DETEKSI *MALWARE* ANDROID MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)

SKRIPSI

Diajukan Untuk Melengkapi Salah Satu Syarat

Memperoleh Gelar Sarjana Ilmu Komputer

OLEH:

HIDAYATULLAH

09011182025024

JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
2024

LEMBAR PENGESAHAN

DETEKSI *MALWARE* ANDROID MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)

SKRIPSI

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Ilmu Komputer

OLEH:
HIDAYATULLAH
09011182025024

Mengetahui,

Palembang, 22 Mei 2024

Ketua Jurusan Sistem Komputer	**Pembimbing Tugas Akhir**
Dr. Ir. H. Sukemi, M.T.	**Prof. Deris Stiawan, M.T., Ph.D.**
NIP.196612032006041001	**NIP. 197806172006041002**

AUTHENTICATION PAGE

ANDROID MALWARE DETECTION USING CONVOLUTIONAL NEURAL NETWORK (CNN)

SKRIPSI

Submitted To Complete One Of The Requirements For Obtaining A Bachelor's Degree in Computer Science

By:
HIDAYATULLAH
09011182025024

Acknowledge,

Palembang, 22 Mei 2024

Head Of Computer System Department	**Final Project Advisor**
Dr. Ir. H. Sukemi, M.T.	**Prof. Deris Stiawan, M.T., Ph.D.**
NIP.196612032006041001	**NIP. 197806172006041002**

HALAMAN PERSETUJUAN

Telah diuji dan lulus pada :

Hari : Rabu
Tanggal : 22 Mei 2024

Tim Penguji

1. **Ketua** : **Kemahyanto Exaudi, M.T.**
2. **Sekretaris** : **Iman Saladin B. Azhar, M.MSI**
3. **Pembimbing** : **Prof. Deris Stiawan, M.T., Ph.D.**
4. **Penguji** : **Ahmad Heryanto, M.T.**

Mengetahui, 22/5/24
Ketua Jurusan Sistem Kompter

Dr. Ir. H. Sukemi, M.T.
NIP. 196612032006041001

HALAMAN PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : Hidayatullah

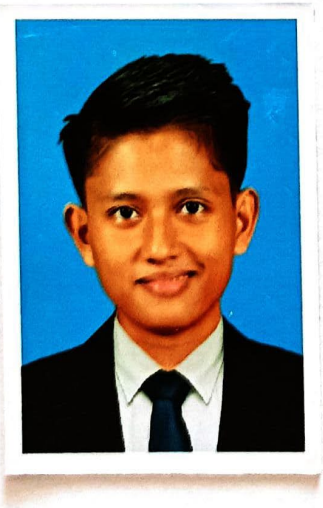
Nim : 09011182025024

Judul : Deteksi *malware* android menggunakan metode convolutional neural network (CNN)

Hasil Pengecekan Software iThenticate/Turnitin : 3%

Menyatakan bahwa laporan tugas akhir saya merupakan hasil karya saya sendiri dan bukan hasil penjiplakan/plagiat. Apabila ditemukan unsur penjiplakan/plagiat dalam laporan tugas akhir ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya.

Demikian pernyataan ini saya buat dengan sebenarnya dan tidak ada paksaan dari siapapun.



Palembang, 2 Mei 2024

Hidayatullah
Nim. 09011182025024

KATA PENGANTAR

Assalamu'alaikum Warrahmatullahi Wabarakatuh.

Alhamdulilahirabbil'alamin, puji dan syukur penulis panjatkan kepada Allah SWT yang telah melimpahkan nikmat, taufik, dan hidayah-Nya yang sangat besar dan tidak pernah berhenti kepada penulis sehingga penulis dapat menyelesaikan Proposal Tugas Akhir ini yang berjudul "**Deteksi *Malware* Android Menggunakan Metode *Convolutional Neural Network* (CNN)"**

Pada kesempatan ini, dengan segala kerendahan hati penulis mengucapkan rasa syukur kepada Allah SWT, dan rasa terima kasih kepada semua pihak atas bantuan, bimbingan, dan saran yang telah diberikan dalam menyelesaikan Proposal Tugas Akhir ini, antara lain:

1. Kedua Orang Tua tercinta yang sudah membesarkan dengan penuh kasih sayang dan terimakasih untuk segala doa, motivasi dan dukungannya baik moril, materil maupun spiritual selama ini.

2. Bapak Prof. Dr. Erwin, S. Si., M. Si selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.

3. Bapak Ir. Sukemi, M.T., selaku Ketua Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.

4. Bapak Prof. Deris Stiawan, M.T., Ph.D., selaku Dosen Pembimbing Tugas Akhir yang telah berkenan meluangkan waktunya dalam membimbing, memberikan saran, dan motivasi kepada penulis dalam penyusunan Proposal Tugas Akhir.

5. Bapak Iman Saladin B. Azhar, S.Kom., M.MSI., selaku Dosen Pembimbing Akademik Jurusan Sistem Komputer.

6. Bapak Yopi Syaputra selaku Admin Jurusan Sistem Komputer yang telah membantu administrasi dalam menyelesaikan Tugas Akhir.

7. Rekan – rekan penulis yang senantiasa membantu dan memberikan saran kepada penulis selama penyusunan laporan skripsi.

8. Kakak-kakak tingkat yang menjadi panutan sekaligus mentor dan seluruh teman-teman seperjuangan angkatan 2020 Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.

