BAB II
TINJAUAN PUSTAKA

2.1 Rujukan Penelitian

Terdapat penelitian-penelitian terdahulu yang akan digunakan sebagai rujukan dalam penelitian kali ini.

Tabel 2.1 Rujukan Penelitian Terdahulu

Judul Penelitian	Tahun	Nama Peneliti	Kesimpulan Penelitian
Hybrid classification of Android malware based on fuzzy clustering and the gradient boosting machine.	2021	Taha, A. A., & Malebary, S. J.	Menggunakan pendekatan Hybrid dalam pengklasifikasian malware Android dengan menggabungkan algoritma Fuzzy C-means clustering dengan LightGBM. Dataset sendiri berjumlah 5560 aplikasi malware dan 123.453 aplikasi benign. Hasil penelitiannya didapatkan model LightGBM memperoleh akurasi sebesar 94,63%, AUC 98,74%, dan Presisi 97,70%
Android malware detection model based on lightGBM.	2020	Wang, G., & Liu, Z.	Mengusulkan model deteksi malware Android dengan menggunakan algoritma LightGBM. Dengan dataset terdiri dari 2000 sampel benign dan 2000 sampel berbahaya. Di terapkan feature selections Chi2 dan Extra Trees. Hasil penelitiannya didapatkan model LightGBM berhasil memperoleh akurasi model sekitar 96,4%
An Efficient Android Malware Prediction Using	2021	Sarah, N. Al, Rifat, F. Y., Hossain, M.	Melakukan pendekatan malware Android prediction menggunakan beberapa algoritma ensemble machine learning. Dengan total feature pada

Ensemble machine learning algorithms		S., & Narman, H. S.	dataset berjumlah 215 feature. Kemudian diterapkannya Recursive Feature Elimination (RFE) dan Recursive Feature Elimination with Cross Validation (RFECV) pada Feature Selections. Dan hasil penelitiannnya didapatkan model LightGBM memperoleh akurasi tertinggi (sebesar 99,5%) dengan menggunakan 100 fitur optimal.
Exploring the Effectiveness and Efficiency of LightGBM Algorithm for Windows Malware Detection.	2022	Onoja, M., Jegede, A., Mazadu, J., Aimufua, G., Oyedele, A., & Olibodum, K.	Menerapkan algoritma LightGBM generik pada malware Android untuk menentukan efisiensi dan efektifitas. Dengan dataset terdiri dari 9.339 sampel dari 25 keluarga malware. Hasil penelitian ini sendiri mendapatkan bahwa model LightGBM berhasil melakukan deteksi pada aplikasi malware dengan waktu training 179,51 detik untuk klasifikasi biner dan 2224,77 detik untuk klasifikasi multi. Waktu deteksi sebesar 0,08 detik dan 0,40 detik untuk klasifikasi biner dan multi-class secara berturut-turut, dengan akurasi klasifikasi sebesar 99% tingkat TP (True Positif).
LightGBM-based Ransomware Detection using API Call Sequences.	2021	Nguyen, D. T., & Lee, S.	Menerapkan teknik Dynamic Analysis dan Machine Learning untuk identifikasi ransomware. Disini diterapkan LightGBM algoritma sebagai klasifikasi malwarenya. Datasetnya sendiri berjumlah 5,811 sampel. Hasil penelitiannya didapatkan bahwa model LightGBM berhasil memperoleh akurasi sekitar

			98,7% dan yang terendah sekitar 89,5%.

Penelitian yang dilakukan oleh Taha, dkk [24], mengusulkan pendekatan hybrid untuk mengklasifikasikan malware Android dengan menggabungkan algoritma fuzzy C-means clustering dengan Light Gradient-boosting Machine. Penelitian tersebut menggunakan metode pengelompokan fuzzy untuk mengelompokkan izin aplikasi Android ke dalam pola-pola tertentu, lalu menggunakan LightGBM untuk mengklasifikasikan aplikasi-aplikasi ini sebagai malware atau benign. Dengan dataset 5560 aplikasi malware dan 123.453 aplikasi benign, penelitian tersebut berhasil memperoleh hasil akurasi sebesar 94,63%, AUC 98,74%, dan presisi 97,70%.

