gambar 4.6 Visualisasi Model

b. Tuning Hyperparameter

Didalam baseclasifer masing masing algoritma akan dilakukan tuning hyperparameter dan dievaluasi dengan crossvalidation untuk menentukan base parameter terbaik seperti pada gambar 4.7 dibawah ini

```
Best Parameters for svm: {'C': 10, 'gamma': 'scale', 'kernel': 'rbf'}
Best Parameters for rf: {'criterion': 'entropy', 'max_depth': 20, 'max_features': 'sqrt', 'n_estimators': 140}
Best Parameters for et: {'criterion': 'gini', 'max_depth': None, 'max_features': 'log2', 'n_estimators': 200}
```

Gambar. 4.7 Best Parameter

Hasil tuning hyperparameter menggunakan GridSearchCV dengan evaluasi metrik ROC AUC menghasilkan parameter terbaik untuk masing-masing model sebagai berikut:

1) Support Vector Machine (SVM):

Parameter terbaik yang ditemukan adalah C=10, gamma='scale', dan kernel='rbf'. Nilai C=10 menunjukkan regularisasi yang seimbang antara overfitting dan underfitting, sementara gamma='scale'

memungkinkan kernel RBF untuk beradaptasi dengan data secara lebih optimal be an nilai inverse jumlah fitur. Penggunaan kernel RBF mem kemampuan untuk menangkap pola non-linear dalam data

2) Random Forest (RF):

Parameter terbaik untuk Random Forest adalah criterion='entropy', max_depth=20, max_features='sqrt', dan n_estimators=140. Penggunaan criterion='entropy' meningkatkan pemilihan split berdasarkan pengurangan informasi, sedangkan max_depth=20 memastikan pembelajaran tidak terlalu mendalam untuk mencegah overfitting. Pemilihan max_features='sqrt' meningkatkan efisiensi pada dataset dengan jumlah fitur besar, dan n_estimators=140 memberikan stabilitas prediksi dengan jumlah pohon yang cukup besar.

3) Extra Trees (ET):

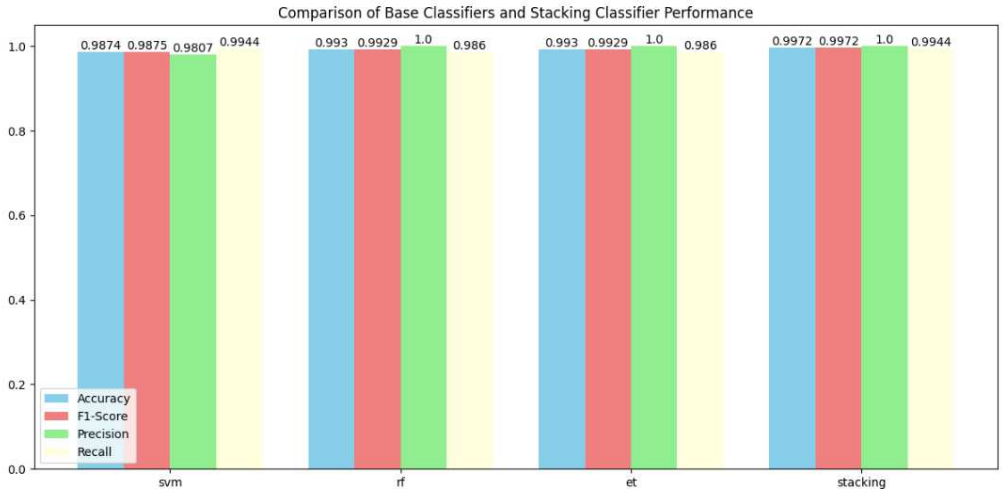
Parameter terbaik untuk Extra Trees adalah criterion='gini', max_depth=None, max_features='log2', dan n_estimators=200. Penggunaan criterion='gini' berfokus pada memaksimalkan kejelasan kelas, sedangkan max_depth=None memungkinkan pohon tumbuh hingga selesai tanpa batas kedalaman. Pemilihan max_features='log2' menyeimbangkan variasi antar pohon, dan n_estimators=200 memberikan kestabilan prediksi dengan jumlah pohon yang lebih banyak.

Secara keseluruhan, hasil tuning ini memastikan bahwa setiap model dioptimalkan untuk menangkap pola dari dataset secara spesifik, sehingga diharapkan meningkatkan performa klasifikasi secara keseluruhan.

4.2.4 Evaluasi

a) Hasil Prediksi base cla lan stacking

Model akan dievaluasi urasi, presisi, f1 score, dan recal. Metrix ini digunakan untuk meng inerja model deteksi secara lebih akurat. Hasil prediksi dapat dilihat pada gambar 4.8.



Gambar 4.8 Hasil Base Classifer dan Stacking

Berdasarkan gambar 4.8 hasil evaluasi model dapat dijelaskan sebagai berikut:

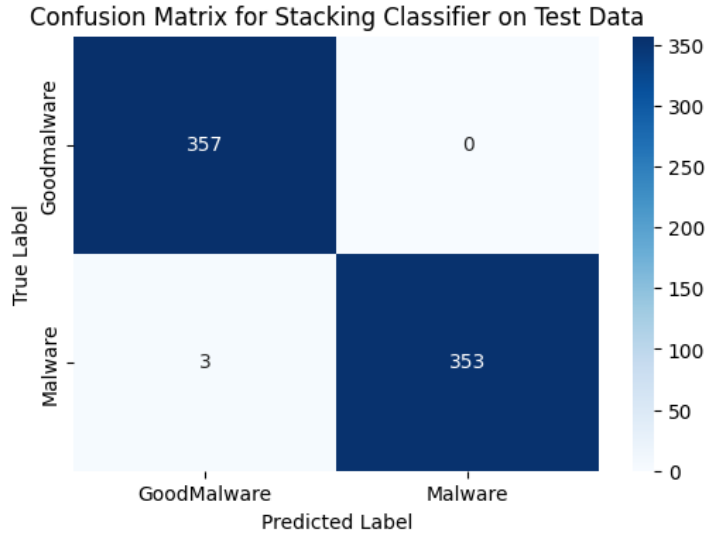
1) Akurasi: Model Stacking memiliki akurasi sebesar 0.9958 untuk kelas malware dan goodware, yang menunjukkan bahwa model ini mampu memprediksi dengan benar 99.58% dari semua data. Model Extra Trees (ET) mencapai akurasi 0.9944, yang berarti 99.44% dari prediksi benar. Model Random Forest (RF) memiliki akurasi 0.9930, yang menunjukkan 99.30% data diprediksi dengan benar. Sementara itu, model SVM memiliki akurasi 0.9748, menunjukkan tingkat keberhasilan prediksi sebesar 97.48%.
2) F1-Score : Model Stacking juga memiliki F1-Score tertinggi, yaitu 0.9958, menunjukkan keseimbangan yang sangat baik antara presisi dan recall. Model ET memiliki F1-Score 0.9944, sedikit di bawah Stacking. Model RF mencapai F1-Score 0.9929, sedangkan SVM memperoleh nilai 0.9750.

3) Presisi: Model Stacking, ET, dan RF mencapai nilai presisi tertinggi sebesar 1.0000, yan[illegible]ti semua prediksi positif yang dibuat oleh model adalah benar[illegible]VM memiliki presisi 0.9643, yang berarti dari semua prediksi[illegible]6.43% adalah benar.
4) Recall: Recall tertinggi dimiliki oleh Stacking, yaitu 0.9916, menunjukkan kemampuan model untuk mendeteksi sebagian besar data positif. Model ET memiliki recall 0.9888, diikuti oleh RF dengan recall 0.9860, dan SVM dengan recall 0.9860.

Secara keseluruhan, model Stacking menunjukkan performa terbaik dalam semua metrik evaluasi, dengan prediksi yang sangat akurat, presisi yang sempurna, dan kemampuan tinggi dalam mendeteksi data positif.

a) Confusion Matrix

Pada Proses evaluasi model ini akan menggunakan cofusion matrix untuk mengukur performa dari model pada dataset pengujian. Confusion matrix akan memberikan infomasi mengenai jumlah prediksi yang benar dan salah yang dilakukan model pada kelas malware dan goodmalware. Hasil evaluasi dapat dilihat pada gambar 4.9.



Gambar 4.9 Confusion Matrix Data Test

Berdasarankan Confusion Matrix tersebut :

1) Prediksi benar : pada kelas malware sebanyak 357 dan 353 pada goodmalware
2) Prediksi salah : pada kelas malware 0 gagal terdeteksi dan 3 pada goodmalware

Dengan data prediksi yang salah prediksi sangat sedikit yaitu hanya terjadi pada data goodmalware yang berjumlah 3 data. Hal ini menunjukan bahwa mdel sudah sangat baik dalam mendetksi malware maupun goodmalware hal ini masih bisa ditingkatkan agar bisa tedeteksi lebih baik lagi dengan cara, seperti penambahan jumlah data latih, tuning hyperparameter dan crossvalidationnya.

4.3 Pembahasan

Berdasarkan hasil pengujian dan analisis yang dilakukan, dapat disimpulkan bahwa:

1) Metode stacking ensemble dengan augmentasi data SMOTE dan tuning hyperparameter mampu meningkatkan akurasi deteksi malware pada dataset tidak seimbang. Model yang dikembangkan memiliki tingkat akurasi, presisi, recall, F1-score, dan AUC yang tinggi, menunjukkan kemampuan yang andal dalam mendeteksi malware Android.
2) Pendekatan yang diterapkan berhasil mengatasi masalah ketidakseimbangan data serta meningkatkan ketahanan model terhadap variasi data.

BAB V
KESIM N DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai pengembangan dan evaluasi metode stacking berbasis ensemble untuk deteksi malware dengan pendekatan augmentasi data pada dataset tidak seimbang, diperoleh beberapa kesimpulan sebagai berikut:

1) Metode stacking ensemble yang dikembangkan berhasil meningkatkan akurasi deteksi malware pada perangkat Android dengan mengintegrasikan optimasi algoritma dan tuning hyperparameter. Kombinasi beberapa algoritma sebagai base classifiers dan Logistic Regression sebagai meta-classifier menghasilkan model dengan performa tinggi, yakni akurasi sebesar 99,58%, presisi 100%, recall 99,16%, F1-score 99,58%, dan AUC 99,5%. Proses tuning hyperparameter menggunakan GridSearchCV berkontribusi signifikan terhadap pemilihan parameter optimal untuk setiap algoritma dasar, sehingga meningkatkan kemampuan generalisasi model.
2) Teknik augmentasi data menggunakan Synthetic Minority Oversampling Technique (SMOTE) terbukti efektif dalam mengatasi ketidakseimbangan dataset antara kelas malware dan goodware. SMOTE meningkatkan representasi kelas minoritas tanpa mengurangi akurasi dalam mendeteksi kelas mayoritas, sehingga model mampu mendeteksi malware maupun goodware secara efektif. Ketahanan model terhadap variasi data juga teruji pada dataset uji eksternal, menunjukkan generalisasi yang baik dalam domain yang sama.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, penulis memberikan beberapa saran untuk penelitian selanjutnya:

a. Pengembangan Metode Deteksi yang Lebih Canggih

1) Penggunaan Dataset yang Lebih Beragam

Disarankan untuk menggunakan dataset yang lebih besar dan mencakup jenis malware yang lebih beragam agar hasil yang diperoleh dapat lebih representatif terhadap kondisi nyata.

2) Eksplorasi Teknik Deteksi Baru

Selain metode stacking ensemble, penelitian masa depan dapat mengeksplorasi penggunaan metode deep learning atau hybrid ensemble untuk meningkatkan akurasi lebih lanjut.

b. Pengembangan Metode Deteksi yang Lebih Canggih

1) Penerapan pada Sistem Nyata

Penelitian selanjutnya diharapkan dapat mengimplementasikan model yang dikembangkan pada sistem deteksi malware nyata di perangkat Android. Hal ini bertujuan untuk mengevaluasi performa model dalam kondisi operasional yang sesungguhnya.

2) Pengujian pada Berbagai Platform

Demi memperluas cakupan penelitian, metode serupa dapat diterapkan pada platform lain seperti iOS, Windows, atau sistem operasi lainnya. Dengan demikian, model yang dikembangkan dapat memiliki cakupan dan dampak yang lebih luas.

3) Pengembangan Teknik Augmentasi Data Lainnya

Dianjurkan untuk mengeksplorasi teknik augmentasi data lain selain SMOTE, seperti ADASYN atau metode berbasis generatif (GAN), untuk mengatasi ketidakseimbangan dataset secara lebih efektif.

4) Penggunaan Fitur Tambahan untuk Deteksi Malware

Penelitian mendatang dapat mempertimbangkan penggunaan fitur tambahan, seperti pol ku aplikasi, analisis lalu lintas jaringan, atau analisis metadata ap tuk meningkatkan kemampuan deteksi terhadap malware ya ompleks dan canggih.

Dengan mengacu pada saran-saran tersebut, diharapkan penelitian di masa depan dapat mengembangkan metode deteksi malware yang lebih canggih, efektif, dan relevan dengan kebutuhan keamanan siber.