Penulis menyadari bahwa Proposal Tugas Akhir ini masih sangat jauh dari kata sempurna. Untuk itu, kritik dan saran sangat penting bagi penulis. Akhir kata, semoga Proposal Tugas Akhir ini dapat bermanfaat dan berguna bagi khalayak.

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Palembang, 2 Mei 2024

Penulis,

HIDAYATULLAH

NIM. 09011182025024

DETEKSI *MALWARE* ANDROID MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK* (CNN)

HIDAYATULLAH (09011182025024)

Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya

Email : hidayatullahdayat2002@gmail.com

ABSTRAK

Sebagai sistem operasi *mobile* yang paling banyak digunakan, android semakin menjadi sasaran utama bagi serangan *malware*. Popularitas sistem operasi ini membuatnya menjadi menarik bagi para penjahat keamanan *cyber* untuk mencuri data berharga, salah satunya dengan cara menginstal aplikasi *malware* android. Beberapa penelitian telah menggunakan berbagai metode *Machine Learning* (ML) untuk mengenali aplikasi *malware* android dari aplikasi jinak. Namun metode ini tidak mampu mendeteksi aplikasi *malware* android yang lebih baru dan canggih. Oleh karena itu, pada penelitian ini disajikan sebuah pendekatan *Deep Learning (DL)* untuk mendeteksi aplikasi *malware* android menggunakan metode *Convolutional Neural Network* (CNN). Eksperimen dilakukan dan diuji pada 20,000 aplikasi *malware* dan 9,999 aplikasi jinak dengan menggunakan 173 fitur izin yang diambil dari aplikasi tersebut. Penelitian ini mencakup beberapa metrik kinerja seperti akurasi, presisi, recall, f1-score dalam mengidentifikasi klasifikasi dengan kinerja terbaik. *Dataset* juga dilakukan *oversampling* untuk mengatasi data yang tidak seimbang. Pada akhir studi didapat hasil bahwa deteksi aplikasi *malware* pada sistem operasi android menggunakan metode CNN mendapat akurasi mencapai 99.99%, dengan presisi sebesar 99.98%, recall 100% dan f1-score sebesar 99.99% dengan nilai error terbaik sebesar 0.0054 %. Hasil kinerja yang diamati dari model ini lebih baik dibandingkan dengan hasil yang dilaporkan dalam penelitian sebelumnya tentang deteksi *malware* android berbasis *machine learning* maupun *deep learning*.

Kata Kunci : Deteksi *Malware* Android, *Malware*, *Deep Learning*, Fitur Izin, *Convolutional Neural Network*

Mengetahui,

Ketua Jurusan Sistem Komputer

Dr. Ir. H. Sukemi, M.T.
NIP.196612032006041001

Palembang, 24 Juni 2024
Pembimbing Tugas Akhir

Prof. Deris Stiawan, M.T., Ph.D.
NIP. 197806172006041002

ANDROID MALWARE DETECTION USING CONVOLUTIONAL NEURAL NETWORK (CNN) METHOD

HIDAYATULLAH (09011182025024)

Department of Computer System, Faculty of Computer Science, Sriwijaya University

Email : hidayatullahdayat2002@gmail.com

ABSTRACT

As the most widely used mobile operating system, Android is increasingly becoming a prime target for malware attacks. The popularity of this operating system makes it attractive for cyber security criminals to steal valuable data, one of which is by installing Android malware applications. Several studies have used various Machine Learning (ML) methods to recognize Android malware applications from benign applications. However, this method is not able to detect newer and more sophisticated Android malware applications. Therefore, this research presents a Deep Learning (DL) approach to detect Android malware applications using the Convolutional Neural Network (CNN) method. Experiments were conducted and tested on 20,000 malware applications and 9,999 benign applications using 173 permission features taken from the applications. This research includes several performance metrics such as accuracy, precision, recall, f1-score in identifying the classifier with the best performance. The dataset was also oversampled to overcome imbalanced data. At the end of the study, the results showed that detecting malware applications on the Android operating system using the CNN method achieved an accuracy of 99.99%, with a precision of 99.98%, a recall of 100% and an f1-score of 99.99% with the best error value of 0.0054%. The performance results observed from this model are better than the results reported in previous research on machine learning and deep learning based Android malware detection.

Keyword : ***Android Malware Detection***, ***Malware***, ***Deep Learning***, **Permission**, ***Convolutional Neural Network***

Acknowledge,

Palembang, 24 Juni 2024

Head Of Computer System Department	**Final Project Advisor**
Dr. Ir. H. Sukemi, M.T.	**Prof. Deris Stiawan, M.T., Ph.D.**
NIP.196612032006041001	**NIP. 197806172006041002**

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1. Latar Belakang

Sejak penemuan *malware* Android pada tahun 2009, jumlah *malware* yang mengincar perangkat Android telah berkembang dengan pesat[1]. Dikarenakan kenyamanan dan efisiensi yang ditawarkan oleh berbagai aplikasi, serta peningkatan perangkat keras dan perangkat lunak pada smartphone, data menunjukkan bahwa jumlah pengguna smartphone diperkirakan mencapai 4,3 miliar pada tahun 2023[2]. Berdasarkan sebuah studi yang diterbitkan oleh Statcounter, Android mendominasi pangsa pasar smartphone dengan persentase 70,46%, sedangkan iOS memiliki pangsa pasar sebesar 28,83% pada Q3 2023[3].