Penelitian yang dilakukan oleh Wang, dkk [25], mengusulkan model deteksi malware Android berbasis LightGBM. Model ini terdiri dari metode seleksi fitur baru, yang mengandung Chi2 dan Extra Trees, serta metode klasifikasi LightGBM. Dataset yang digunakan pada penelitian tersebut terdiri dari 2000 sampel benign dari Baidu Apps Store dan Google Play Store, serta 2000 sampel berbahaya yang diambil dari virusshare[dot]com. Setelah itu, diekstraksi empat fitur yaitu ‘permissions’, ‘actions’, ‘categories’, dan ‘SO file names’ untuk membangun vektor fitur, kemudian disiapkan model seleksi fitur berdasarkan Chi2, yang memilih 500 fitur dengan peringkat tinggi dalam Chi2, dan Extra Trees. Klasifikasinya diimplementasikan dalam Python dan dirancang menggunakan LightGBM. Hasilnya menunjukkan akurasi model yang tinggi (sekitar 96,4%) dan penurunan yang drastis dalam waktu training dibandingkan dengan model yang ada.

Penelitian yang dilakukan oleh Sarah, dkk [26] melakukan prediksi android malware menggunakan beberapa ensemble machine learning algorima dan diantaranya menggunakan algoritma LightGBM. Tujuan penelitian sarah, dkk sendiri melakukan analisa untuk mencari model prediksi yang terbaik dalam mendeteksi android malware dengan diterapkannya Recursive Feature Elimination (RFE) feature selections untuk mengurangi jumlah feature dalam men-optimisasi

waktu dan resource. Penelitian ini tidak menyebutkan berapa total dataset yang digunakan, didapat dari mana, dan hanya menyebutkan bahwa jumlah feature pada datasetnya berjumlah 215 feature. Disini menggunakan 8 kategori jumlah fitur yang nanti akan dipilih, terdapat pilihan jumlah fitur 100, 75, 65, 55, 45, 35, 25, dan 15. Kemudian secara otomatis memilih jumlah fitur yang akan dipilih menggunakan RFE dan RFECV (Recursive Feature Elimination with Cross Validation). Disini terdapat 8 algoritma diantaranya yaitu pada Traditional Machine Learning Algoritma terdapat Logistic Regression, Gaussian Naïve Bayes, SVM, dan DT, kemudian untuk Ensemble Machine Learning Algoritma sendiri terdapat Random Forest, Gradient Boosting Decision Tree, XGBoost, dan LightGBM. Hasilnya menunjukkan bahwa lightGBM mencapai akurasi tertinggi (99,5%) di antara metode ensemble lainnya. Fitur optimal untuk LightGBM adalah 100 dengan akurasi 99,4%, namun dalam Random Forest, jumlah fitur optimal adalah 55 dengan akurasi 99,1%. Jika keakuratan yang lebih tinggi menjadi perhatian, LightGBM bisa menjadi solusi yang lebih baik, tetapi jika efisiensi atau jumlah fitur yang lebih sedikit menjadi perhatian, maka Random Forest memberikan akurasi yang konsisten dan lebih baik [26].

Kemudian terdapat penelitian terdahulu selain rana klasifikasi malware android namun masih dengan pengklasifikasian menggunakan pendekatan metode LightGBM, penelitian yang dilakukan oleh Onoja, dkk [27] menerapkan algoritma LightGBM generik pada malware Windows untuk menentukan efisiensi dan efektivitasnya dalam hal training time, prediction time, dan akurasi klasifikasi. Dataset eksperimental untuk studi malware terdiri dari 9.339 sampel dari 25 keluarga malware. Sampel-sampel malware diubah menjadi gambar skala abu-abu dengan nilai piksel berkisar dari 0 hingga 255. Dataset benign (aman) yang digunakan untuk perbandingan mencakup 1.042 gambar yang diubah dari file executable Windows. Karena model LightGBM generik tidak menerima input gambar, dataset gambar di-"pickled" untuk eksperimen awal. Dataset juga dibagi menjadi set pelatihan dan pengujian. Dalam tugas klasifikasi, sebanyak 10.381 gambar digunakan untuk klasifikasi biner, sementara 26 kelas dengan jumlah gambar yang sama digunakan untuk klasifikasi multi-kelas. Karena LightGBM tidak menerima gambar sebagai input, dataset gambar di-"pickled" dan fitur-fitur