DAFTAR PUSTAKA

[1] M. Zyout, R. Shatnawi, a adat, "Malware classification approaches utilizing binary and text e f permissions," *Int. J. Inf. Secur.*, vol. 22, no. 6, pp. 1687–1712, 2023, doi: 10.1007/s10207-023-00712-z.

[2] S. Sharma, R. Kumar, and C. Rama Krishna, "A survey on analysis and detection of Android ransomware," *Concurr. Comput. Pract. Exp.*, vol. 33, no. 16, pp. 1–24, 2021, doi: 10.1002/cpe.6272.

[3] W. Al-Kahla, E. Taqieddin, A. S. Shatnawi, and R. Al-Ouran, "Malware Detection and Classification in Android Application using Simhash-Based Feature Extraction and Machine Learning," *IEEE Access*, vol. 12, no. October, pp. 174255–174273, 2024, doi: 10.1109/ACCESS.2024.3501277.

[4] Rafrasta, F. A. Ra, C. Supriyanto, C. Paramita, and Y. P. Astuti, "Deteksi Malware menggunakan Metode Stacking berbasis Ensemble," *J. Inform. J. Pengemb. IT*, vol. 8, no. 1, pp. 11–16, 2023, doi: 10.30591/jpit.v8i1.4606.

[5] E. B. Soritua and D. N. Utama, "Enhancing Android Malware Detection Through Ensemble Stacking Classifiers and Regularization-Based Feature Selection," *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 3, pp. 801–811, 2024.

[6] A. Albin Ahmed, A. Shaahid, F. Alnasser, S. Alfaddagh, S. Binagag, and D. Alqahtani, "Android Ransomware Detection Using Supervised Machine Learning Techniques Based on Traffic Analysis," *Sensors*, vol. 24, no. 1, pp. 1–21, 2024, doi: 10.3390/s24010189.

[7] T. Hastie, R. Tibshirani, G. James, and D. Witten, "An Introduction to Statistical Learning, Springer Texts," *Springer Texts*, vol. 102, p. 618, 2006.

[8] I. Muhamad Malik Matin, "Hyperparameter Tuning Menggunakan GridsearchCV pada Random Forest untuk Deteksi Malware," *Multinetics*, vol. 9, no. 1, pp. 43–50, 2023, doi: 10.32722/multinetics.v9i1.5578.

[9] M. P. Pulungan, A. Purnomo, and A. Kurniasih, "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Kepribadian MBTI Menggunakan Naive Bayes Classifier," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 7, pp. 1493–1502, 2023, doi: 10.25126/jtiik.1077989.

[10] Sharipuddin, R. S. Putra, M. F. Aulia, S. A. Maulana, and P. A. Jusia, "Android Security: Malv tection with Convolutional Neural Network and Feature Analysis," *M en. Comput. Sci.*, vol. 1, no. 1, pp. 7–13, 2023, doi: 10.62205/mjg

[11] A. F. Muhtadi and A. Almaarif, "Analysis of Malware Impact on Network Traffic using Behavior-based Detection Technique," *Int. J. Adv. Data Inf. Syst.*, vol. 1, no. 1, pp. 17–25, 2020, doi: 10.25008/ijadis.v1i1.14.

[12] Y. I. Rizqony, D. R. Akbi, and F. D. S. Sumadi, "Analisis Karakteristik Malware Joker Berdasarkan Fitur Menggunakan Metode Statik Pada Platform Android," *J. Repos.*, vol. 2, no. 10, pp. 1368–1379, 2020, doi: 10.22219/repositor.v2i10.1145.

[13] R. Hidayat, M. Koprawi, and P. Ferdiansyah, "Implementasi Neural Network 1-Dimensi Dalam Identifikasi Malware Android," *J. SAINTIKOM (Jurnal Sains Manaj. Inform. dan Komputer)*, vol. 23, no. 1, p. 252, 2024, doi: 10.53513/jis.v23i1.9587.

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

[15] J. Al Amien, Yoze Rizki, and Mukhlis Ali Rahman Nasution, "Implementasi Adasyn Untuk Imbalance Data Pada Dataset UNSW-NB15 Adasyn Implementation For Data Imbalance on UNSW-NB15 Dataset," *J. CoSciTech (Computer Sci. Inf. Technol.*, vol. 3, no. 3, pp. 242–248, 2022, doi: 10.37859/coscitech.v3i3.4339.

[16] A. K. Putri and Hari Suparwito, "Uji Algoritma Stacking Ensemble Classifier pada Kemampuan Adaptasi Mahasiswa Baru dalam Pembelajaran Online," *KONSTELASI Konvergensi Teknol. dan Sist. Inf.*, vol. 3, no. 1, pp. 1–12, 2023, doi: 10.24002/konstelasi.v3i1.7009.

[17] M. Rijal, A. A. Ilham, and A. W. Paundu, "Evaluasi Algoritma Klasifikasi dengan Berbagai Metode Seleksi Fitur untuk Mendeteksi Aktivitas Trojan," *J. Pekommas*, vol. 7, no. 2, pp. 85–98, 2022, doi: 10.56873/jpkm.v7i2.4842.

[18] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applicatio Prospects," *IEEE Access*, vol. 10, no. September, pp. 99129–9 2, doi: 10.1109/ACCESS.2022.3207287.

[19] F. Aziz, "Klasifikasi A anusia menggunakan metode Ensemble Stacking berbasis Smartphone," *J. Syst. Comput. Eng.*, vol. 1, no. 2, pp. 53–58, 2021, doi: 10.47650/jsce.v1i2.171.

[20] B. M. Karomah, "Penerapan Metode Stacking Dalam Mengklasifikasikan Penderita Penyakit Diabetes," *J. Publ. Ilmu Komput. dan Multimed.*, vol. 1, no. 3, pp. 188–194, 2022, doi: 10.55606/jupikom.v1i3.522.

[21] M. sigit T. Pamungkas, "Optimasi Performa Algoritma Naive Bayes Untuk Deteksi Malware pada Sistem Operasi Android dengan Particle Swarm Optimization," pp. 1–87, 2023.

[22] V. Issue and M. Hal, "Jurnal Litbang Edusaintech (JLE)," vol. 3, no. 1, pp. 33–41, 2022.

[23] E. KAVALCI YILMAZ and H. BAKIR, "Hyperparameter Tunning and Feature Selection Methods for Malware Detection," *Politek. Derg.*, vol. 27, no. 1, pp. 343–353, 2024, doi: 10.2339/politeknik.1243881.

[24] Togu Novriansyah Turnip, Chatrine Febryanti Manurung, Yogi Septian Lubis, and Rachel Gultom, "Klasifikasi Malware Android Aplikasi Menggunakan Random Forest Berdasarkan Fitur Statik," *Tek. Inform. dan Sist. Inf.*, vol. 10, no. 1, pp. 926–936, 2023, [Online]. Available: blob:https://jurnal.mdp.ac.id/b5c3fbd3-cbb1-4677-a16b-2c5300377815

[25] A. Ramadhan, L. Lindawati, and M. M. Rose, "Komparasi Algoritma Neural Network dan K-Nearest Neighbor Dalam Mendeteksi Malware Android," *Build. Informatics, Technol. Sci.*, vol. 5, no. 1, pp. 191–199, 2023, doi: 10.47065/bits.v5i1.3538.

[26] R. B. Hadiprakoso, W. R. Aditya, and F. N. Pramitha, "Analisis Statis Deteksi Malware Android Menggunakan Algoritma Supervised Machine Learning," *Cyber Secur. dan Forensik Digit.*, vol. 5, no. 1, pp. 1–5, 2022, doi: 10.14421/csecurity.2022.5.1.3116.

[27] N. Aslam *et al.*, "Explainable Classification Model for Android Malware

Analysis Using API and Permission-Based Features," *Comput. Mater. Contin.*, vol. 76, 3, pp. 3167–3188, 2023, doi: 10.32604/cmc.2023.039

[28] N. Wichitaksorn, Y. Kar Zhang, "Random feature selection using random subspace logistic regression," *Expert Syst. Appl.*, vol. 217, no. April 2022, p. 119535, 2023, doi: 10.1016/j.eswa.2023.119535.

[29] A. Wajahat *et al.*, "Outsmarting Android Malware with Cutting-Edge Feature Engineering and Machine Learning Techniques," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 651–673, 2024, doi: https://doi.org/10.32604/cmc.2024.047530.

[30] A. Wajahat *et al.*, "An effective deep learning scheme for android malware detection leveraging performance metrics and computational resources," *Intell. Decis. Technol.*, vol. 18, pp. 1–23, Feb. 2024, doi: 10.3233/IDT-230284.

[31] A. Putra Wijaya and H. Santoso, "Komparasi Performansi Algoritma Naive Bayes dan Logistic Regression pada Malware Android," *J. INTEK*, vol. 4, no. 2, pp. 31–40, 2021.

[32] M. A. Hossain and M. S. Islam, "Enhanced detection of obfuscated malware in memory dumps: a machine learning approach for advanced cybersecurity," *Cybersecurity*, vol. 7, no. 1, 2024, doi: 10.1186/s42400-024-00205-z.

[33] A. Z. Syahputri, F. Della Fallenia, and R. Syafitri, "Kerangka berfikir penelitian kuantitatif," *Tarb. J. Ilmu Pendidik. dan Pengajaran*, vol. 2, no. 1, pp. 160–166, 2023.

[34] "which has a scalable exercise of sustained capability to detect malicious network A SYSTEMATIC REVIEW OF DATA MINING AND MACHINE," no. August 2024.

KEPUTUSAN
DEKAN FAKULTAS ILMU TEKNIK
UNIVERSITAS BINA INSAN
Nomor: 0[illegible]/UNIV.BI/FIT.2/SK/2024

Tentang

DOSEN PEMBIMBING SKRIPSI
PROGRAM STUDI REKAYASA SISTEM KOMPUTER TAHUN AKADEMIK 2024/2025
UNIVERSITAS BINA INSAN

DEKAN FAKULTAS ILMU TEKNIK
UNIVERSITAS BINA INSAN

Menimbang :
a. Bahwa untuk kelancaran penyelesaian skripsi mahasiswa pada Program Studi Rekayasa Sistem Komputer Tahun Akademik 2024/2025 pada Universitas Bina Insan, perlu ditunjuk pembimbing skripsi yang bertanggung jawab penuh pada pelaksanaan bimbingan tersebut;
b. Bahwa untuk keperluan sebagaimana poin satu tersebut di atas perlu ditetapkan dengan surat keputusan Dekan.

Mengingat :
1. Undang-undang No. 12 tahun 2012 tentang Pendidikan Nasional;
2. Peraturan Pemerintah Republik Indonesia No. 4 tahun 2014 tentang penyelenggaraan Pendidikan Tinggi;
3. Keputusan Menteri Pendidikan Nasional Republik Indonesia No. 232/U/2000 tentang Pedoman Penyusunan Kurikulum Pendidikan Tinggi dan Penilaian Hasil Belajar Mahasiswa;
4. Keputusan Menteri Pendidikan Nasional Republik Indonesia No. 184/U/2001 tentang pedoman Pengawasan-Pengendalian dan Pembinaan Program Diploma, Sarjana dan Pascasarjana di Perguruan Tinggi;
5. SK Menteri Riset, Teknologi Dan Pendidikan Tinggi Republik Indonesia Nomor 223/KPT/I/2019 Tentang Izin Penggabungan Sekolah Tinggi Ilmu Ekonomi Musi Rawas dan Sekolah Tinggi Manajemen Ilmu Komputer Musi Rawas Menjadi Universitas Bina Insan;
6. SK Ketua Yayasan Nomor 01.113/YPDT-Plg/KP/SK/IV/2019 Tentang Pengangkatan Dekan Universitas Bina Insan Lubuklinggau;
7. SK Dekan Universitas Bina Insan Nomor 1235/UNIV.BI/R/KP/SK/2020 Tentang Pengangkatan Pejabat Pada Universitas Bina Insan Lubuklinggau;
8. Statuta Universitas Bina Insan Lubuklinggau;

MEMUTUSKAN

Menetapkan :
Pertama : Mengangkat nama-nama yang tercantum pada lampiran surat keputusan ini sebagai Dosen Pembimbing 1 dan Pembimbing 2 Skripsi Program Studi Rekayasa Sistem Komputer Tahun Akademik 2024/2025 pada Universitas Bina Insan.
Kedua : Semua biaya yang timbul akibat dikeluarkanya surat keputusan ini dibebankan pada anggaran Universitas Bina Insan.
Ketiga : Surat keputusan ini berlaku sejak tanggal ditetapkan, dengan ketentuan apabila ternyata dikemudian hari terdapat kekeliruan dalam penetapan surat keputusan ini akan diperbaiki sebagaimana mestinya;

Ditetapkan di : Lubuklinggau
Pada tanggal : 29 Agustus 2024
Dekan Fakultas Ilmu Teknik

Dr. Rudi Kurniawan, S.T., M.Kom

Tembusan Yth:
1. Ketua Yayasan Pendidikan Dwi Tunggal Palembang (sebagai laporan)
2. Rektor Unversitas Bina Insan (sebagai laporan)
3. Arsip