Berdasarkan meningkatnya popularitas perangkat seluler dan sistem operasi Android tersebut, perangkat seluler berbasis Android telah menjadi target yang cocok bagi para penyerang. Selain itu, karena kebijakan kernel sumber terbuka Android, para penyerang dapat menganalisis sistem operasi Android secara rinci untuk merancang *malware* yang canggih[4]. Menurut laporan ilmiah dan bisnis, kira-kira 1 juta file *malware* dibuat setiap hari, dan kejahatan dunia maya diperkirakan akan merugikan ekonomi dunia sekitar $6 triliun setiap tahunnya sejak tahun 2021[5]. Disisi lain laporan terbaru yang dipublikasi oleh Kaspersky, sebuah perusahaan keamanan *cyber* global yang menyatakan bahwa pada tahun Q3 2023 ini terdapat 5,704,599 aplikasi *malware*,adware maupun riskware *mobile* yang berhasil di blokir dimana sebanyak 370,327 paket instalasi berbahaya terdeteksi[6].

Banyak aplikasi Android dapat diunduh secara gratis melalui pasar aplikasi baik secara resmi ataupun pada pihak ketiga, sehingga penggunaan aplikasi gratis ini sangat umum. Peretas dapat mencoba meretas aplikasi ini untuk mengakses data pengguna secara besar-besaran atau melakukan aktivitas ilegal, terutama jika tidak ada mekanisme keamanan yang memadai diimplementasikan dalam aplikasi[7]. Sebuah penelitian terkini mengungkap bahwa mayoritas *malware* Android sering diubah ulang (*repackaged*) ke dalam

aplikasi yang sah agar dapat mengelak dari penghalang keamanan. Dalam proses *repackaging*, para pembuat *malware mobile* mengunduh aplikasi sah yang populer dari Google Play Store, mendekompilasi aplikasi tersebut, menyuntikkan konten jahat, dan akhirnya, mengunggah kembali aplikasi yang telah dimodifikasi ke pasar pihak ketiga agar dapat diunduh oleh pengguna. Dengan banyaknya aplikasi *mobile*, menjadi krusial untuk secara cepat dan otomatis menganalisis serta memeriksa aplikasi yang ada di pasar. Diperlukan pembangunan sistem otomatis yang dapat mengenali dan mendeteksi aplikasi berbahaya dengan tujuan menghapusnya dari pasar resmi maupun pasar non-resmi, sehingga tidak dapat diunduh lebih lanjut[8]. Namun, kendala pengawasan dari instansi terkait sering kali terbatas, sehingga beberapa aplikasi berisiko dapat merambah pasar, dan penyebaran luas melalui unduhan oleh pengguna telah menyebabkan kerugian substansial bagi mereka[9].

Ketika muncul laporan tentang keluarga *malware* Android yang baru, tidak semua platform aplikasi mampu memberikan tanggapan dengan cepat. Sementara itu, beberapa pengguna mungkin memilih untuk mengunduh aplikasi dari situs web pihak ketiga, yang dimana keamanannya tidak dapat dijamin[10]. Guna mencegah masuknya *malware*, Google menyediakan beberapa mekanisme untuk menemukan aplikasi Android berbahaya selama proses pengajuan aplikasi di repositori aplikasi. Beberapa aplikasi Android mengunduh konten berbahaya setelah diinstal, sehingga sulit untuk dideteksi menggunakan pendekatan deteksi *malware* Android yang dimiliki oleh Google[11]. Sebagai contoh, game Android yang dikenal sebagai *Angry Birds* di-hack, dan peretas berhasil masuk ke dalam file APK-nya dan menyematkan kode berbahaya yang mengirim pesan teks tanpa sepengetahuan pengguna. Biayanya adalah 15 GBP untuk setiap pesan kepada pengguna. Lebih dari seribu pengguna terkena dampaknya[12].

Untuk mengatasi masalah ini, banyak teknik deteksi *malware* Android lainnya telah diusulkan. Setiap teknik memiliki kelebihan dan kekurangan masing-masing. Namun, Google selalu merekomendasikan pengguna untuk mengunduh aplikasi Android hanya dari Google Play Store, yang juga memberikan keamanan terhadap aplikasi selama waktu pengajuan dari para

pengembang aplikasi. Karena sifat terbatas dan masalah kompatibilitas perangkat dari Google Play Store, sebagian besar pengguna menggunakan toko aplikasi pihak ketiga untuk mengakses aplikasi. Namun, hanya sedikit dari toko tersebut memungkinkan pengembang untuk menempatkan aplikasi mereka setelah memverifikasi ketiadaan ancaman. Dengan demikian, ada kemungkinan besar aplikasi berbahaya diunduh ke ponsel pintar pengguna tanpa pengetahuan mereka. Aplikasi Android yang di verifikasi baik dari Google Play Store maupun toko aplikasi pihak ketiga tidak selalu menjamin bahwa aplikasi tersebut bebas dari *malware*[11].

Di sisi lain, Google memperkenalkan *"Bouncer"* sebagai layanan pada tahun 2012 melalui platform resmi Android, yaitu Google Play, dengan tujuan melakukan pemindaian otomatis terhadap aplikasi baru dan yang diunggah. Meskipun Bouncer memberikan lapisan tambahan keamanan, namun terdapat beberapa keterbatasan. Pertama, Bouncer hanya dapat melakukan pemindaian aplikasi Android untuk periode tertentu, sehingga memungkinkan aplikasi berbahaya menghindari deteksi dengan tidak melakukan aktivitas berbahaya selama fase pemindaian. Kedua, saat diuji oleh Bouncer, tidak boleh ada kode berbahaya yang terkandung dalam pemasangan awal. Oleh karena itu, aplikasi berbahaya memiliki peluang lebih besar untuk terhindar dari deteksi oleh Bouncer. Setelah berhasil melewati proses pemindaian keamanan Bouncer dan diinstal pada perangkat Android pengguna, aplikasi berbahaya dapat mengunduh kode berbahaya tambahan untuk dijalankan atau terhubung ke server kontrolnya, sehingga dapat dikendalikan dari jarak jauh untuk mengunggah data yang diambil atau menerima instruksi tambahan[13].