diekstrak secara otomatis menggunakan model LightGBM. Didapatkan hasil dari penelitian ini menunjukkan bahwa algoritma LightGBM dapat diterapkan pada malware Windows dan mampu melakukan deteksi aplikasi malware dengan waktu pelatihan selama 179,51 detik untuk klasifikasi biner dan 2224,77 detik untuk klasifikasi multi. Waktu deteksi sebesar 0,08 detik dan 0,40 detik untuk klasifikasi biner dan multi-kelas secara berturut-turut, dengan akurasi klasifikasi sebesar 99% Tingkat Positif Benar (TPR). Akurasi klasifikasi dan hasil kinerja model dapat ditingkatkan, sementara waktu pelatihan dan waktu prediksi dapat dikurangi.

Selanjutnya penelitian yang dilakukan oleh Nguyen, dkk [28], menggunakan teknik analisis dinamis dan machine learning untuk identifikasi ransomware. Penelitian tersebut fokus pada ekstraksi API Call Sequence dari sampel executable. Datasetnya sendiri terdiri dari 4,008 file benign dari Windows system, 1,373 sampel malware dari virusshare[dot]com, dan 430 sampel dari virustotal[dot]com, total 5,811 sampel. Penelitian tersebut menemukan variasi besar dalam jumlah fungsi API yang digunakan oleh satu sampel, dari 1 hingga 172 fungsi dengan sekitar 286 fungsi yang berbeda. Meskipun ransomware dan file benign menggunakan fungsi API yang sama, penulisnya tetap memakai fungsi itu untuk tujuan yang berbeda. Ransomware cenderung menggunakan lebih dari 100 fungsi API, dengan beberapa fungsi dipanggil ratusan ribu kali selama eksekusi file. Algoritma LightGBM yang penelitian tersebut gunakan berhasil dengan tingkat akurasi sekitar 98,7%. Bahkan, metodologi ini sangat baik dalam mengklasifikasikan tiga keluarga ransomware (LockScreen, WannaCry, dan Win32:FileCoder) tanpa kesalahan. Meskipun, tingkat akurasi untuk mengidentifikasi semua jenis ransomware berbeda-beda, dengan tingkat terendah untuk TeslaCrypt sekitar 89,5%. Perbandingan dengan penelitian sebelumnya menunjukkan metode penelitian tersebut tidak hanya lebih akurat dalam membedakan ransomware dari perangkat lunak benign, tapi juga lebih efisien dalam mengklasifikasikan berbagai jenis ransomware. Penggunaan algoritma LightGBM juga mempercepat waktu pemrosesan dibandingkan dengan algoritma machine learning lainnya.

Dari penelitian-penelitian diatas, penelitian yang ada telah banyak menjelajahi seberapa efektifnya LightGBM. Namun, masih ada kesenjangan dalam integrasi Recursive Feature Elimination (RFE) bersama LightGBM untuk pemilihan fitur. Meskipun beberapa penelitian memanfaatkan LightGBM untuk klasifikasi, hanya sedikit penelitian, seperti yang dilakukan oleh Sarah, dkk[26], yang menggabungkan RFE untuk mengoptimalkan fitur set, menghasilkan hasil yang sangat akurat. Mengintegrasikan RFE dengan LightGBM dapat mengatasi kebutuhan akan pendekatan sistematis dalam pemilihan fitur, yang berpotensi meningkatkan efisiensi model, mengurangi beban komputasi, dan membantu dalam interpretasi. Integrasi ini dapat menutup celah penting dengan menyempurnakan fitur set, yang berpotensi meningkatkan kinerja model dan penggunaan resource, sehingga berkontribusi pada sistem deteksi malware Android yang lebih efisien dan efektif.