Lampiran Surat Keputusan Dekan Fakultas Ilmu Teknik
Universitas Bina Insan
Nomor : 089 /UNIV.BI/F.IT.2/SK/2024
Tanggal : 29 Agustus 2024
Tentang : Susunan Pengangkatan Dosen Pembimbing Skripsi Program Studi Rekayasa Sistem Komputer Tahun Akademik 2024/2025

No	NIM	Nama Mahasiswa	Pembimbing 1	Pembimbing 2
1	2102010001	Tiara Saputri	Dr. M. Agus Syamsul Arifin, S.St., M.Kom	Novi Lestari, M.Kom
2	2102010003	Nadiya Rezika	Elmayati, M. Kom	Novi Lestari, M.Kom
3	2102010004	Masriani	Dr. M. Agus Syamsul Arifin, S.St.,M.Kom	Muhammad Nur Alamsyah, M.Kom
4	2102010005	Alfia Tiara Permatasari	Dr. M. Agus Syamsul Arifin, S.St., M.Kom	Rusdiyanto, M.Kom
5	2102010008	Muhamad Akbar Okta Wijaya	Dr. Muhamad Akbar, S.T.,M.IT	Armanto,M.Kom
6	2102010009	I Made Dendy Nirayana	Nelly Khairani Daulay, M.Kom	Cindi Wulandari, M.Kom
7	2102010010	Aji Aris Nasution	Davit Irawan, M.Kom	Fido Rizki, M.Kom
8	2102010011	Ferdayatus Soleha	Asep Toyib Hidayat, M.Kom	Antoni Zulius, M. Kom
9	2102010012	Lilis Anggraini	Dr. Rudi Kurniawan, S.T., M.Kom	Harma Oktavia Lingga Wijaya, M. Kom
10	2102010013	Ferro Audi Pajrin	Dr. Muhamad Akbar, S.T.,M.IT	Deni Nurdiansyah,M.Kom
11	2102010016	Satrya S	Novi Lestari, M.Kom	Joni Karman,M.Kom
12	2102010018	Saptama Hardika	Dr. M. Agus Syamsul Arifin, S.St., M.Kom	Deni Nurdiansyah,M.Kom
13	2102010019	Aisah Rahmawati	Dr. Muhamad Akbar, S.T.,M.IT	Deni Nurdiansyah,M.Kom
14	2102010020	Lia Putri Fadillah	Bunga Intan, M.Kom	Nelly Khairani Daulay, M. Kom
15	2102010021	Deska Dhea Zalbillah	Dr. M. Agus Syamsul Arifin, S.St., M. Kom	Armanto,M.Kom
16	2102010023	Dwi Puspita Sari	Dr. Rudi Kurniawan, S.T., M.Kom	Deni Nurdiansyah,M.Kom
17	2102010024	Kharisma Deni Saputra	Andri Anto Tri Susilo, M.Kom	Lukman Hakim, M. Kom
18	2102010026	M. Sandy Tirta	Dr. Rudi Kurniawan, S.T., M.Kom	Antoni Zulius, M.Kom
19	2102010046	Jeksa Apriansah	Nelly Khairani Daulay, M.Kom	Deni Nurdiansyah,M.Kom
20	2102010048	Melki Oktarigen	Lukman Sunardi, M.Kom	Muhammad Irvai, M.Kom
21	2102010050	Ronaldo	Dr. Susanto, M.Kom	Ahmad Sobri, M.Kom
22	2102010051	M.Ikhsan Rizki Pratama	Armanto,M.Kom	Satrianansyah, M.Kom
23	2102010052	Riski Kurniawan	Budi Santoso, M.Kom	Bunga Intan, M.Kom

YAYASAN PENDIDIKAN DWI TUNGGAL PALEMBANG
UNIVERSITAS BINA INSAN
FAKULTAS ILMU TEKNIK
Jalan Jendral Besar H.M. Soeharto KM.13 Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provinsi Sumatera Selatan

KEPUTUSAN
DEKAN FAKULTAS ILMU TEKNIK
Nomor: [illegible]BI/FIT.2/SK/2025

TENTANG
PENGANGKATAN DOSEN PENGUJI PROPOSAL SKRIPSI MAHASISWA
PROGRAM STUDI REKAYASA SISTEM KOMPUTER FAKULTAS ILMU TEKNIK
UNIVERSITAS BINA INSAN LUBUKLINGGAU

DENGAN RAHMAT TUHAN YANG MAHA ESA, UNIVERSITAS BINA INSAN LUBUKLINGGAU

Memperhatikan : Bahwa dengan selesainya mahasiswa menyusun Proposal Skripsi Program Studi Rekayasa Sistem Komputer Fakultas Ilmu Teknik Tahun Akademik 2024/2025, maka perlu menunjuk dan mengangkat Dosen Penguji Proposal Skripsi untuk menguji mahasiswa tersebut dalam menyelesaikan kuliahnya di lingkungan Universitas Bina Insan Lubuklinggau;

Menimbang :
1. Bahwa dalam upaya menyelenggarakan pendidikan tinggi yang berkualitas dipandang perlu mengangkat Dosen Penguji Proposal Skripsi di lingkungan Universitas Bina Insan Lubuklinggau;
2. Sehubungan dengan Butir 1 (satu) tersebut di atas, maka dipandang perlu mengeluarkan Surat Keputusan sebagai landasan hukumnya;

Mengingat :
1. Undang-Undang No. 20 Tahun 2003 tentang Sistem Pendidikan Nasional;
2. Peraturan Pemerintah Republik Indonesia No. 60 Tahun 1999 tentang Pendidikan Tinggi;
3. Keputusan Menteri Pendidikan Nasional Republik Indonesia No. 232/U/2000 tentang Pedoman Penyusunan Kurikulum Pendidikan Tinggi dan Penilaian Hasil Belajar Mahasiswa;
4. Keputusan Menteri Pendidikan Nasional Republik Indonesia No. 184/U/2001 tentang Pedoman Pengawasan-pengendalian dan Pembinaan Program Diploma, Sarjana dan Pascasarjana di Perguruan Tinggi;
5. SK Menteri Riset, Teknologi Dan Pendidikan Tinggi Republik Indonesia Nomor 223/KPT/I/2019 Tentang Izin Penggabungan Sekolah Tinggi Ilmu Ekonomi Musi Rawas dan Sekolah Tinggi Manajemen dan Ilmu Komputer Musi Rawas Menjadi Universitas Bina Insan;
6. SK Ketua Yayasan Nomor 01.113/YPDT-Plg/KP/SK/IV/2019 Tentang Pengangkatan Rektor Universitas Bina Insan Lubuklinggau;
7. SK Rektor Universitas Bina Insan Nomor 1235/UNIV.BI/R/KP/SK/2020 Tentang Pengangkatan Pejabat Pada Universitas Bina Insan Lubuklinggau;
8. Statuta Universitas Bina Insan Lubuklinggau;

MEMUTUSKAN

Menetapkan

Pertama : Mengangkat Saudara yang namanya tercantum pada lampiran ini, sebagai penguji Proposal Skripsi Program Studi Rekayasa Sistem Komputer Fakultas Ilmu Teknik Tahun Akademik 2024/2025 di Universitas Bina Insan Lubuklinggau;

Kedua : Semua biaya yang timbul akibat dikeluarkannya Surat Keputusan ini dibebankan kepada anggaran Universitas Bina Insan Lubuklinggau atau dana khusus yang disediakan untuk itu;

Ketiga : Kepada yang bersangkutan diberikan honorarium yang besarnya sesuai dengan peraturan Universitas Bina Insan Lubuklinggau;

Keempat : Surat Keputusan ini berlaku sejak tanggal ditetapkan, dengan ketentuan apabila ternyata dikemudian hari terdapat kekeliruan dalam penetapan surat keputusan ini, akan diperbaiki sebagaimana mestinya;

Demikan Surat Keputusan ini ditetapkan untuk dilaksanakan sebagaimana mestinya.

Ditetapkan di : Lubuklinggau
Pada tanggal : 09 Januari 2025
Dekan Fakultas Ilmu Teknik

Dr. Rudi Kurniawan, S.T., M.Kom

Tembusan Yth.
1. Ketua Yayasan Pendidikan Dwi Tunggal Palembang (sebagai laporan)
2. Rektor Universitas Bina Insan (sebagai laporan)
3. Arsip

Lampiran Surat Keputusan Dekan Fakultas Ilmu Teknik Universitas Bina Insan Lubuklinggau
Nomor : [illegible]
Tanggal : 09 Januari 2025
Tentang : Susunan Pengangkatan Dosen Penguji Sidang Proposal Skripsi Program Studi Rekayasa Sistem Komputer TA. 2024/2025

No	Nama Mahasiswa	NIM	Ketua	Sekretaris	Anggota	Hari	Tanggal	Jam	Ruang
1	Jeksa Aprianshah	2102010046	Nelly Khairani Daulay, M.Kom	[illegible]	Dr. Muhamad Akbar, S.T., M.IT	Jum'at	10/01/2025	10.00-11.00	Ruang Sidang 3
2	Sisi Apriyani	21020100570	Novi Lestari, M.Kom	[illegible]	Nelly Khairani Daulay, M.Kom	Jum'at	10/01/2025	15.00-16.00	Ruang Sidang 1
3	Zaid Haritsyah	2102010032	Dr. Muhamad Akbar, S.T., M.IT	[illegible]	Dr. Rudi Kurniawan, S.T., M.Kom	Jum'at	10/01/2025	15.00-16.00	Ruang Sidang 2
4	Ferro Audi Pajrin	2102010013	Dr. Muhamad Akbar, S.T., M.IT	Deni Nurdiansyah, M.Kom	Novi Lestari, M.Kom	Jum'at	10/01/2025	16.00-17.00	Ruang Sidang 1
5	Prayogo Lavendo	2102010024	Dr. Rudi Kurniawan, S.T., M.Kom	Nelly Khairani Daulay, M.Kom	Dr. Muhamad Akbar, S.T., M.IT	Jum'at	10/01/2025	16.00-17.00	Ruang Sidang 2

Lubuklinggau, 09 Januari 2025
Dekan Fakultas Ilmu Teknik

Dr. Rudi Kurniawan, S.T., M.Kom

T KEPUTUSAN
LTAS ILMU TEKNIK
UNIV.BI/FIT.2/SK/2025

ENTANG
PENGA N PENGUJI SKRIPSI MAHASISWA
PROGRAM STUDI REKAYASA SISTEM KOMPUTER FAKULTAS ILMU TEKNIK
UNIVERSITAS BINA INSAN LUBUKLINGGAU

DENGAN RAHMAT TUHAN YANG MAHA ESA, UNIVERSITAS BINA INSAN LUBUKLINGGAU

Memperhatikan : Bahwa dengan selesainya mahasiswa menyusun Skripsi Program Studi Rekayasa Sitem Komputer Fakultas Ilmu Teknik Tahun Akademik 2024/2025, maka perlu menunjuk dan mengangkat Dosen Penguji Skripsi untuk menguji mahasiswa tersebut dalam menyelesaikan kuliahnya di lingkungan Universitas Bina Insan Lubuklinggau;

Menimbang :
1. Bahwa dalam upaya menyelenggarakan pendidikan tinggi yang berkualitas dipandang perlu mengangkat Dosen Penguji Proposal Skripsi di lingkungan Universitas Bina Insan Lubuklinggau;
2. Sehubungan dengan Butir 1 (satu) tersebut di atas, maka dipandang perlu mengeluarkan Surat Keputusan sebagai landasan hukumnya;

Mengingat :
1. Undang-Undang No. 20 Tahun 2003 tentang Sistem Pendidikan Nasional;
2. Peraturan Pemerintah Republik Indonesia No. 60 Tahun 1999 tentang Pendidikan Tinggi;
3. Keputusan Menteri Pendidikan Nasional Republik Indonesia No. 232/U/2000 tentang Pedoman Penyusunan Kurikulum Pendidikan Tinggi dan Penilaian Hasil Belajar Mahasiswa;
4. Keputusan Menteri Pendidikan Nasional Republik Indonesia No. 184/U/2001 tentang Pedoman Pengawasan-pengendalian dan Pembinaan Program Diploma, Sarjana dan Pascasarjana di Perguruan Tinggi;
5. SK Menteri Riset, Teknologi Dan Pendidikan Tinggi Republik Indonesia Nomor 223/KPT/I/2019 Tentang Izin Penggabungan Sekolah Tinggi Ilmu Ekonomi Musi Rawas dan Sekolah Tinggi Manajemen dan Ilmu Komputer Musi Rawas Menjadi Universitas Bina Insan;
6. SK Ketua Yayasan Nomor 01.113/YPDT-Plg/KP/SK/IV/2019 Tentang Pengangkatan Rektor Universitas Bina Insan Lubuklinggau;
7. SK Rektor Universitas Bina Insan Nomor 1235/UNIV.BI/R/KP/SK/2020 Tentang Pengangkatan Pejabat Pada Universitas Bina Insan Lubuklinggau;
8. Statuta Universitas Bina Insan Lubuklinggau;

M E M U T U S K A N

Menetapkan

Pertama : Mengangkat Saudara yang namanya tercantum pada lampiran ini, sebagai penguji Skripsi Program Studi Rekayasa Sistem Komputer Fakultas Ilmu Teknik Tahun Akademik 2024/2025 di Universitas Bina Insan Lubuklinggau;

Kedua : Semua biaya yang timbul akibat dikeluarkannya Surat Keputusan ini dibebankan kepada anggaran Universitas Bina Insan Lubuklinggau atau dana khusus yang disediakan untuk itu;

Ketiga : Kepada yang bersangkutan diberikan honorarium yang besarnya sesuai dengan peraturan Universitas Bina Insan Lubuklinggau;

Keempat : Surat Keputusan ini berlaku sejak tanggal ditetapkan, dengan ketentuan apabila ternyata dikemudian hari terdapat kekeliruan dalam penetapan surat keputusan ini, akan diperbaiki sebagaimana mestinya;

Demikan Surat Keputusan ini ditetapkan untuk dilaksanakan sebagaimana mestinya.