Dalam proses pembaruan dan pengembangan sistem, para pengembang Android telah memberikan perhatian besar terhadap peningkatan fungsi keamanan. Sebagai contoh, Android Q, yang dirilis pada tahun 2019, menambahkan beberapa fitur keamanan baru, termasuk enkripsi berbasis file, kontrol akses untuk informasi sensitif, pengaturan akses untuk kamera/mikrofon latar belakang, mode kunci, pencadangan terenkripsi, dan mekanisme yang dikenal sebagai Google Play Protect. Android Q dirancang untuk melindungi privasi dan keamanan pengguna dari berbagai aspek. Sistem

ini juga menyertakan mekanisme kontrol izin yang ditingkatkan, memberikan lebih banyak kendali kepada pengguna terkait penyingkapan lokasi, melarang aplikasi latar belakang untuk memulai aktivitas, membatasi akses aplikasi terhadap identifikasi perangkat yang tidak dapat diatur ulang (seperti IMEI dan nomor seri), serta mengaktifkan randomisasi alamat MAC secara default[14][15].

Selain itu sebagai tindakan keamanan pada platform Android, setiap aplikasi wajib memiliki identifikasi unik, nama paket, dan tanda tangan dengan sertifikat pengembang[16][17]. Sistem ini memastikan bahwa hanya aplikasi yang berasal dari pengembang yang sama yang dapat diinstal di perangkat Android. Selain itu, instalasi APK yang tidak memiliki tanda tangan tidak dimungkinkan. Kebijakan keamanan mendorong para pembuat *malware* untuk menciptakan berbagai varian *malware* dengan menggunakan nama paket dan sertifikat acak[17].

Pada awalnya, *malware* android diciptakan dengan tujuan yang sederhana sehingga lebih mudah terdeteksi. Jenis *malware* ini dapat disebut sebagai *malware* tradisional yang sederhana[5]. *Malware* ini dapat diteksi dengan mudah menggunakan *scanner* anti- *malware* yang biasanya pendeteksian tersebut berbasis tanda tangan pada aplikasi. Untuk menilai efisiensi *scanner* anti- *malware mobile*, penelitian lain menerapkan berbagai teknik obfuscation pada sepuluh aplikasi berbahaya dari enam keluarga yang berbeda, dan menjalankan biner obfuscated ini melawan sepuluh *scanner* anti- *malware* terkenal. Hasilnya menunjukkan bahwa tidak ada dari *scanner* anti- *malware* yang dapat mendeteksi aplikasi berbahaya tersebut[8].

Namun, saat ini, *malware* generasi baru muncul, yang dapat beroperasi dalam mode kernel, lebih merusak, dan lebih sulit dideteksi dibandingkan dengan *malware* tradisional. *Malware* generasi baru dapat dengan mudah mengelabui perangkat lunak perlindungan yang berjalan dalam mode kernel, seperti firewall dan perangkat lunak antivirus. Secara umum, *malware* tradisional terdiri dari satu proses dan tidak menggunakan teknik yang rumit untuk menyembunyikan dirinya. Sebaliknya, *malware* generasi baru menggunakan beberapa proses yang sudah ada atau baru pada saat yang sama,

dan mengimplementasikan beberapa teknik tersembunyi untuk menyembunyikan diri dan tetap bertahan dalam sistem. *Malware* generasi baru dapat melancarkan serangan yang lebih merusak, seperti serangan yang ditargetkan dan persisten yang belum pernah terjadi sebelumnya, dan dapat menggunakan lebih dari satu jenis *malware* selama serangan tersebut[5]. Sebagian besar aplikasi *malware* saat ini memerlukan metode deteksi *malware* yang lebih canggih[18].

Aplikasi pada ponsel pintar dapat dimanfaatkan untuk beragam keperluan, seperti mencari lokasi, melakukan transfer uang, berkomunikasi dengan teman, menonton film, dan berbagai fungsionalitas lainnya. Agar dapat menggunakan fitur-fitur tersebut, pengguna ponsel pintar perlu memberikan izin kepada aplikasi untuk mengakses perangkat telepon, GPS, internet, dan kamera[19][20]. Sayangnya, dengan memanfaatkan izin aplikasi tersebut, para pelaku kejahatan daring secara terus-menerus mengembangkan aplikasi yang terinfeksi *malware*[20]. *Malware* Android terinstal dan berjalan di ponsel tanpa memberi tahu pengguna secara tegas atau tanpa izin pengguna. Terkadang, pengguna memberikan izin tanpa menyadari kebutuhan yang sebenarnya, menciptakan beberapa kerentanan. Peretas dan pengembang aplikasi jahat juga berupaya mencuri data pengguna dengan memaksa mereka memberikan izin tertentu. Referensi menyatakan bahwa sebagian besar masalah keamanan dalam aplikasi seluler disebabkan oleh tindakan pengguna, yang dapat diminimalkan dengan pengembangan aplikasi menggunakan kode sumber yang kurang rentan[7]

Umumnya, *malware* android memiliki satu atau lebih dari perilaku berikut: instalasi paksa, peretasan browser, pencurian dan perubahan data pengguna, pengumpulan informasi pengguna dengan maksud jahat, instalasi berbahaya, pengikatan berbahaya, dan perilaku berbahaya lainnya. Perilaku-perilaku ini dapat serius melanggar hak pengguna dan bahkan dapat menyebabkan kerugian yang substansial bagi mereka. Berdasarkan perilaku-perilaku ini, *malware* Android dapat dibagi menjadi empat kategori, termasuk instalasi *malware* (misalnya, repacking, serangan pembaruan, dan pengunduhan tanpa izin), aktivasi *malware*, muatan berbahaya (misalnya, eskalasi hak istimewa, kontrol

jarak jauh, biaya finansial, dan pengumpulan informasi), dan penyalahgunaan izin[21].