2.2 Android

Android merupakan sistem operasi berbasis Linux yang dikembangkan oleh Google untuk perangkat mobile [8]. Android dapat dianggap sebagai platform perangkat lunak yang menyediakan lingkungan komputasi untuk perangkat mobile. Sebagai sistem operasi berbasis Linux, Android menggunakan kernel Linux sebagai dasar inti yang mengendalikan akses ke perangkat keras. Kernel Linux memberikan abstraksi dan kontrol terhadap perangkat keras, sehingga Android dapat berinteraksi dengan komponen perangkat seperti prosesor, memori, dan perangkat jaringan. Android memiliki arsitektur yang terdiri dari lapisan-lapisan yang saling berhubungan. Lapisan terendah adalah kernel Linux yang bertanggung jawab untuk manajemen perangkat keras. Di atasnya, terdapat lapisan yang disebut lapisan perangkat keras abstraksi Hardware Abstraction Layer (HAL) yang menyediakan antarmuka standar untuk mengakses perangkat keras pada berbagai perangkat mobile. Lapisan selanjutnya adalah lapisan Android Runtime (ART), yang bertanggung jawab untuk menjalankan dan mengelola kode aplikasi. ART menggunakan metode kompilasi just-in-time (JIT) atau ahead-of-time (AOT) untuk mengubah kode Java menjadi instruksi mesin yang dapat dieksekusi oleh prosesor. Di atas lapisan ART, terdapat Android Framework yang menyediakan berbagai API dan komponen yang memfasilitasi pengembangan aplikasi. Android Framework

mencakup beragam modul, seperti window management, source management, dan layanan sistem, yang memungkinkan developer untuk membuat aplikasi yang berinteraksi dengan sistem secara efisien. Bagian teratas dari arsitektur Android adalah lapisan Aplikasi, di mana aplikasi pengguna berjalan. Pengembang dapat membangun aplikasi menggunakan Java, Kotlin, atau bahasa pemrograman lain yang kompatibel dengan Java Virtual Machine (JVM). Android menyediakan komponen seperti aktivitas, layanan, dan penyimpanan data, yang memungkinkan pengembang untuk membangun aplikasi yang beragam dan interaktif [8].

Sebagai sistem operasi yang open source, Android juga memiliki develop community yang aktif dan beragam [8]. Komunitas ini berkontribusi pada pengembangan Android melalui pengembangan modul, aplikasi, dan berbagi pengetahuan melalui forum dan proyek open source. Hal ini memungkinkan inovasi dan peningkatan berkelanjutan dalam lingkup Android. Jadi secara keseluruhan, Android merupakan sistem operasi berbasis Linux yang menyediakan lingkungan komputasi untuk perangkat mobile. Android dapat dianggap sebagai platform perangkat lunak yang terdiri dari lapisan-lapisan yang terhubung satu sama lain, dengan kernel Linux sebagai dasar inti dan beragam modul serta komponen yang memungkinkan pengembangan aplikasi yang beragam.

2.3 Malware

Malware, yang merupakan singkatan dari "malicious software", adalah jenis perangkat lunak yang dirancang dengan niat jahat untuk menginfeksi sistem komputer atau perangkat lainnya, dengan tujuan menyebabkan kerusakan, mencuri data, atau memperoleh akses tidak sah. Bahkan malware saat ini dapat dengan mudah mem-bypass perangkat lunak perlindungan yang ada berjalan dalam mode kernel seperti firewall, perangkat lunak antivirus, dll [9]. Malware sendiri ada beberapa jenis yaitu sebagai berikut.

1. *Virus:* Virus adalah jenis malware yang menggandakan dirinya sendiri dengan menyisipkan salinan dirinya ke dalam program atau file lain. Ketika program atau file yang terinfeksi dijalankan, virus dapat menyebar dan merusak sistem dengan menggantikan atau mengubah file yang ada. Virus

seringkali memanfaatkan celah keamanan untuk menyebar dan dapat menyebabkan kerusakan yang signifikan pada sistem.