Ditetapkan di : Lubuklinggau
Pada tanggal : 23 Januari 2025
Dekan Fakultas Ilmu Teknik,

Dr. Rudi Kurniawan, S.T., M.Kom

Tembusan Yth.
1. Ketua Yayasan Pendidikan Dwi Tunggal Palembang (sebagai laporan)
2. Arsip

Lampiran Surat Keputusan Dekan Fakultas Komputer Universitas Bina Insan Lubuklinggau
Nomor : [illegible]/UNIV BI/FIT-2/SK/2025
[illegible]
Program Studi Rekayasa Sistem Komputer TA. 2024/2025

No	Nama Mahasiswa	NIM	Ketua	Sekretaris	Anggota	Hari	Tanggal	Jam	Ruang
1	Fikri Haikal	2102010002	Armanto, M.Kom	[illegible], M.Kom	Budi Santoso, M.Kom	Jum'at	24/01/2025	13.00-14.00	Ruang Sidang 3
2	Ferro Audi Pajrin	2102010013	Dr.Muhamad Akbar, S.T., M[illegible]	[illegible] M.Kom	Novi Lestari, M.Kom	Jum'at	24/01/2025	13.00-14.00	Ruang Sidang 4
3	Lia Putri Fadillah	2102010020	Nelly Khairani Daulay, M. K[illegible]	[illegible]	Elmayati, M.Kom	Jum'at	24/01/2025	13.00-14.00	Ruang Sidang 5
4	Maryandi Andika Putra	2102010030	Dr.M.Agus Syamsul A.S.St.,[illegible]	[illegible]	Novi Lestari, M.Kom	Jum'at	24/01/2025	14.00-15.00	Ruang Sidang 1
5	M. Sandy Tirta	2102010026	Dr. Rudi Kurniawan, ST., M.Kom	Antoni Zulius, M.Kom	Armanto, M.Kom	Jum'at	24/01/2025	14.00-15.00	Ruang Sidang 2
6	Muhammad Ikhsan Rizki Pratama	2102010051	Armanto, M.kom	Satrianansyah, M.kom	Dr M.Agus Syamsul A.S.St., M.Kom	Jum'at	24/01/2025	14.00-15.00	Ruang Sidang 3
7	Ilham Ramadoni	1902010012	Novi Lestari, M.Kom	Davit irawan M.kom	Budi Santoso, M.Kom	Jum'at	24/01/2025	14.00-15.00	Ruang Sidang 4
8	Ferdayatus Solcha	2102010011	Asep Toyib hidayat, M.Kom	Antoni Zulius, M.Kom	Dr.Muhamad Akbar, S.T., M.IT	Jum'at	24/01/2025	15.00-16.00	Ruang Sidang 1
9	Deska Dhea Zalbillah	2102010021	Dr.M.Agus Syamsul A,S.St., M.Kom	Armanto, M.Kom	Dr. Susanto, M.Kom	Jum'at	24/01/2025	15.00-16.00	Ruang Sidang 2
10	Aisah Rahmawati	2102010019	Dr. Muhamad Akbar. S.T., M.I.T	Deni Nurdiansyah, M.Kom	Budi Santoso, M.Kom	Jum'at	24/01/2025	15.00-16.00	Ruang Sidang 3
11	I Made Dendy Nirayana	2102010009	Nelly Khairani Daulay, M. Kom	Cindi Wulandari, M. Kom	Bunga Intan,M.Kom	Jum'at	24/01/2025	15.00-16.00	Ruang Sidang 4
12	Aji Aris Nasution	2102010010	Davit irawan M.kom	Fido rizki M.kom	Armanto, M.kom	Jum'at	24/01/2025	15.00-16.00	Ruang Sidang 5
13	Rahmat Alfarabi	1902010052	Budi Santoso, M.Kom	Antoni Zulius, M.Kom	Armanto, M.Kom	Jum'at	24/01/2025	16.00-17.00	Ruang Sidang 1
14	Ade Wahyuda Pratama	1902010014	Dr. M. Agus Syamsul A,ST.,M.Kom	M. Nur Alamsyah,M.Kom	Dr. Rudi Kurniawan, S.T, M.Kom	Jum'at	24/01/2025	16.00-17.00	Ruang Sidang 2

YAYASAN PENDIDIKAN UCY TUNGGAL PALEMBANG
UNIVERSITAS BINA INSAN
Jalan Jendral Besar H.M. Soeharto KM.13 Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provisi Sumatera Selatan

Form ...iuan Judul Skripsi
Program ...yasa Sistem Komputer

Nama	: Ferro
NIM	: 2102010013
Alamat	: Desa Muara Kati Baru I
No.Hp	: 083169916185

Rumusan Masalah 1	:	Bagaimana mengembangkan dan mengevaluasi metode stacking berbasis ensemble untuk deteksi malware yang mampu mengoptimalkan algoritma, melakukan tuning hyperparameter secara efektif, serta mengimplementasikan pendekatan augmentasi data guna menangani dataset tidak seimbang dan beradaptasi terhadap variasi malware baru?
Judul 1	:	Pengembangan dan Evaluasi Metode Stacking Berbasis Ensemble untuk Deteksi Malware: Optimalisasi Algoritma, Tuning Hyperparameter, dan Pendekatan Augmentasi Data dengan Fokus pada Dataset Tidak Seimbang dan Adaptasi terhadap Variasi Malware Baru
Rumusan Masalah 2	:	Bagaimana mengembangkan sistem deteksi phishing yang mampu meningkatkan akurasi melalui integrasi machine learning, analisis konten, dan user experience untuk memberikan perlindungan real-time yang adaptif terhadap ancaman phishing yang berkembang?
Judul 2	:	Inovasi dalam Deteksi Phishing: Integrasi Machine Learning, Analisis Konten, dan User Experience untuk Sistem Real-Time
Rumusan Masalah 3	:	Bagaimana penerapan algoritma pembelajaran mesin dapat meningkatkan akurasi dalam mengklasifikasikan serangan siber berdasarkan analisis log dari Web Application Firewall, sehingga dapat mendeteksi berbagai jenis serangan dengan lebih efisien dan real-time?
Judul 3	:	Penerapan Algoritma Pembelajaran Mesin untuk Klasifikasi Serangan Siber Menggunakan Analisis Log Web Application Firewall

Diusulkan Judul Nomor 1(satu)/ 2(Dua)/ 3(Tiga)*

0733-4553932 (Rektorat Universitas Bina Insan)
0733-3280300
0733-3280200 (Pascasarjana)
0812-1826-6228 (Marketing UNIVBI)
0852-3151-5800 (Admin UNIVBI)
Admin@univbinainsan.ac.id
univbinainsan.ac.id - pasca.univbinainsan.ac

Lubuklinggau, 21 Oktober 2024
Mahasiswa yang mengusulkan,

Ferro Audi Pajrin

Menyetujui Dosen Pembimbing,

Pembimbing 1 Dr. Muhamad Akbar, S.T., M.IT (.....)

Pembimbing 2 Deni Nurdiansyah, M.Kom. (.....)

Mengesahkan
Dekan Fakultas Ilmu Teknik

Dr.Rudi Kurniawan, ST., M.Kom.

Mengetahui
Ketua Program Studi,

Armanto, S.Kom.

YAYASAN PENDIDIKAN DWI TUNGGAL PALEMBANG
UNIVERSITAS BINA INSAN
FAKULTAS ILMU TEKNIK
Jalan Jendral Besar H.M. Soeharto KM.13 Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provinsi Sumatera Selatan

LEMBAR UJIAN SKRIPSI

Nama Mahasiswa : Ferro Audi P
NIM : 210201001
Jenjang Pendidikan : Strata 1 (S1)
Fakultas : Ilmu Teknik
Program Studi : Rekayasa Sistem Komputer
Konsentrasi : -
Judul : Pengembangan dan Evaluasi metode Stacking berbasis Ensemble Untuk deteksi Malware dengan Pendekatan Augmentasi data pada Dataset tidak seimbang

No	Dosen Penguji	Komentar Perbaikan	Tanda Tangan Ujian	Tanda Tangan Revisi
1	M. Akbar			
2	Deni Nurdiangyah, M.kom			
3	Novi Lestari, M.kom	Sudah baik		

Lubuklinggau, ...2025
Ketua Program Studi Rekayasa Sistem Komputer

Armanto, M.Kom

YAYASAN PENDIDIKAN DWI TUNGGAL PALEMBANG
UNIVERSITAS BINA INSAN
FAKULTAS ILMU TEKNIK
Jalan Jendral Besar H.M. Soeharto KM.13 Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provisi Sumatera Selatan

LEMBAR PER[illegible]MINAR PROPOSAL SKRIPSI

Nama Mahasiswa : Ferro A
NIM : 210201[illegible]
Jenjang Pendidikan : Strata 1 (S1)
Fakultas : Ilmu Teknik
Program Studi : Rekayasa Sistem Komputer
Konsentrasi : .....
Judul : Pengembangan dan Evaluasi Metode Stacking Berbasis Ensemble untuk Deteksi Malware : Optimasi Algoritma, Tuning Hyperparameter, dan Pendekatan Augmentasi Data dengan Fokus pada Dataset Tidak Seimbang dan adaptasi terhadap variasi Malware Baru

No	Dosen Penguji	Komentar Perbaikan	Tanda Tangan Ujian	Tanda Tangan Revisi
1	Dr. Muhamad Akbar, S.T., M.IT			
2	Deni Nurdiansyah, M. kom			
3	Novi Lestari, M.kom			

Lubuklinggau, ...2025
Ketua Program Studi Rekayasa Sistem Komputer

(.....)

0733-4553932 (Rektorat Universitas Bina Insan)
0733-3280300
0733-3280200 (Pascasarjana)
0812-1826-6228 (Marketing UNIVBI)
0852-3151-5800 (Admin UNIVBI)
Admin@univbinainsan.ac.id
univbinainsan.ac.id - pasca.univbinainsan.ac.id

YAYASAN PENDIDIKAN DUTA LUNGGAU PALEMBANG
UNIVERSITAS BINA INSAN
FAKULTAS ILMU KOMPUTER
Jalan Jendral Besar H.M. Soeharto KM.13 Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provinsi Sumatera Selatan

LEMBAR BIMBINGAN SKRIPSI

Nama : Ferro Audi Pajri
Nim : 2102010013
Program Studi : Rekayasa Sistem
Pembimbing 1 : Dr. Muhamad Al... T.
Pembimbing 2 : Deni Nurdiansyah, M.Kom.
Judul : Pengembangan dan Evaluasi metode Stacking berbasis Ensemble untuk deteksi Malware Dengan pendekatan Augmentasi data pada Dataset tidak Seimbang

NO	TANGGAL	TOPIK	KOMENTAR PEMBIMBING	TANDA TANGAN PEMBIMBING	
				1	2
			Revisi Hasil, kesh analisis.		
			CM dicek lagi. Semoga. arahan		
	19/ 2025		Acc ujian		

Lubuklinggau, ...2025
Ketua Program Studi Rekayasa Sistem Komputer

(Armanto, S.Kom.)

YAYASAN PENDIDIKAN DWI TUNGGAL PALEMBANG
UNIVERSITAS BINA INSAN
FAKULTAS ILMU TEKNIK
Jalan Jendral Besar H.M. Soeharto KM.13 Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provisi Sumatera Selatan

LEM BINGAN SKRIPSI

Nama : Ferro Audi Pajrin
Nim : 2102010013
Program Studi : Rekayasa Sistem
Pembimbing 1 : Dr. Muhamad Akbar, S.T., M.IT.
Pembimbing 2 : Deni Nurdiansyah, M.Kom.
Judul : Pengembangan dan Evaluasi metode Stacking berbasis Ensemble untuk deteksi Malware Dengan pendekatan Augmentasi data pada Dataset tidak Seimbang

NO	TANGGAL	TOPIK	KOMENTAR PEMBIMBING	TANDA TANGAN PEMBIMBING	
				1	2
1.	3/1/2025	Skripsi	-perbaiki BAB IV hasil dan Pembahasan		
2.	6/1/2025	Skripsi	- perbaiki BAB V kesimpulan dan saran		
3.	8/1/2025	Skripsi	- Lengkapi Berkas Acc lanjut PI		

Lubuklinggau, ...2025
Ketua Program Studi Rekayasa Sistem Komputer

(Armanto, S.Kom.)