Android memberikan hak istimewa kepada pengguna untuk memberikan atau mencabut izin selama proses instalasi, sebuah fitur yang tidak ada pada versi Android sebelumnya. Guna mencegah penyalahgunaan izin, Google memperkenalkan fasilitas ini pada Android 6.0 (API 23), atau yang dikenal sebagai Marshmallow. Kelemahan dalam model izin aplikasi Android telah menarik perhatian para pelaku kejahatan siber, yang kemudian mengembangkan aplikasi berbahaya dan memanfaatkan data pribadi pengguna[22]. Untuk membedakan *malware* dari aplikasi yang baik, pengguna memerlukan pengetahuan yang sangat metodis. Izin yang sama, diperlukan untuk aplikasi baik ataupun berbahaya, sehingga tidak dapat dibedakan oleh sistem berbasis izin ini. Secara umum, metodologi berbasis izin pada umumnya tidak dikembangkan untuk deteksi *malware*, tetapi digunakan untuk penilaian risiko[13].

Kesalahan dapat terjadi baik dari pengguna saat menggunakan aplikasi maupun juga oleh pengembang saat mengembangkan aplikasi. Pengembang juga sering membuat kesalahan dengan tidak menjalankan proses pengujian dan validasi yang ekstensif sejak awal siklus pengembangan aplikasi. Meskipun ada mekanisme yang tersedia, banyak pengembang aplikasi seluler tidak memprioritaskan penulisan kode yang aman saat mengembangkan aplikasi. Kesalahan pengembang ini dapat mengakibatkan kerentanan, termasuk penyebutan izin yang tidak diinginkan dalam file *AndroidManifest.xml*. Hal ini dapat menimbulkan kerentanan pada aplikasi seluler, terutama jika izin tersebut memiliki tingkat risiko yang tinggi. Pengguna kemudian diharuskan memberikan izin tersebut, yang dapat menyebabkan penolakan aplikasi oleh sejumlah pengguna[23]. Kerentanan semacam ini mungkin tidak terdeteksi saat aplikasi diunggah ke Google Play, karena Google Play tidak melakukan analisis menyeluruh terhadap kode aplikasi seluler saat proses unggah, berbeda dengan praktik yang diterapkan di Apple App Store[24]. Oleh karena itu, perlu mengintegrasikan mekanisme deteksi kerentanan dalam kode sumber aplikasi secara tepat.

Untuk itu pada penelitian ini akan dilakukan pendeteksian *malware* android dengan menggunakan metode *convolutional neural network* pada sebuah *dataset* berupa kumpulan aplikasi jinak dan *malware* baik dari platform resmi maupun pihak ketiga, dimana berisi informasi mengenai nama aplikasi, paket, kategori, peringkat, dan harga. Serta aplikasi di compile untuk mendapatkan fitur izin yang di kelompokkan berdasarkan kategori berbahaya atau aman izin tersebut.

1.2. Perumusan dan Batasan Masalah

Adapun perumusan masalah dan batasan masalah yang akan dilakukan dalam penelitian tugas akhir ini adalah sebagai berikut :

1.2.1. Perumusan Masalah

Adapun rumusan masalah dari penelitian tugas akhir ini, yaitu :

1. Bagaimana mendeteksi *malware* pada sistem operasi android menggunakan metode *Convolutional Neural Network* (CNN).
2. Bagaimana efektivitas penggunaan metode *Convolutional Neural Network* dalam mendeteksi serangan *malware* android tersebut.
3. Bagaimana perbandingan hasil deteksi dibandingkan dengan metode *machine learning* maupun *deep learning* lainnya

1.2.2. Batasan Masalah

Adapun batasan masalah dari penyusunan tugas akhir ini, yaitu:

1. *Dataset* yang digunakan yaitu *"android permission dataset*", dimana *dataset* ini terdiri dari sample *malware* android dari tahun 2018.
2. Penelitian ini hanya menggunakan satu metode yaitu *Convolutional Neural Network* (CNN).
3. Tidak dilakukannya uji sample secara dinamis

1.3. Tujuan

Adapun tujuan dari penyusunan tugas akhir, yaitu:

1. Menerapkan *Convolutional Neural Network* (CNN) pada *dataset "android permission dataset"* untuk mendeteksi serangan *malware* android berdasarkan ekstraksi fitur *permission*
2. Membandingkan hasil deteksi terbaik dari metode *Convolutional Neural Network* (CNN) dengan metode *machine learning* maupun *deep learning* lainnya

1.4. Manfaat

Adapun manfaat dari penyusunan tugas akhir, yaitu:

1. Dapat mengetahui cara menggunakan *Convolutional Neural Network* (CNN) dalam mendeteksi *malware* pada sistem operasi android
2. Dapat menunjukkan hasil deteksi dari metode *Convolutional Neural Network (CNN)* dalam mendeteksi *malware* pada sistem operasi android
3. Dapat mengetahui cara kerja *malware* pada android khususnya yang memanfaatkan fitur *permission*
4. Mampu mendapatkan hasil deteksi terbaik dari metode *Convolutional Neural Network* (CNN)

1.5. Metodologi Penelitian

Pada Tugas Akhir ini, metodologi yang digunakan adalah sebagai berikut:

1.5.1. Metode Studi Pustaka dan Literatur

Pada metode ini, penulis melakukan pencarian dan pengumpulan referensi berupa literatur yang terdapat pada artikel, jurnal dan internet yang berkaitan dengan Tugas Akhir yang sedang dikerjakan.