2. *Worm:* Worm adalah jenis malware yang dapat menyebar sendiri melalui jaringan tanpa memerlukan interaksi pengguna. Worm memanfaatkan kerentanan sistem atau celah keamanan untuk menginfeksi komputer yang terhubung dalam jaringan yang sama. Mereka dapat mengirim salinan diri mereka melalui jaringan, mengkonsumsi sumber daya jaringan, dan mengganggu operasi sistem.
3. *Trojan:* Trojan atau Trojan horse adalah jenis malware yang menyembunyikan dirinya dalam program atau file yang tampak sah atau bermanfaat. Ketika program atau file tersebut dijalankan, Trojan dapat membuka pintu belakang pada sistem yang terinfeksi, memungkinkan penyerang untuk mengendalikan komputer dari jarak jauh, mencuri informasi sensitif, atau memperkenalkan malware tambahan.
4. *Spyware:* Spyware adalah jenis malware yang dirancang untuk memantau dan mengumpulkan informasi tentang aktivitas pengguna tanpa persetujuan mereka. Spyware dapat mencuri informasi pribadi, seperti kata sandi, informasi keuangan, atau riwayat penelusuran internet, dan mengirimkannya ke pihak yang tidak berwenang. Biasanya, spyware tersembunyi dalam perangkat lunak atau aplikasi yang tampaknya sah.
5. *Ransomware:* Ransomware adalah jenis malware yang mengenkripsi file pada sistem komputer dan meminta tebusan untuk memberikan kunci dekripsi. Ransomware dapat mencegah pengguna mengakses data mereka, dan penyerang akan menuntut pembayaran dalam bentuk mata uang digital agar memberikan kunci dekripsi. Serangan ransomware dapat menyebabkan kerugian finansial dan kerugian data yang signifikan.
6. *Adware:* Adware adalah jenis malware yang menghasilkan iklan yang tidak diinginkan pada sistem komputer atau perangkat pengguna. Iklan tersebut seringkali muncul secara agresif dan mengganggu pengalaman pengguna. Adware seringkali terpasang bersamaan dengan perangkat lunak atau aplikasi gratis yang diunduh dari sumber yang tidak tepercaya.

7. *Keylogger:* Keylogger adalah jenis malware yang merekam setiap keystroke yang dilakukan oleh pengguna pada sistem komputer atau perangkat lainnya. Keylogger dapat mencatat kata sandi, informasi pribadi, dan data sensitif lainnya yang diketik oleh pengguna, yang kemudian dapat diakses oleh penyerang untuk tujuan jahat.
8. *Botnet:* Botnet adalah jaringan komputer yang terinfeksi oleh malware dan dikendalikan oleh penyerang secara jarak jauh. Komputer dalam botnet, yang disebut bot, dapat digunakan untuk menjalankan serangan koordinasi, seperti serangan DDoS (Distributed Denial of Service), spam, atau kegiatan ilegal lainnya tanpa pengetahuan pemilik komputer.

2.4 Machine Learning

Machine Learning adalah cabang artificial intelligence (AI) dan ilmu komputer yang berfokus pada penggunaan data dan algoritme untuk meniru cara manusia belajar, secara bertahap meningkatkan akurasinya [10]. Menurut penjelasan Katrina [11] algoritma machine learning sendiri memiliki 4 jenis, yaitu.

1. *Supervised*

   Dalam *supervised learning*, mesin diajarkan melalui contoh. Operator menyediakan algoritme machine learning dengan kumpulan data yang diketahui yang menyertakan input dan output yang diinginkan, dan algoritme harus menemukan metode untuk menentukan cara mendapatkan input dan output tersebut. Sementara operator mengetahui jawaban yang benar untuk masalah tersebut, algoritme mengidentifikasi pola dalam data, belajar dari pengamatan, dan membuat prediksi. Algoritme membuat prediksi dan dikoreksi oleh operator – dan proses ini berlanjut hingga algoritme mencapai tingkat akurasi/kinerja yang tinggi.