LEMBAR B[illegible]N PROPOSAL SKRIPSI

Nama : Ferro Audi Pajrin
Nim : 2102010013
Program Studi : Rekayasa Sistem
Pembimbing 1 : Dr. Muhamad Akbar, S.T., M.T
Pembimbing 2 : Deni Nurdiansyah.,M.Kom
Judul : Pengembangan Dan Evaluasi Metode StackingBerbasis Ensemble Untuk Deteksi Malware: Optimasi Algortima, Tuning Hyperparameter, Dan Pendekatan Augmentasi Data Dengan Fokus Pada Dataset Tidak Seimbang dan Adaptasi Terhadap Variasi Malware Baru

NO	TANGGAL	TOPIK	KOMENTAR PEMBIMBING	TANDA TANGAN PEMBIMBING	
				1	2
1.	27/12/2024	Proposal	Identifikasi masalah Rumusan masalah Rancangan penelitian		
2.	28/12/2024	Proposal	Penulisan perbaiki Metode pengumpulan data		
3.	30/12/2024	Proposal	Rancangan Sistem		
4.	31/12/2024	proposal	Acc lanjut P1		

Lubuklinggau, ...2024
Ketua Program Studi Rekayasa Sistem Komputer

Armanto.,M.Kom

YAYASAN PENDIDIKAN DWI TUNGGAL PALEMBANG
UNIVERSITAS BINA INSAN
FAKULTAS ILMU TERKAIT
Jalan Jendral Besar H.M. Soeharto KM.13 Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provisi Sumatera Selatan

LEMBAR B[illegible]AN PROPOSAL SKRIPSI

Nama : Ferro Audi Pajri[illegible]
Nim : 2102010013
Program Studi : Rekayasa Sistem[illegible]
Pembimbing 1 : Dr. Muhamad A[illegible]
Pembimbing 2 : Deni Nurdiansyah.,M.Kom
Judul : Pengembangan Dan Evaluasi Metode ~~Staeking~~Berbasis Ensemble Untuk Deteksi Malware: Optimasi Algortima, Tuning Hyperparameter, Dan Pendekatan Augmentasi Data Dengan Fokus Pada Dataset Tidak Seimbang dan Adaptasi Terhadap Variasi Malware Baru

NO	TANGGAL	TOPIK	KOMENTAR PEMBIMBING	TANDA TANGAN PEMBIMBING	
				1	2
			Stacking / ensemble Smote Acc ujian.		

Lubuklinggau, ...2024
Ketua Program Studi Rekayasa Sistem Komputer

Armanto.,M.Kom

LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT
UNIVERSITAS BINA INSAN
Jl. Jenderal Besar HM. Soeharto KM.13 Kelurahan Lubuk Kupang Kecamatan Lubuklinggau Selatan I
Kota Lubuklinggau Provinsi Sumatera Selatan

SURAT K[illegible]N BEBAS PLAGIASI

Menerangkan bahwa mahasiswa :

Nama	: Ferro Audi Pajrin
NIM	: 2102010013
Fakultas	: Fakultas Ilmu Teknik
Program Studi	: Rekayasa Sistem Komputer

Memiliki jurnal dengan Judul **"Pengembangan dan Evaluasi Mtetode Stacking Berbasis Ensemble Untuk Deteksi Malware Dengan Pendekatan Augemntasi Data Pada Dataset Tidak Seimbang"** Telah diterbitkan pada Prosiding **:** ESCAF (*Economic, Social Science, Computer, Agriculture and Fisheries*) 4th tahun 2025, sehingga dinyatakan memenuhi standar bebas plagiasi dari Universitas Bina Insan.

Demikian surat keterangan ini disampaikan dengan sebenarnya untuk dapat dipergunakan sebagaimana mestinya.

Lubuklinggau, 13 Februari 2025

Kepala LPPM,

Elmayati, M.Kom

0733-452218 (Rektorat Universitas Bina Insan)
0733-3280300
0733-3280200 (Pascasarjana)
0821-8420-2909 (Rektorat Universitas Bina Insan)
0822-8084-5222
0823-7474-0255 (Pacasarjana)
Lppm@univbinainsan.ac.id
Lppm.univbinainsan.ac.id

FORMULIR KELAYAKAN PENJILIDAN SKRIPSI

Nama : Ferro Audi Pajrin
NIM : 2102010013
Konsentrasi :
Program Studi : Rekayasa Sistem Komputer
Judul : Perbandingan dan Evaluasi Metode Stacking Berbasis Ensemble untuk Deteksi Malware dengan Pendekatan Augmentasi Data Pada Dataset Tidak Seimbang
Dosen Pembimbing I : Dr. Muhamad Akbar, S.T., M.IT.
Dosen Pembimbing II : Deni Nurdiansyah, M.Kom.
Tanggal Ujian Skripsi : 24 Januari 2025

Point Check **:**

1. SAMPUL SKRIPSI	
2. HALAMAN JUDUL SKRIPSI	
3. HALAMAN PENGESAHAN PEMBIMBING SKRIPSI	
4. HALAMAN PENGESAHAN KOMISI PENGUJI SKRIPSI	
5. SURAT PERNYATAAN	
6. ABSTRAK (BAHASA INDONESIA)	
7. *ABSTRACT* (BAHASA INGGRIS)	
8. MOTTO DAN HALAMAN PERSEMBAHAN	
9. KATA PENGANTAR	
10. DAFTAR ISI	
11. DAFTAR TABEL	
12. DAFTAR GAMBAR	
13. DAFTAR LAMPIRAN	
14. DAFTAR RIWAYAT HIDUP	
15. ISI SKRIPSI (BAB I S/D BAB V)	
16. DAFTAR PUSTAKA	
17. LAMPIRAN – LAMPIRAN	
➢ SK. PEMBIMBING DAN PENGUJI (PROPOSAL, HASIL, SKRIPSI)	
➢ SURAT KETERANGAN TELAH MELAKSANAKAN RISET	
➢ FORMULIR PERBAIKAN UJIAN SKRIPSI	
➢ FORMULIR PERBAIKAN SEMINAR PROPOSAL	
➢ FORMULIR BIMBINGAN SKRIPSI	
➢ FORMULIR BIMBINGAN PROPOSAL	
➢ *PLAGIARISM SCAN REPORT* (*TURNITIN)*	
➢ JURNAL (TEMPLATE ADA DI LPPM)	
➢ *LISTING* PROGRAM/HASIL WAWANCARA/KUISIONER DIGUNAKAN	
➢ LAIN-LAIN YANG DIPERLUKAN	

Dengan ini dinyatakan layak untuk **layak** untuk di jilid sesuai dengan format yang berlaku dilingkungan Program Studi …….. Universitas Bina Insan Lubuklinggau.

Pemeriksa Kelayakan,
Ketua Prodi Rekayasa Sistem Komputer

……

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

Pengembangan dan Evaluasi Metode Stacking Berbasis Ensemble untuk Deteksi Malware dengan Pendekatan Augmentasi Data Pada Dataset Tidak Seimbang

Ferro Audi Pajrin[1], M[illegible]kbar[2], Deni Nurdiansyah[3]
[1]Program Studi Rekayasa Sistem [illegible] Universitas Bina Insan, Lubuklinggau

e-mail: *[1]ferroaudipajrin@gmail.com, [2]muhamad.akbar@univbinainsan.ac.id,
[3]deninurdiansyah@univbinainsan.ac.id

Abstrak

Serangan malware pada perangkat Android semakin meningkat, sementara metode deteksi berbasis tanda tangan memiliki keterbatasan dalam mengenali varian baru. Penelitian ini mengembangkan metode stacking berbasis ensemble dengan augmentasi data menggunakan SMOTE untuk meningkatkan akurasi deteksi malware pada dataset tidak seimbang. Dataset TUANDROMD digunakan dalam penelitian ini, terdiri dari 4.465 sampel malware dan goodware. Base classifiers yang digunakan mencakup Random Forest, Support Vector Machine (SVM), dan Extra Trees, sementara Logistic Regression digunakan sebagai meta-classifier. Model dioptimalkan menggunakan GridSearchCV dengan validasi silang 5-fold untuk mendapatkan kombinasi hyperparameter terbaik. Setelah pelatihan model selesai, evaluasi dilakukan menggunakan data uji terpisah dengan metrik akurasi, presisi, recall, F1-score, dan AUC. Hasil pengujian menunjukkan bahwa metode stacking ensemble yang dikembangkan mencapai akurasi 99,58%, presisi 100%, recall 99,16%, F1-score 99,58%, dan AUC 99,95%. Teknik SMOTE terbukti efektif dalam menyeimbangkan distribusi kelas minoritas dan meningkatkan performa model secara keseluruhan. Hasil penelitian ini menunjukkan bahwa pendekatan stacking ensemble dengan augmentasi data dapat menjadi solusi efektif dalam meningkatkan deteksi malware Android.

Kata Kunci—Deteksi Malware, Metode Stacking, Augmentasi Data, SMOTE, Machine Learning.

Abstract

The increasing malware attacks on Android devices pose a challenge as signature-based detection methods struggle to recognize new variants. This study develops an ***ensemble stacking*** *method with SMOTE data augmentation to enhance malware detection accuracy on imbalanced datasets. The TUANDROMD dataset, consisting of 4,465 malware and goodware samples, is used. Base classifiers include Random Forest, Support Vector Machine (SVM), and Extra Trees, while Logistic Regression is used as the meta-classifier. The model is optimized using GridSearchCV with 5-fold cross-validation to obtain the best hyperparameter combination. After training, evaluation is conducted using a separate test set with accuracy, precision, recall, F1-score, and AUC metrics. The results indicate that the proposed stacking ensemble method achieves 99.58% accuracy, 100% precision, 99.16% recall, 99.58% F1-score, and 99.95% AUC.*

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

The SMOTE technique effectively balances the minority class distribution, improving overall model performance. These findings suggest that ensemble stacking with data augmentation provides an effective solution for enhancing Android malware detection.

Keywords—Malware Detection, Stacking Ensemble, Data Augmentation, SMOTE, Machine Learning

I. PENDAHULUAN

Seiring dengan pertumbuhan penggunaan smartphone yang terus meningkat, pada tahun 2023 diperkirakan pengiriman smartphone secara global mencapai 1,48 miliar perangkat. Pada tahun 2021, jumlah pengguna smartphone diproyeksikan mencapai 6,4 miliar, dan angka ini kemungkinan akan meningkat menjadi 7,5 miliar pada 2026 [1]. Android, sebagai sistem operasi mobile dengan pangsa pasar 73%, menjadi pilihan utama berkat fleksibilitasnya yang mendukung berbagai inovasi teknologi, seperti GPS, Wi-Fi, kamera, dan aplikasi berbasis internet. Namun, popularitas ini juga menjadikannya sasaran utama serangan siber, seperti malware yang dirancang untuk mengeksploitasi data sensitif pengguna dan mengganggu perangkat mereka [2]. Malware, atau malicious software, adalah perangkat lunak berbahaya yang dirancang untuk mencuri data, mengganggu sistem, atau menyusupi perangkat tanpa izin pengguna. Malware Android hadir dalam berbagai bentuk, termasuk ransomware, spyware, dan trojan horse, yang sering memanfaatkan kerentanan pada aplikasi atau sistem operasi Android untuk menjalankan serangannya [3]. Ancaman ini terus berkembang seiring dengan meningkatnya jumlah pengguna Android dan kompleksitas malware yang muncul.

Metode tradisional dalam deteksi malware, seperti berbasis tanda tangan (signature-based), memiliki keterbatasan dalam mendeteksi malware yang belum dikenal ataupun yang telah dimodifikasi. Pendekatan ini cenderung kurang adaptif terhadap evolusi malware yang terus berkembang. Oleh karena itu, diperlukan metode deteksi yang lebih canggih, adaptif, dan akurat. Machine learning, khususnya pendekatan berbasis ensemble seperti stacking, telah terbukti sebagai solusi yang menjanjikan untuk meningkatkan akurasi deteksi malware [4]. Beberapa penelitian terkait deteksi malware Android telah dilakukan. Penelitian oleh Soritua & Utama (2023) [5] mengembangkan metode deteksi malware berbasis ensemble stacking dengan seleksi fitur berbasis regularisasi. Penelitian ini menunjukkan akurasi tinggi dalam deteksi malware, tetapi tidak mengimplementasikan teknik augmentasi data, yang merupakan tantangan utama dalam deteksi malware pada dataset tidak seimbang. Penelitian selanjutnya dilakukan oleh Rafrastara et al. (2023) [4], yang menggunakan meta-classifier Logistic Regression dalam pendekatan stacking ensemble. Walaupun hasilnya menunjukkan akurasi yang cukup tinggi, penelitian ini tidak membahas teknik augmentasi data, yang penting untuk penanganan ketidakseimbangan data dalam dataset. Penelitian lain yang dilakukan oleh Rahman et al. (2023) [6] berfokus pada deteksi ransomware Android menggunakan algoritma supervised learning dan augmentasi data (SMOTE), tetapi tidak menggunakan metode stacking ensemble.