1.5.2. Metode Konsultasi

Dalam metode ini penulis melakukan konsultasi secara langsung dan atau tidak langsung kepada semua pihak narasumber yang memiliki pengetahuan serta wawasan yang baik dalam mengatasi permasalahan yang ditemui pada penulisan tugas akhir dengan judul Deteksi *Malware* Android Menggunakan Metode *Convolutional Neural Network* (CNN).

1.5.3. Metode Pembuatan Model

Pada metode ini dilakukan untuk membuat suatu perancangan pemodelan dengan menggunakan berbagai perangkat lunak dan simulasi untuk memudahkan proses pembuatan model.

1.5.4. Metode Pengujian dan Validasi

Metode ini dilakukan pengujian terhadap sistem yang telah dibuat karena perlu dilakukan untuk melihat batasan-batasan kinerja sistem tersebut dapat menghasilkan nilai akurasi yang baik atau sebaliknya.

1.5.5. Metode Analisis, Kesimpulan dan Saran

Hasil dari pengujian pada metode *Convolutional Neural Network* (CNN) untuk mendeteksi *malware* pada android akan dianalisis mengenai proses pendeteksian, kelebihan serta kekurangannya, sehingga menghasilkan suatu kesimpulan dan saran yang diharapkan dapat digunakan sebagai referensi yang baik untuk penelitian selanjutnya.

1.6. Sistematika Penulisan

Untuk mempermudah penyusunan Tugas Akhir ini dan memperjelas isi dari setiap bab, penulisan dilakukan dengan sistematika sebagai berikut :

BAB I PENDAHULUAN

Pada bab pertama membahas tentang Latar Belakang Masalah, Tujuan dan Manfaat, Perumusan dan Batasan Masalah, Metode Penelitian, serta Sistematika Penulisan dari penelitian yang dilakukan sebagai dasar penelitian ini.

BAB II TINJAUAN PUSTAKA

Bab kedua menjelaskan Dasar Teori, Konsep dan Prinsip Dasar yang diperlukan untuk memecahkan masalah dalam penelitian ini.

BAB III METODOLOGI

Bab ketiga ialah metodologi yang diterapkan akan dibahas secara rinci tentang teknik, metode, dan alur proses yang dilakukan dalam penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab keempat ialah hasil pengujian dan analisis yang telah diperoleh dari penelitian ini dan pembahasan terhadap hasil yang telah dicapai meliputi kelebihan dan kekurangan dari penelitian.

BAB V KESIMPULAN DAN SARAN

Pada bab kelima ialah kesimpulan yang bersumber dari hasil penelitian yang telah dilakukan dan juga saran untuk penelitian selanjutnya khususnya tentang Tugas Akhir yang dikerjakan..

DAFTAR PUSTAKA

[1] L. Shu, S. Dong, H. Su, and J. Huang, "Android *Malware* Detection Methods Based on *Convolutional Neural Network*: A Survey," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 7, no. 5, pp. 1330–1350, 2023, doi: 10.1109/TETCI.2023.3281833.

[2] "Number of Mobile Phone UsersWorldwide from 2016 to 2023 (In Billions)." Accessed: Nov. 25, 2023. [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[3] StatCounter, "Mobile Operating System Market Share Worldwide," gs.StatCounter.com. Accessed: Nov. 21, 2023. [Online]. Available: https://gs.statcounter.com/os-market-share/*mobile*/worldwide/#quarterly-202303-202303-bar

[4] S. Hojjatinia, S. Hamzenejadi, and H. Mohseni, "Android *botnet* detection using *convolutional neural networks*," *2020 28th Iran. Conf. Electr. Eng. ICEE 2020*, no. Ml, pp. 1–6, 2020, doi: 10.1109/ICEE50131.2020.9260674.

[5] O. Aslan and R. Samet, "A Comprehensive Review on *Malware* Detection Approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[6] securelist by Kaspersky, "IT threat evolution in Q2 2023. Mobile statistics." [Online]. Available: https://securelist.com/it-threat-evolution-q2-2023-*mobile*-statistics/110427/

[7] J. Senanayake, H. Kalutarage, M. O. Al-Kadri, A. Petrovski, and L. Piras, "Android Source Code Vulnerability Detection: A Systematic Literature Review," *ACM Comput. Surv.*, vol. 55, no. 9, 2023, doi: 10.1145/3556974.

[8] M. Alazab, M. Alazab, A. Shalaginov, A. Mesleh, and A. Awajan, "Intelligent *mobile malware* detection using *permission* requests and API calls," *Futur. Gener. Comput. Syst.*, vol. 107, pp. 509–521, 2020, doi: 10.1016/j.future.2020.02.002.

[9] C. Zhao, C. Wang, and W. Zheng, "Android *Malware* Detection Based on Sensitive Permissions and APIs," *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, vol. 284, no. 15, pp. 105–113, 2019, doi: 10.1007/978-3-030-21373-2_10.