   Di dalam supervised learning sendiri terdapat: *Classification, Regression*, dan *Forecasting*

2. *Semi-supervised learning*

Semi-supervised learning mirip dengan supervised learning, tetapi menggunakan data berlabel dan tidak berlabel. Data berlabel pada dasarnya adalah informasi yang memiliki tag yang bermakna sehingga algoritme dapat memahami data tersebut, sedangkan data yang tidak berlabel tidak memiliki informasi tersebut. Dengan menggunakan ini kombinasi, algoritme pembelajaran mesin dapat belajar memberi label pada data yang tidak berlabel.

3. *Unsupervised learning*

   Di sini, algoritma machine learning mempelajari data untuk mengidentifikasi pola. Tidak ada answer key atau operator manusia untuk memberikan instruksi. Sebaliknya, mesin menentukan korelasi dan hubungan dengan menganalisis data yang tersedia. Dalam proses pembelajaran tanpa pengawasan, algoritme pembelajaran mesin dibiarkan menginterpretasikan kumpulan data besar dan mengalamatkan data tersebut sesuai dengan itu. Algoritma mencoba mengatur data itu dengan cara tertentu untuk menggambarkan strukturnya. Ini mungkin berarti mengelompokkan data ke dalam kelompok atau mengaturnya dengan cara yang terlihat lebih teratur.

   Di dalam unsupervised learning sendiri terdapat: *Clustering* dan *Dimension Reduction*

4. *Reinforcement learning*

   Pembelajaran penguatan berfokus pada proses pembelajaran yang diatur, di mana algoritma pembelajaran mesin dilengkapi dengan serangkaian tindakan, parameter, dan nilai akhir. Dengan mendefinisikan aturan, algoritme pembelajaran mesin kemudian mencoba mengeksplorasi opsi dan kemungkinan yang berbeda, memantau dan mengevaluasi setiap hasil untuk menentukan mana yang optimal. Pembelajaran penguatan mengajarkan trial and error mesin. Itu belajar dari pengalaman masa lalu dan mulai menyesuaikan pendekatannya dalam menanggapi situasi untuk mencapai hasil yang terbaik.

2.5 Light Gradient Boosting Machine (LightGBM)

LightGBM, kependekan dari *light gradient-boosting machine*, adalah framework peningkatan gradien terdistribusi gratis dan open source untuk machine learning, yang dikembangkan oleh Microsoft dan direlease sejak tahun 2016, LightGBM ini didasarkan pada algoritme decision tree dan digunakan untuk ranking, klasifikasi, dan tugas machine learning lainnya [14]. LightGBM dioptimalkan sebagai berkinerja tinggi dengan sistem terdistribusi [12]. LightGBM juga sesuai penjelasan [12] menciptakan decision trees yang tumbuhnya membentuk pola daun (leaf wise), yang mana memberikan kondisi hanya satu daun (leaf) yang ter-split, dan tergantung pada yang diperoleh. *Leaf-wise tree* terkadang bisa overfitting terkhusus terhadap dataset yang sedikit. Membatasi tree depth bisa membantu menghindari overfitting. LightGBM menggunakan metode berbasis histogram di mana data dimasukkan ke dalam keranjang menggunakan histogram distribusi. Keranjangnya, alih-alih setiap data point, digunakan untuk iterate, menghitung perolehan, dan split data. Metode ini juga dapat dioptimalkan untuk dataset yang jarang. Karakteristik lain dari LightGBM adalah bundling fitur eksklusif di mana algoritme menggabungkan fitur eksklusif untuk mengurangi dimensi, menjadikannya lebih cepat dan lebih efisien [12].

Gradient-based One Side Sampling (GOSS) digunakan untuk pengambilan sampel dataset di LightGBM. GOSS memberi bobot poin data dengan gradien yang lebih besar lebih tinggi sambil menghitung gainnya. Dalam metode ini, contoh yang belum digunakan dengan baik untuk pelatihan berkontribusi lebih banyak. Titik data dengan gradien yang lebih kecil akan dihapus secara acak dan sebagian dipertahankan untuk menjaga akurasi. Metode ini biasanya lebih baik daripada pengambilan sampel acak dengan laju pengambilan sampel yang sama [13].