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

Berdasarkan kajian tersebut, penelitian ini bertujuan untuk mengatasi tantangan dataset yang tidak seimbang melalui penerapan teknik augmentasi data. Sistem deteksi malware yang dikembangkan akan lebih akurat, adaptif, dan tangguh dengan menggunakan metode stacking berbasis ensemble [4]. Pendekatan ini dipilih karena dapat meningkatkan performa algoritma data mining dibandingkan dengan penggunaan single algorithm secara konvensional [4]. Selain itu, penelitian ini akan mengintegrasikan optimasi algoritma dengan Cross-validation untuk memperkirakan tingkat kesalahan uji dengan cara memisahkan sebagian data pelatihan untuk divalidasi [7]. Proses tuning hyperparameter akan dilakukan menggunakan GridSearchCV, yang terbukti dapat meningkatkan performa model secara signifikan dalam mendeteksi malware [8]. Untuk mengatasi ketidakseimbangan dataset, diterapkan metode SMOTE (Synthetic Minority Over-sampling Technique), yang digunakan untuk menangani ketidakseimbangan kelas dalam data serta membantu model agar lebih peka terhadap kasus dalam kelas minoritas yang mungkin memiliki nilai prediktif penting [9].

Berdasarkan latar belakang dan kajian tersebut, penelitian ini bertujuan untuk Penelitian ini bertujuan mengatasi ketidakseimbangan dataset malware Android melalui metode stacking berbasis ensemble [4]. Pendekatan ini dipilih karena mampu meningkatkan akurasi dibandingkan penggunaan single algorithm [4]. SMOTE (Synthetic Minority Over-sampling Technique) digunakan untuk memperbaiki representasi kelas minoritas, sedangkan GridSearchCV diterapkan dalam tuning hyperparameter guna mengoptimalkan performa model [8]. Dataset yang digunakan adalah TUANDROMD, berisi 4.465 sampel malware dan goodware berbasis izin dan API Android dari Tezpur University, untuk memastikan kinerja model. Penelitian ini berjudul **"Pengembangan dan Evaluasi Metode Stacking Berbasis Ensemble untuk Deteksi Malware dengan Pendekatan Augmentasi Data pada Dataset Tidak Seimbang"**.

II. METODOLOGI PENELITIAN

2.1 Metode Penelitian

Metodologi penelitian ini menggunakan pendekatan *Cross-Industry Standard Process for Data Mining* (CRISP-DM), yang terdiri dari lima tahap: Business Understanding, Data Understanding, Data Preparation, Modeling, dan Evaluation. Pendekatan ini dipilih sebagai kerangka kerja terstruktur dalam pengembangan metode deteksi malware berbasis stacking ensemble[10].

Gambar 2.1. Model CRISP-DM[10].

a. Business Understanding
   Penelitian bertujuan mengembangkan metode **stacking ensemble** untuk deteksi malware Android dengan **tuning hyperparameter GridSearchCV** dan **SMOTE** guna meningkatkan akurasi serta menghindari overfitting.
b. Data Understanding
   Dataset TUANDROMD dianalisis untuk memahami distribusi kelas, mendeteksi nilai hilang, dan menemukan anomali. Analisis korelasi antar fitur dilakukan dengan heatmap guna memilih fitur yang relevan.
c. Data Preparation.

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

Nilai hilang diatasi dengan imputasi, feature scaling dilakukan dengan **MinMaxScaler**, dan **SMOTE** diterapkan untuk memperbaiki representasi kelas minoritas.

d. Modeling.
Model dilatih menggunakan base classifiers (Random Forest, SVM, KNN) dengan **cross-validation**. **Stacking ensemble** menggunakan Logistic Regression sebagai meta-classifier untuk menghasilkan prediksi akhir.

e. Evaluation
Performa model diukur dengan **Confusion Matrix**, akurasi, presisi, recall, F1-score, serta AUC untuk menilai efektivitas deteksi malware.

3.2 Metode Pengumpulan Data

Penelitian ini menggunakan dataset TUANDROMD dari Tezpur University, tersedia di UCI Machine Learning Repository, berisi 4.465 aplikasi Android (3.565 malware dan 899 goodware). Dataset mencakup 178 fitur izin dan 186 fitur API, dengan total 242 atribut biner. Meskipun informatif, dataset ini tidak seimbang, karena malware lebih dominan. Dataset digunakan untuk melatih model deteksi malware guna mengidentifikasi ancaman keamanan Android secara akurat:

3.3 Metode Analisa

Metode Analisis dalam penelitian ini difokuskan pada bagaimana teknik stacking ensemble dapat digunakan untuk mendeteksi malware pada dataset yang tidak seimbang. Kami akan mengevaluasi dampak dari penggunaan teknik SMOTE terhadap performa model dalam mendeteksi malware melalui beberapa langkah berikut:

a. Evaluasi Model

Untuk mengukur seberapa Menggunakan beberapa metrik evaluasi seperti *akurasi*, *presisi*, *recall*, *F1-score*, dan *Area Under Curve (AUC)* untuk mengukur seberapa baik model dalam mendeteksi malware pada dataset yang tidak seimbang

b. Analisis Performa Base Classifiers

Mengevaluasi kontribusi setiap algoritma dasar yang digunakan dalam stacking untuk memahami peran mereka dalam meningkatkan performa model.

c. Evaluasi Hasil Stacking Ensemble

Menganalisis performa meta-classifier (Logistic Regression) pada level 1 untuk mengukur kemampuan stacking ensemble dalam mengintegrasikan kekuatan base classifiers secara efektif.

3.4 Metode Pengujian dan Pengolahan Data

a. Metode Pengujian

Dalam tahap pengujian, metode stacking berbasis ensemble diterapkan dalam kerangka CRISP-DM untuk menjelaskan langkah-langkah terstruktur. Proses ini mencakup pembagian dataset, pengolahan data, pelatihan model, tuning hyperparameter, dan evaluasi performa model menggunakan metrik relevan. CRISP-DM menjadi panduan untuk memastikan setiap tahapan mendukung pengembangan metode deteksi malware yang efektif pada dataset tidak seimbang.

1) Pembagian Dataset untuk Pelatihan dan Pengujian:
   a) Dataset Tuandromd akan displit menjadi 80 % training dan 20% testing Dataset terdiri dari dua kelas utama: *malware* dan *goodmalware.*
   b) ***Cross-validation:*** Dataset dibagi menjadi lima lipatan (folds), di mana empat fold digunakan untuk pelatihan dan satu fold digunakan untuk pengujian. Proses ini diulang lima kali,

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

sehingga setiap fold berperan sebagai data uji sekali.

2) Pengujian Model dengan Stack Ensemble:
   a) Level 0 (Base Classifie Model dilatih menggunal beberapa algoritma dasar, seperti *Random Forest*, *Catboost*, *AdaBoost*, *SVM*, *Logistic Regression*, *KNN*.
   b) Stacking pada level 0: Prediksi dari setiap base classifier pada Level 0 digabungkan menjadi fitur baru sebagai input untuk meta-classifier
   c) Level 1 (Meta-classifier): Logistic Regression digunakan sebagai meta-classifier untuk mengolah input dari Level 0 dan menghasilkan prediksi akhir yang lebih akurat.
3) *Tuning Hyperparameter* dengan *GridSearchCV*

Setelah model dilatih, parameter dari setiap base classifier dioptimalkan menggunakan ***GridSearchCV*** untuk mendapatkan kombinasi parameter terbaik. Teknik ini berjalan seiring dengan proses cross-validation, memastikan setiap iterasi menghasilkan model yang lebih baik.

4) Evaluasi Model

Evaluasi dilakukan untuk menguji efektivitas sistem dalam mendeteksi malware di platform Android menggunakan teknologi *machine learning*. Penilaian ini mencakup pengukuran akurasi, presisi, *recall*, dan *F1-score* untuk memastikan kesesuaian dengan rancangan awal dan mengidentifikasi potensi kesalahan seperti gambar 3.2 dibawah ini [11].

		Sebenarnya	
		Positif	Negatif
Prediksi	Positif	(True Positive)	FN (False Positive)
	Negatif	FP (False Negative)	TN (True Negative)

Gambar 2.2 *Confusion Matrix*[11].

Confusion matrix adalah komponen dasar untuk menemukan akurasi, presisi, dan recall. Akurasi merupakan perbandingan dari prediksi yang benar yang diklasifikasikan oleh sistem. Rumus dasar dalam pengukuran dalam penelitian ini sebagai berikut :

a. Accuracy

Accuracy adalah nilai untuk mengetahui seberapa akurat sistem mengklasifikasikan data tersebut secara benar yang dirumuskan pada formula 1.

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

b. Precision *(Positive Predictive Value)*

Precision (presisi) adalah nilai untuk mengetahui jumlah data positif yang diklasifikasikan secara benar dibagi total data yang diklasifikasikan positif yang dirumuskan pada formula 2.

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

c. Recall (True Positive Rate)

Recall adalah nilai untuk mengetahui berapa persen data kategori positif yang diklasifikasikan dengan benar oleh sistem yang dirumuskan pada formula 3.

$$\text{recall} = \frac{TP}{TP+FN}$$

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

d. F-1 Skor

F1-Score adalah nilai harmonic mean dari presisi dan recall. Nilai terbaik f1-score adalah 1.0 dan terburuknya ada 0. Jika nilai f1-score memiliki skor b maka mengindikasikan bahwa met klasifikasi yang dibangun memi presisi dan recall yang dirumuskan pada formula 3.

$$F1-score = 2x = \frac{Precision \ x \ Recall}{Precision+Recall}$$

$$= \frac{2TP}{2TP+FP+FN}$$

e. Area di bawah kurva (AUC) adalah ukuran area di bawah kurva ROC. Semakin luas wilayahnya, semakin baik model klasifikasi yang diusulkan. ROC adalah representasi grafis dari hubungan antara sensitivitas dan spesifisitas. Nilai ini menunjukkan seberapa baik model dapat memisahkan kelas[12].

b. Metode Pengolahan Data

Proses pengolahan data mengikuti tahapan dalam metodologi ***CRISP-DM*** untuk menyiapkan dataset agar siap digunakan untuk pelatihan dan pengujian model. Langkah-langkah yang dilakukan adalah sebagai berikut:

1) Dataset:

Dataset ***TUANDROMD*** terdiri dari 4.465 contoh (3.565 *malware* dan 899 *goodmalware*) dengan 241 atribut berbasis izin dan API.

2) Prepocessing

a) Pembersihan Data:

(1) Penanganan Nilai Hilang: Nilai hilang dalam dataset diidentifikasi dan diisi menggunakan teknik imputasi seperti rata-rata atau median.

(2) **Penyaringan Data:** Data yang tidak relevan atau duplikat dihapus dari dataset.

b) ***Feature Engineering:***

(1) Seleksi Fitur: Fitur yang relevan untuk deteksi malware dipilih, dan fitur yang tidak relevan dihapus.

(2) ***Feature Scaling:*** ***MinMaxScaler*** digunakan untuk menormalkan fitur agar berada dalam rentang yang sama, memastikan fitur memiliki bobot yang setara dalam model.

3) **Augmentasi Data:**

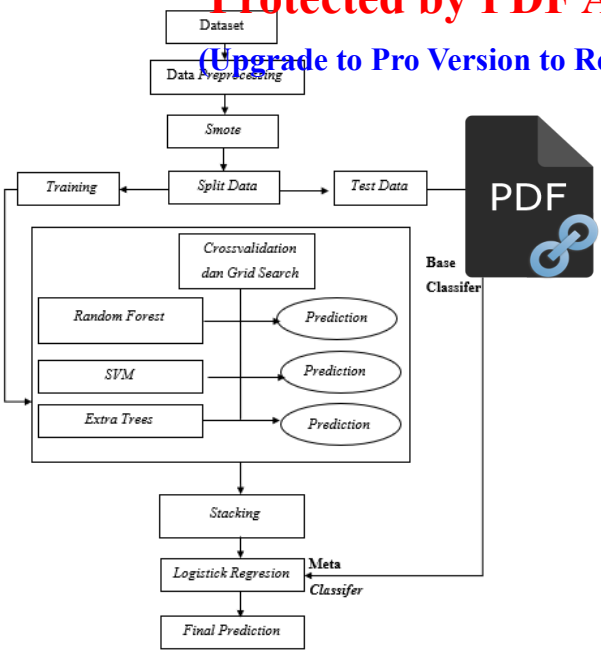
SMOTE (Synthetic Minority Over-sampling Technique): SMOTE diterapkan untuk mengatasi ketidakseimbangan kelas dalam dataset dengan mensintesis sampel baru untuk kelas malware, sehingga distribusi data menjadi lebih seimbang.

4) **Persiapan Dataset:**

Dataset akan displit menjadi data latih dan data uji kemudian crossvalidation akan diterapkan pada data training berdasarkan skema ***5-fold cross-validation***, seperti yang dijelaskan di metode pengujian.

Alur diagram tahapan modeling system pada penelitian ini dapat dilihat di gambar 2.3 dibawah ini.