[10] T. Peng *et al.*, "A Lightweight Multi-Source Fast Android *Malware* Detection Model," *Appl. Sci.*, vol. 12, no. 11, 2022, doi: 10.3390/app12115394.

[11] A. K. Singh, C. D. Jaidhar, and M. A. A. Kumara, "Experimental analysis of Android *malware* detection based on combinations of *permission*s and API-calls," *J. Comput. Virol. Hacking Tech.*, vol. 15, no. 3, pp. 209–218,

2019, doi: 10.1007/s11416-019-00332-z.

[12] J. Senanayake, H. Kalutarage, and M. O. Al-Kadri, “Android *mobile malware* detection using *machine learning*: A systematic review,” *Electron.*, vol. 10, no. 13, pp. 1–34, 2021, doi: 10.3390/electronics10131606.

[13] S. Millar, N. McLaughlin, J. M. del Rincon, and P. Miller, “Android *Malware* Detection Using *Deep Learning*,” *Artif. Intell. Cybersecurity Theory Appl.*, no. Icoei, pp. 209–246, 2022, doi: 10.1007/978-3-031-15030-2_10.

[14] “Android 10.” Accessed: Nov. 25, 2023. [Online]. Available: https://developer.android.google.cn/about/versions/10

[15] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, and H. Liu, “A Review of Android *Malware* Detection Approaches Based on *Machine Learning*,” *IEEE Access*, vol. 8, pp. 124579–124607, 2020, doi: 10.1109/ACCESS.2020.3006143.

[16] “Android Certificate.” Accessed: Nov. 25, 2023. [Online]. Available: https://developer.android.com/studio/publish/app-signing

[17] W. Y. Lee, J. Saxe, and R. Harang, “SeqDroid: Obfuscated android *malware* detection using stacked convolutional and recurrent neural networks,” *Adv. Sci. Technol. Secur. Appl.*, pp. 197–210, 2019, doi: 10.1007/978-3-030-13057-2_9.

[18] O. N. Elayan and A. M. Mustafa, “Android *malware* detection using *deep learning*,” *Procedia Comput. Sci.*, vol. 184, no. 2019, pp. 847–852, 2021, doi: 10.1016/j.procs.2021.03.106.

[19] “Android Permission.” [Online]. Available: https://developer.android.com/guide/topics/*permission*s/overview

[20] A. Mahindru and A. L. Sangal, *MLDroid—framework for Android malware detection using machine learning techniques*, vol. 33, no. 10. Springer London, 2021. doi: 10.1007/s00521-020-05309-4.

[21] Y. Pan, X. Ge, C. Fang, and Y. Fan, “A Systematic Literature Review of Android *Malware* Detection Using Static Analysis,” *IEEE Access*, vol. 8, pp. 116363–116379, 2020, doi: 10.1109/ACCESS.2020.3002842.

[22] A. Mahindru and A. L. Sangal, *SOMDROID: android malware detection by artificial neural network trained using unsupervised learning*, vol. 15, no. 1. Springer Berlin Heidelberg, 2022. doi: 10.1007/s12065-020-00518-1.

[23] P. Calciati, K. Kuznetsov, A. Gorla, and A. Zeller, “Automatically Granted Permissions in Android apps: An Empirical Study on their Prevalence and on the Potential Threats for Privacy,” *Proc. - 2020 IEEE/ACM 17th Int. Conf. Min. Softw. Repos. MSR 2020*, pp. 114–124, 2020, doi: 10.1145/3379597.3387469.

[24] S. Garg and N. Baliyan, “Comparative analysis of Android and iOS from

security viewpoint," *Comput. Sci. Rev.*, vol. 40, p. 100372, 2021, doi: 10.1016/j.cosrev.2021.100372.

[25] S. Y. Yerima and M. K. Alzaylaee, "Mobile *Botnet* Detection: A *Deep Learning* Approach Using *Convolutional Neural Network*s," *2020 Int. Conf. Cyber Situational Awareness, Data Anal. Assessment, Cyber SA 2020*, 2020, doi: 10.1109/CyberSA49311.2020.9139664.

[26] S. Y. Yerima, M. K. Alzaylaee, A. Shajan, and P. Vinod, "*Deep learning* techniques for android *botnet* detection," *Electron.*, vol. 10, no. 4, pp. 1–17, 2021, doi: 10.3390/electronics10040519.

[27] P. Zegzhda, D. Zegzhda, E. Pavlenko, and G. Ignatev, "Applying *deep learning* techniques for Android *malware* detection," *ACM Int. Conf. Proceeding Ser.*, 2018, doi: 10.1145/3264437.3264476.

[28] E. M. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, "MalDozer: Automatic framework for android *malware* detection using *deep learning*," *DFRWS 2018 EU - Proc. 5th Annu. DFRWS Eur.*, vol. 24, pp. S48–S59, 2018, doi: 10.1016/j.diin.2018.01.007.

[29] A. Darwaish and F. Nait-Abdesselam, "RGB-based Android *Malware* Detection and Classification Using *Convolutional Neural Network*," *2020 IEEE Glob. Commun. Conf. GLOBECOM 2020 - Proc.*, vol. 2020-Janua, pp. 8–13, 2020, doi: 10.1109/GLOBECOM42002.2020.9348206.

[30] H. Chen, Z. Li, Q. Jiang, A. Rasool, and L. Chen, "A hierarchical approach for android *malware* detection using authorization-sensitive features," *Electron.*, vol. 10, no. 4, pp. 1–24, 2021, doi: 10.3390/electronics10040432.