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx



Gambar 2.3 Tahap modeling sistem[4].

III. HASIL DAN PEMBAHASAN

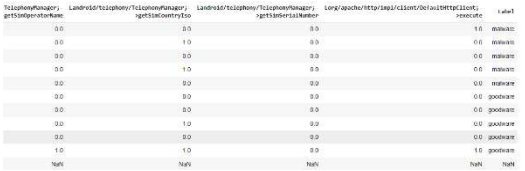
3.1 Hasil Penelitian

Penelitian ini bertujuan mengembangkan metode deteksi malware berbasis *stacking ensemble* dengan pendekatan augmentasi data menggunakan SMOTE pada dataset tidak seimbang. Sistem yang dibangun mengintegrasikan beberapa algoritma pembelajaran mesin sebagai *base classifiers*, seperti Random Forest, Support Vector Machine (SVM), dan Extra Trees. Meta-classifier yang digunakan adalah Logistic Regression. Teknik augmentasi data diterapkan menggunakan SMOTE untuk menyeimbangkan distribusi data antara kelas malware dan goodware. Sistem ini diuji menggunakan validasi silang 5-fold dan dioptimalkan dengan *GridSearchCV* untuk memperoleh kombinasi parameter terbaik. Tahapan sistem dimulai dari pengumpulan data, preprocessing, augmentasi data, pelatihan model, hingga evaluasi performa model menggunakan metrik akurasi, presisi, recall, F1-score, dan AUC.

3.1.1 Dataset Penelitian

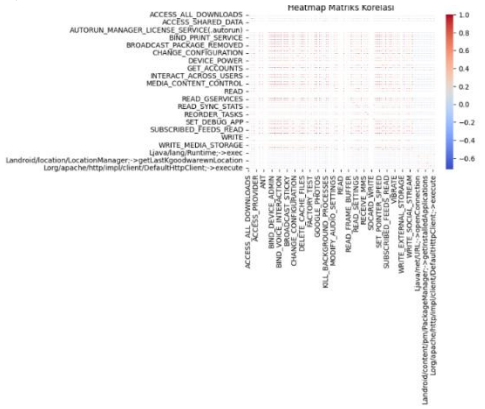
a. dataset

Data yang digunakan telah diproses oleh Tezpur University 1-214: Fitur berbasis izin 215-241: Fitur berbasis API, Dataset TUNADROMD berisi 4465 contoh dan 241 atribut. Atribut target untuk klasifikasi adalah kategori (malware vs goodware). Isi dataset nya dapat dilihat pada gambar 3.1 disana ada 2 kelas yaiut malware dan good malware serta terdapat juga data nan (Not a Number) atau nilai yang hilang atau tidak valid dan akan diatasi dengan fitur numerik dengan median setelah itu akan dilakukan heatmap matrix korelasi dengan memilih hanya kolom yang bertipe data numerik seperti float64 dan int64 sehingga terdapat lah 31 buah seperti pada gambar 3.2 selanjutnya akan dilakukan distribusi label seperti seperti pada gambar 3.3 untuk melihat jumlah sample malware dan goodmalware yang ada.



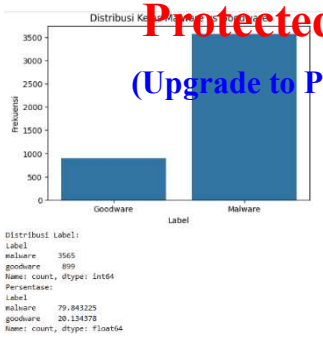
Gambar 3.1 dataset

selanjutnya akan dilakukan distribusi label seperti pada gambar 3.1 untuk melihat jumlah sample malware dan good malware nya.



Gambar 3.2 Heatmap

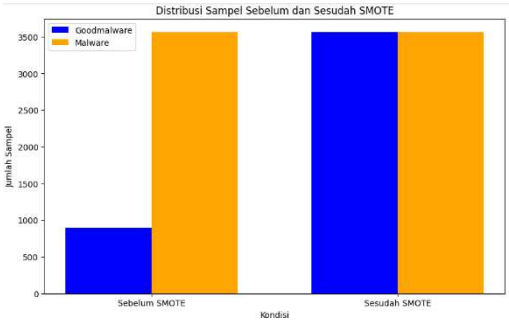
ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx



Gambar 3.3 visualisasi distribusi kelas

b. Processing

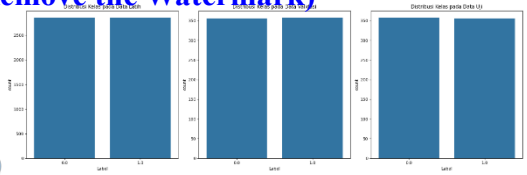
1) **Pembersihan Data:** Menghapus data duplikat dan menangani nilai yang hilang menggunakan imputasi rata-rata.
2) **Feature Scaling:** Menggunakan *MinMaxScaler* untuk memastikan semua fitur berada dalam rentang 0 hingga 1.
3) **Augmentasi Data:** Diterapkan teknik SMOTE untuk menghasilkan sampel sintetis dari kelas minoritas (goodware). Jumlah sample sebelum smote label 0 berjumlah 899 dan label 1.0 berjumlah 3565. Setelah dilakukan smote label 0 berjumlah 3565 dan label 1.0 berjumah 3566 seperti pada gambar 3.4 dibawah ini.



Gambar 3.4 Sebelum dan Sesudah Smote

3.1.2 Split Data

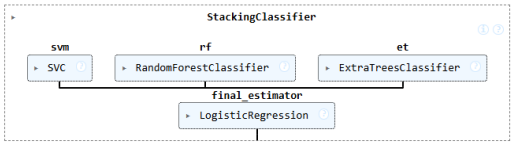
Setelah dilakukan Smote maka akan dilakukan pembagian Dataset atau split data menjadi 3 buah data yaitu data train berjumlah 5.704 sample, data validasi berjumlah 713 data dan untuk data test berjumlah 714 data seperti pada gambar 3.5 dibawah ini.



Gambar 3.5 Split Data

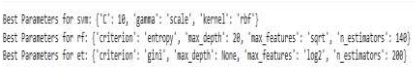
3.1.3 Pengujian Model

Model Base Classifier dan Stacking Base classifiers yang digunakan adalah Random Forest, Support Vector Machine (SVM), dan Extra Trees. Model stacking menggunakan Logistic Regression sebagai meta-classifier untuk menggabungkan prediksi dari base classifiers. Proses ini dirancang untuk mengoptimalkan performa dengan memanfaatkan kekuatan masing-masing algoritma. Visualisasi terkait ditampilkan pada Gambar 3.6



Gambar 3.6 Visualisasi Model

a. Tuning Hyperparameter
Didalam baseclasifer masing masing algoritma akan dilakukan tuning hyperparameter dan dievaluasi dengan crossvalidation untuk menentukan base parameter terbaik seperti pada gambar 3.7 dibawah ini



Gambar. 3.7 Best Parameter

Hasil tuning hyperparameter menggunakan GridSearchCV dengan evaluasi menggunakan crossvalidation 5-folds menghasilkan parameter terbaik untuk masing-masing model sebagai berikut:

1) Support Vector Machine (SVM):
Parameter terbaik yang ditemukan adalah C=10, gamma='scale', dan

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

kernel='rbf'. Nilai C=10 menunjukkan regularisasi yang seimbang antara overfitting dan underfitting, sementara gamma='scale' memungkinkan ke[rnel] RBF untuk beradaptasi dengan d[ata] secara lebih optimal berdasarkan n[ilai] inverse jumlah fitur. Penggunaan ke[rnel] RBF memberikan kemampuan untuk menangkap pola non-linear dalam data.

2) Random Forest (RF):
Parameter terbaik untuk Random Forest adalah criterion='entropy', max_depth=20, max_features='sqrt', dan n_estimators=140. Penggunaan criterion='entropy' meningkatkan pemilihan split berdasarkan pengurangan informasi, sedangkan max_depth=20 memastikan pembelajaran tidak terlalu mendalam untuk mencegah overfitting. Pemilihan max_features='sqrt' meningkatkan efisiensi pada dataset dengan jumlah fitur besar, dan n_estimators=140 memberikan stabilitas prediksi dengan jumlah pohon yang cukup besar.

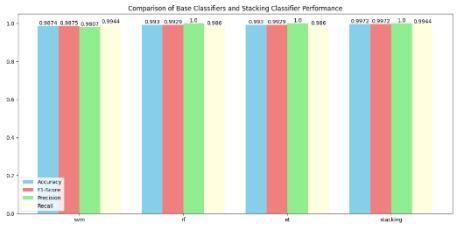
3) Extra Trees (ET):
Parameter terbaik untuk Extra Trees adalah criterion='gini', max_depth =None, max_features='log2', dan n_estimators=200. Penggunaan criterion='gini' berfokus pada memaksimalkan kejelasan kelas, sedangkan max_depth=None memungkinkan pohon tumbuh hingga selesai tanpa batas kedalaman. Pemilihan max_features='log2' menyeimbangkan variasi antar pohon, dan n_estimators=200 memberikan kestabilan prediksi dengan jumlah pohon yang lebih banyak.

Secara keseluruhan, hasil tuning ini memastikan bahwa setiap model dioptimalkan untuk menangkap pola dari dataset secara spesifik, sehingga diharapkan meningkatkan performa klasifikasi secara keseluruhan.

3.1.4 Evaluasi

a. Hasil Prediksi base classifer dan stacking

Model akan dievaluasi dengan acurasi, presisi, f1 score, dan recal. Metrix ini digunakan untuk mengevaluasi kinerja model deteksi secara lebih akurat. Hasil prediksi dapat dilihat pada gambar 3.8.



Gambar 3.8 Hasil Base Classifer dan Stacking

Berdasarkan gambar 3.8 hasil evaluasi model dapat dijelaskan sebagai berikut:

1) Akurasi: Model Stacking memiliki akurasi sebesar 0.9958 untuk kelas malware dan goodware, yang menunjukkan bahwa model ini mampu memprediksi dengan benar 99.58% dari semua data. Model Extra Trees (ET) mencapai akurasi 0.9944, yang berarti 99.44% dari prediksi benar. Model Random Forest (RF) memiliki akurasi 0.9930, yang menunjukkan 99.30% data diprediksi dengan benar. Sementara itu, model SVM memiliki akurasi 0.9748, menunjukkan tingkat keberhasilan prediksi sebesar 97.48%.
2) F1-Score : Model Stacking juga memiliki F1-Score tertinggi, yaitu 0.9958, menunjukkan keseimbangan yang sangat baik antara presisi dan recall. Model ET memiliki F1-Score 0.9944, sedikit di bawah Stacking. Model RF mencapai F1-Score 0.9929, sedangkan SVM memperoleh nilai 0.9750.
3) Presisi: Model Stacking, ET, dan RF mencapai nilai presisi tertinggi sebesar 1.0000, yang berarti semua prediksi positif yang dibuat oleh model adalah

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

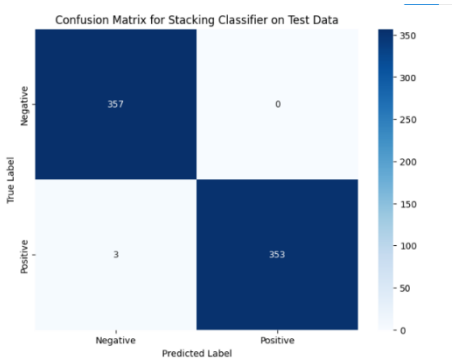
benar. Model SVM memiliki presisi 0.9643, yang berarti dari semua prediksi positif, 96.43% adalah benar.

4) Recall: Recall tertinggi dimiliki oleh Stacking, yaitu 0.9916, menunjukkan kemampuan model untuk mendeteksi sebagian besar data positif. Model RF memiliki recall 0.9888, diikuti oleh KNN dengan recall 0.9860, dan SVM dengan recall 0.9860.

Secara keseluruhan, model Stacking menunjukkan performa terbaik dalam semua metrik evaluasi, dengan prediksi yang sangat akurat, presisi yang sempurna, dan kemampuan tinggi dalam mendeteksi data positif.

b. Confusion Matrix

Pada Proses evaluasi model ini akan menggunakan cofusion matrix untuk mengukur performa dari model pada dataset pengujian. Confusion matrix akan memberikan infomasi mengenai jumlah prediksi yang benar dan salah yang dilakukan model pada kelas malware dan goodmalware. Hasil evaluasi dapat dilihat pada gambar 3.9.



Gambar 3.9 Confusion Matrix Data Test

Berdasarankan Confusion Matrix tersebut :

1) Prediksi benar : pada data kelas malware sebanyak 357 dan 353 pada goodmalware
2) Prediksi salah : pada data kelas malware 0 gagal terdeteksi dan 3 pada goodmalware

Dari penjelasan diatas data prediksi yang salah sangat sedikit yaitu hanya terjadi pada data goodmalware yang berjumlah 3 data. Hal ini menunjukan bahwa mdel sudah sangat baik dalam mendetksi malware maupun goodmalware hal ini masih bisa ditingkatkan agar bisa tedeteksi lebih baik lagi dengan cara, seperti penambahan jumlah data latih, tuning hyperparameter dan crossvalidation-nya.

3.2 Pembahasan

Berdasarkan hasil pengujian dan analisis yang dilakukan, dapat disimpulkan bahwa:

1) Metode stacking ensemble dengan augmentasi data SMOTE dan tuning hyperparameter mampu meningkatkan akurasi deteksi malware pada dataset tidak seimbang. Model yang dikembangkan memiliki tingkat akurasi, presisi, recall, F1-score, dan AUC yang tinggi, menunjukkan kemampuan yang andal dalam mendeteksi malware Android.
2) Pendekatan yang diterapkan berhasil mengatasi masalah ketidakseimbangan data serta meningkatkan ketahanan model terhadap variasi data.

3.2.1 Perbandingan penelitian sebelumnya

1. Penelitian berjudul "**Explainable Classification Model for Android Malware Analysis Using API and Permission-Based Features**" Penelitian ini menerapkan pembelajaran mesin untuk mendeteksi malware Android menggunakan fitur izin dan API dari dataset TUANDROMD. Teknik balancing data seperti RandomOver-

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

Sampler, SMOTETomek, dan RandomUnderSampler diterapkan. Model Extra Tree, Random Forest, dan SVM diuji, dengan Extra Tre RandomOverSampler menca akurasi 99,53% (0,0198 det Explainable AI mengidentifikasi f utama dalam deteksi malware. Studi ini akurat, cepat, dan transparan[13].

2. Penelitian kedua berjudul Jurnal ini berjudul *"**Random Feature Selection Using Random Subspace Logistic Regression**"* oleh Nuttanan Wichitaksorn et al. Penelitian ini mengusulkan metode *random subspace logistic regression* untuk seleksi fitur secara acak melalui simulasi bootstrap, dengan penerapan pada *standard logistic regression* dan *lasso logistic regression*. Metode ini bertujuan mengatasi masalah beban komputasi tinggi pada regresi logistik konvensional, khususnya pada dataset berdimensi tinggi. Evaluasi dilakukan menggunakan data simulasi serta dataset besar dari UCI Machine Learning Repository dan Kaggle. Hasil penelitian menunjukkan bahwa metode ini mampu mengurangi waktu komputasi secara signifikan sekaligus meningkatkan akurasi prediksi dibandingkan metode regresi logistik standar, dengan akurasi terbaik mencapai 98,43% pada dataset TUANDROMD. Penelitian ini dapat menjadi referensi penting dalam penelitian saya, terutama dalam konteks penggunaan dataset TUANDROMD serta pendekatan seleksi fitur dan optimasi algoritma. Meskipun tidak menggunakan metode *stacking ensemble* seperti penelitian saya, pendekatan random subspace ini dapat dipertimbangkan sebagai metode alternatif dalam meningkatkan performa model klasifikasi[14].

3. Penelitian ketiga berjudul "Outsmarting Android Malware with Cutting-Edge Feature Engineering and Machine Learning Techniques" Jurnal ini membahas deteksi malware Android menggunakan teknik machine learning dengan fokus pada preprocessing data, pemilihan fitur, dan algoritma seperti Random Forest (RF), Support Vector Classifier (SVC), serta K-Nearest Neighbors (KNN). Data dari Drebin dan TUANDROMD digunakan sebagai benchmark, dengan preprocessing meliputi penghapusan duplikasi, imputasi nilai hilang, serta normalisasi data menggunakan Min-Max. Fitur penting seperti permissions, API calls, dan intents dipilih menggunakan Gain-Ratio dan Chi-Squared Test. Hasil penelitian menunjukkan akurasi tinggi, di mana RF dan SVC mencapai 98,9% dalam mendeteksi malware. Meskipun menawarkan framework deteksi yang kuat, metode ini masih terbatas pada dataset tertentu dan bergantung pada feature engineering manual[15].

4. Penelitian Keempat berjudul "Jurnal ini berjudul "**An effective deep learning scheme for android malware detection leveraging performance metrics and computational resources**". Penelitian ini mengusulkan penggunaan model deep learning berbasis Deep Neural Decision Forest (DNDF) dan Deep Belief Network (DBN) untuk mendeteksi malware Android. Dua dataset digunakan dalam

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

evaluasi, yaitu Drebin Dataset untuk membandingkan dengan studi sebelumnya dan TUANDROMD Dataset (2021) untuk mendeteksi ancam terbaru dengan teknik obfuscation morphing yang lebih canggih. Fo utama penelitian ini ada membandingkan performa DNDF dengan teknik machine learning lainnya, serta menghitung waktu eksekusi dan konsumsi sumber daya komputasi. Hasil eksperimen menunjukkan bahwa model DNDF mencapai akurasi 99%, sensitivitas 1, dan AUC sebesar 0,98%. Hasil ini setara atau bahkan lebih baik dibandingkan dengan metode berbasis machine learning lainnya dan beberapa antivirus komersial. Studi ini relevan sebagai pembanding dalam penelitian saya, khususnya dalam penggunaan dataset TUANDROMD, meskipun perbedaan utamanya adalah pendekatan deep learning dan tidak digunakannya metode stacking ensemble atau augmentasi data seperti SMOTE [16].

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai pengembangan dan evaluasi metode stacking berbasis ensemble untuk deteksi malware dengan pendekatan augmentasi data pada dataset tidak seimbang, diperoleh beberapa kesimpulan sebagai berikut:

1) Metode stacking ensemble yang dikembangkan berhasil meningkatkan akurasi deteksi malware pada perangkat Android dengan mengintegrasikan optimasi algoritma dan tuning hyperparameter. Kombinasi beberapa algoritma sebagai base classifiers dan Logistic Regression sebagai meta-classifier menghasilkan model dengan performa tinggi, yakni akurasi sebesar 99,58%, presisi 100%, recall 99,16%, F1-score 99,58%, dan AUC 99,5%. Proses tuning hyperparameter menggunakan GridSearchCV berkontribusi signifikan terhadap pemilihan parameter optimal untuk setiap algoritma dasar, sehingga meningkatkan kemampuan generalisasi model.
2) Teknik augmentasi data menggunakan Synthetic Minority Oversampling Technique (SMOTE) terbukti efektif dalam mengatasi ketidakseimbangan dataset antara kelas malware dan goodware. SMOTE meningkatkan representasi kelas minoritas tanpa mengurangi akurasi dalam mendeteksi kelas mayoritas, sehingga model mampu mendeteksi malware maupun goodware secara efektif. Ketahanan model terhadap variasi data juga teruji pada dataset uji eksternal, menunjukkan generalisasi yang baik dalam domain yang sama.

V. SARAN

Berdasarkan hasil penelitian yang telah dilakukan, penulis memberikan beberapa saran untuk penelitian selanjutnya:

a) Pengembangan Metode Deteksi yang Lebih Canggih
   1) Penggunaan Dataset yang Lebih Beragam.
      Disarankan untuk menggunakan dataset yang lebih besar dan mencakup jenis malware yang lebih beragam agar hasil yang diperoleh dapat lebih representatif terhadap kondisi nyata.
   2) Eksplorasi Teknik Deteksi Baru.
      Selain metode stacking ensemble, penelitian masa depan dapat mengeksplorasi penggunaan metode deep learning atau hybrid ensemble untuk meningkatkan akurasi lebih lanjut.

b) Pengembangan Metode Deteksi yang Lebih Canggih

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

1) Penerapan pada Sistem Nyata.
   Penelitian selanjutnya diharapkan dapat mengimplementasikan model yang dikembangkan pada sistem deteksi malware nyata di perangkat Android. Hal ini bertujuan untuk mengevaluasi performa model dalam kondisi operasional yang sesungguhnya.
2) Pengujian pada Berbagai Platform.
   Demi memperluas cakupan penelitian, metode serupa dapat diterapkan pada platform lain seperti iOS, Windows, atau sistem operasi lainnya. Dengan demikian, model yang dikembangkan dapat memiliki cakupan dan dampak yang lebih luas.
3) Pengembangan Teknik Augmentasi Data Lainnya
   Dianjurkan untuk mengeksplorasi teknik augmentasi data lain selain SMOTE, seperti ADASYN atau metode berbasis generatif (GAN), untuk mengatasi ketidakseimbangan dataset secara lebih efektif.
4) Penggunaan Fitur Tambahan untuk Deteksi Malware
   Penelitian mendatang dapat mempertimbangkan penggunaan fitur tambahan, seperti pola perilaku aplikasi, analisis lalu lintas jaringan, atau analisis metadata aplikasi, untuk meningkatkan kemampuan deteksi terhadap malware yang lebih kompleks dan canggih.

Dengan mengacu pada saran-saran tersebut, diharapkan penelitian di masa depan dapat mengembangkan metode deteksi malware yang lebih canggih, efektif, dan relevan dengan kebutuhan keamanan siber.

VI. DAFTAR PUSTAKA

[1] M. Zyout, R. Shatnawi, and H. Najadat, “Malware classification approaches utilizing binary and text encoding of permissions,” *Int. J. Inf. Secur.*, vol. 22, no. 6, pp. 1687–1712, 2023, doi: 10.1007/s10207-023-00712-z.

[2] S. Sharma, R. Kumar, and C. Rama Krishna, “A survey on analysis and detection of Android ransomware,” *Concurr. Comput. Pract. Exp.*, vol. 33, no. 16, pp. 1–24, 2021, doi: 10.1002/cpe.6272.

[3] W. Al-Kahla, E. Taqieddin, A. S. Shatnawi, and R. Al-Ouran, “Malware Detection and Classification in Android Application using Simhash-Based Feature Extraction and Machine Learning,” *IEEE Access*, vol. 12, no. October, pp. 174255–174273, 2024, doi: 10.1109/ACCESS.2024.3501277.

[4] Rafrasta, F. A. Ra, C. Supriyanto, C. Paramita, and Y. P. Astuti, “Deteksi Malware menggunakan Metode Stacking berbasis Ensemble,” *J. Inform. J. Pengemb. IT*, vol. 8, no. 1, pp. 11–16, 2023, doi: 10.30591/jpit.v8i1.4606.

[5] E. B. Soritua and D. N. Utama, “Enhancing Android Malware Detection Through Ensemble Stacking Classifiers and Regularization-Based Feature Selection,” *J. Theor. Appl. Inf. Technol.*, vol. 102, no. 3, pp. 801–811, 2024.

[6] A. Albin Ahmed, A. Shaahid, F. Alnasser, S. Alfaddagh, S. Binagag, and D. Alqahtani, “Android Ransomware Detection Using Supervised Machine Learning Techniques Based on Traffic Analysis,” *Sensors*, vol. 24, no. 1, pp. 1–21, 2024, doi: 10.3390/s24010189.

ESCAF 3rd 2025
p-ISSN : xxx-xxx
e-ISSN:xxx-xxx

[7] G. James, D. Witten, T. Hastie, and R. Tibshirani, *Springer Texts in Statistics An Introduction to Statistical Learning wth application in R*. 2013.

[8] I. Muhamad Malik Ma[illegible] "Hyperparameter Tun[illegible] Menggunakan GridsearchCV p[illegible] Random Forest untuk Deteksi Malware," *Multinetics*, vol. 9, no. 1, pp. 43–50, 2023, doi: 10.32722/multinetics.v9i1.5578.

[9] M. P. Pulungan, A. Purnomo, and A. Kurniasih, "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Kepribadian MBTI Menggunakan Naive Bayes Classifier," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 10, no. 7, pp. 1493–1502, 2023, doi: 10.25126/jtiik.1077989.

[10] F. alan M. Junta Zeniarja, Abu Salam, "seleksi fitur dan perbandingan algoritma klasifikasi untuk prediksi kelulusan mahasiswa," 2022. doi: 10.17529/jre.v18i2.24047.

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

[12] R. B. Hadiprakoso, W. R. Aditya, and F. N. Pramitha, "Analisis Statis Deteksi Malware Android Menggunakan Algoritma Supervised Machine Learning," *Cyber Secur. dan Forensik Digit.*, vol. 5, no. 1, pp. 1–5, 2022, doi: 10.14421/csecurity.2022.5.1.3116.

[13] N. Aslam *et al.*, "Explainable Classification Model for Android Malware Analysis Using API and Permission-Based Features," *Comput. Mater. Contin.*, vol. 76, no. 3, pp. 3167–3188, 2023, doi: 10.32604/cmc.2023.039721.

[14] N. Wichitaksorn, Y. Kang, and F. Zhang, "Random feature selection using random subspace logistic regression," *Expert Syst. Appl.*, vol. 217, no. January, p. 119535, 2023, doi: 10.1016/j.eswa.2023.119535.

[15] A. Wajahat *et al.*, "Outsmarting Android Malware with Cutting-Edge Feature Engineering and Machine Learning Techniques," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 651–673, 2024, doi: 10.32604/cmc.2024.047530.

[16] A. Wajahat *et al.*, "An effective deep learning scheme for android malware detection leveraging performance metrics and computational resources," *Intell. Decis. Technol.*, vol. 18, pp. 1–23, 2024, doi: 10.3233/IDT-230284.