[31] A. Sharma, P. Malacaria, and M. H. R. Khouzani, "*Malware* detection using 1-dimensional *convolutional neural networks*," *Proc. - 4th IEEE Eur. Symp. Secur. Priv. Work. EUROS PW 2019*, pp. 247–256, 2019, doi: 10.1109/EuroSPW.2019.00034.

[32] Y. Zhang, Y. Yang, and X. Wang, "A novel android *malware* detection approach based on *convolutional neural network*," *ACM Int. Conf. Proceeding Ser.*, pp. 144–149, 2018, doi: 10.1145/3199478.3199492.

[33] D. Zhu, T. Xi, P. Jing, D. Wu, Q. Xia, and Y. Zhang, "A transparent and multimodal *malware* detection method for android apps," *MSWiM 2019 - Proc. 22nd Int. ACM Conf. Model. Anal. Simul. Wirel. Mob. Syst.*, pp. 51–60, 2019, doi: 10.1145/3345768.3355915.

[34] K. Sugunan, T. Gireesh Kumar, and K. A. Dhanya, *Static and dynamic analysis for android malware detection*, vol. 645. Springer Singapore, 2018. doi: 10.1007/978-981-10-7200-0_13.

[35] Z. Wang, Q. Liu, and Y. Chi, "Review of Android *Malware* Detection Based on *Deep Learning*," vol. 8, 2020, doi: 10.1109/ACCESS.2020.3028370.

[36] M. Choudhary and B. Kishore, "HAAMD: Hybrid Analysis for Android

Malware Detection," *2018 Int. Conf. Comput. Commun. Informatics, ICCCI 2018*, pp. 1–4, 2018, doi: 10.1109/ICCCI.2018.8441295.

[37] "Apktool A tool for reverse engineering Android apk files."

[38] J. Qiu, J. Zhang, W. Luo, L. Pan, S. Nepal, and Y. Xiang, "A Survey of Android *Malware* Detection with Deep Neural Models," *ACM Comput. Surv.*, vol. 53, no. 6, 2021, doi: 10.1145/3417978.

[39] P. Calciati, K. Kuznetsov, A. Gorla, and A. Zeller, "Automatically Granted Permissions in Android apps," pp. 114–124, 2020, doi: 10.1145/3379597.3387469.

[40] M. Alazab, M. Alazab, A. Shalaginov, and A. Mesleh, "Intelligent *mobile malware* detection using *permission* requests and API calls," *Futur. Gener. Comput. Syst.*, vol. 107, pp. 509–521, 2020, doi: 10.1016/j.future.2020.02.002.

[41] K. Xu, Y. Li, R. H. Deng, and K. Chen, "DeepRefiner: Multi-layer Android *Malware* Detection System Applying Deep Neural Networks," *Proc. - 3rd IEEE Eur. Symp. Secur. Privacy, EURO S P 2018*, pp. 473–487, 2018, doi: 10.1109/EuroSP.2018.00040.

[42] "Android *Malware* Detection Based on *Convolutional Neural Network*s," 2019.

[43] A. Mahindru, "Android *permission dataset*," 2018, [Online]. Available: 10.17632/8y543xvnsv.1

[44] M. E. Khoda, J. Kamruzzaman, I. Gondal, T. Imam, and A. Rahman, "*Malware* detection in edge devices with fuzzy *oversampling* and dynamic class weighting," *Appl. Soft Comput.*, vol. 112, p. 107783, 2021, doi: 10.1016/j.asoc.2021.107783.

[45] H. Shamsudin, U. K. Yusof, A. Jayalakshmi, and M. N. Akmal Khalid, "Combining *oversampling* and undersampling techniques for *imbalance*d classification: A comparative study using credit card fraudulent transaction *dataset*," *IEEE Int. Conf. Control Autom. ICCA*, vol. 2020-Octob, pp. 803–808, 2020, doi: 10.1109/ICCA51439.2020.9264517.

[46] U. S. Jannat, S. M. Hasnayeen, M. K. Bashar Shuhan, and M. S. Ferdous, "Analysis and Detection of *Malware* in Android Applications Using *Machine Learning*," *2nd Int. Conf. Electr. Comput. Commun. Eng. ECCE 2019*, pp. 1–7, 2019, doi: 10.1109/ECACE.2019.8679493.

[47] J. Lopes, C. Serrão, L. Nunes, A. Almeida, and J. Oliveira, "Overview of *machine learning* methods for Android *malware* identification," *7th Int. Symp. Digit. Forensics Secur. ISDFS 2019*, pp. 1–6, 2019, doi: 10.1109/ISDFS.2019.8757523.

[48] S. Arshad, M. A. Shah, A. Wahid, A. Mehmood, H. Song, and H. Yu, "SAMADroid: A Novel 3-Level Hybrid *Malware* Detection Model for Android Operating System," *IEEE Access*, vol. 6, no. c, pp. 4321–4339,

2018, doi: 10.1109/ACCESS.2018.2792941.

[49] Y. Zhao, G. Xu, and Y. Zhang, *Quality, Reliability, Security and Robustness in Heterogeneous Systems*, vol. 234. Springer International Publishing, 2018. doi: 10.1007/978-3-319-78078-8.

[50] E. Suherman, D. Hindarto, A. Makmur, and H. Santoso, "Comparison of *Convolutional Neural Network* and Artificial Neural Network for Rice Detection," *Sinkron*, vol. 8, no. 1, pp. 247–255, 2023, doi: 10.33395/sinkron.v8i1.11944.