IMPLEMENTASI SCANNING APLIKASI ANDROID MENGGUNAKAN APKID

[1]Gilang Ramadhan, [2]Anang Sularsa, [3]Setia Juli Irzal Ismail.

[1] Computer Engginering, Telkom University, Bandung, Indonesia.

[1] gilangr@student.telkomuniversity.ac.id, [2]anang@tass.telkomuniversity.ac.id, [3]jul@tass.telkomuniversity.ac.id.

Abstrak: *Saat ini android telah menjadi salah satu sistem operasi yang popular. Perkembangannya dari tahun ke tahun semakin meningkat. Baik dari segi fitur maupun aplikasi pendukungnya. Begitu banyak ragam aplikasi dengan fungsi yang dapat membantu pekerjaan sehari-hari menjadi mudah. Perkembangan ini di ikuti dengan semakin banyaknya malware yang dibuat untuk android. Sebanyak 632.451 upaya serangan malware pada android di klaim berhasil diblokir oleh Kaspersky dari bulan januari hingga September 2019. Jumlah ini menjadikan Indonesia sebagai negara dengan jumlah ancaman android paling banyak terdeteksi Begitu banyak ancaman malware android saat ini, dan diperburuk dengan kebiasaan sebagian orang Indonesia yang tidak menginstalasi aplikasi dari google market resmi karena alasan berbayar sehingga membuat malware semakin menyebar. Android Application Identifier (APKiD) adalah salah satu software yang dapat digunakan untuk membantu proses analisis malware pada file yang berekstensi Apk, fungsinya untuk mengetahui apabila file Apk berpotensi mengandung malware. Android Application Identifier (APKiD) dapat memberikan informasi tentang compilers, packers, obfuscators dan hal aneh lainnya yang menjadi potensi adanya malware.*

Kata Kunci ***: Android Application Identifier (APKiD)*****, Aplikasi** ***android***

1. Pendahuluan

1.1 Latar Belakang

Saat ini *android* telah menjadi salah satu sistem operasi yang popular. Perkembangannya dari tahun ke tahun semakin meningkat. Baik dari segi fitur maupun aplikasi pendukungnya. Begitu banyak ragam aplikasi dengan fungsi yang dapat membantu pekerjaan sehari-hari menjadi mudah. Perkembangan ini diikuti dengan munculnya berbagai malware baru yang dibuat untuk menyerang *android.* Sebanyak 632.451 upaya serangan *malware* pada *android* di klaim berhasil diblokir oleh Kaspersky dari bulan januari hingga September 2019. Jumlah ini menjadikan Indonesia sebagai negara dengan jumlah ancaman *android* paling banyak terdeteksi di Asia Tenggara. Baru-baru ini juga telah ditemukan *malware* jenis *trojan* yang bernama *Joker. Malware* ini ditemukan di balik puluhan aplikasi *android* di *Google Play Store* dan mendorong untuk berlangganan layanan premium secara diam-diam tanpa disadari pengguna. *Malware* ini berperan sebagai komponen latar belakang dan secara diam-diam mengklik iklan dalam aplikasi dan melakukan proses pendaftaran ketika dalam situs. Kemudian mengakses pesan SMS pengguna, menyalin kode otorasi yang telah mereka kirim untuk memverifikasi pembayaran berlangganan. [1] Begitu banyak ancaman *malware android* saat ini, dan diperburuk dengan kebiasaan orang Indonesia yang tidak menginstalasi aplikasi dari google market resmi karena alasan berbayar sehingga membuat *malware* semakin menyebar.

Android Application Identifier *(APKiD)* adalah salah satu software yang dapat digunakan untuk membantu proses analisis *malware* pada file yang berekstensi Apk, fungsinya untuk mengetahui apabila file Apk berpotensi mengandung *malware.* Android Application Identifier *(APKiD)* dapat memberikan informasi tentang *compilers, packers, obfuscators* dan hal aneh lainnya. *Android* Application Identifier *(APKiD)* saat ini hanya dapat mengidentifikasi *compilers, packers, obfuscators* yang di pakai file Apk. Fungsi ini masih kurang efektif untuk mengetahui ada tidaknya potensi *malware* pada file Apk. Untuk mengembangkan proyek akhir ini *Android* Application Identifier *(APKiD)* di tambahkan *Tools VirusTotal dan Behavioral Analysis* untuk membuktikan file Apk sebagai *malware.*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disampaikan maka dapat dirumuskan masalah proyek akhir ini adalah cara menggunakan Android Application Identifier *(APKiD)* untuk mengidentifikasi *malware* yang ada di aplikasi *android.*

1.3 Tujuan

Tujuan dari proyek akhir ini adalah menggunakan Android Application Identifier *(APKiD) untuk* mengidentifikasi *malware* pada file yang berekstensi Apk.

1.4 Batasan Masalah

Batasan masalah untuk proyek akhir ini adalah :

1. Menggunakan *Android Application Identifier (APKiD)* sebagai *software* untuk mengidentifikasi aplikasi *android*.
2. Menggunakan file apk sebagai sampel pengujian.
3. Menggunakan *Virtual Machine* sebagai *software* pendukung untuk pengujian.

1.5 Definisi Operasional

Adapun definisi operasional pada proyek akhir ini adalah sebagai berikut:

1. **Keamanan Aplikasi.** keadaan aplikasi yang berjalan pada suatu sistem terbebas dari bahaya seperti tindak kejahatan, segala bentuk kecelakaan dan lain-lain.
2. ***Android***. *android* merupakan sistem operasi berbasis *Linux* yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. *Android* menyediakan platform terbuka untuk para pengembang agar dapat menciptakan aplikasi mereka sendiri.
3. ***APKiD.*** *Android Application Identifier* (*APKiD*) adalah salah satu *software* yang dapat digunakan untuk membantu proses analisis *malware* pada file yang berekstensi Apk. Dengan memberikan informasi tentang *compilers*, *packers*, *obfucators* dan hal aneh lain-lain.

2. Tinjauan Pustaka

2.1 Komparasi Dari Penelitian Sebelumnya

1. *PDF Forensic Analysis System using YARA*

   a) Persamaan: menggunakan *YARA rules* sebagai cara untuk mendeteksi *malware* atau file lainnya dengan membuat *rules* untuk mencari karakteristik tertentu.

   b) Perbedaan: digunakan untuk mendeteksi *malware* pada file berekstensi PDF.

2. *Implementation of Malware Detection Service on Malware*

   a) Persamaan: digunakan untuk mendeteksi *malware* pada file yang berekstensi Apk.

   b) Perbedaan: dapat diakses melalui *web browser* dan perangkat *android.*

3. Analisis Forensik *Malware* Pada *Platform Android*

   a) Persamaan: digunakan untuk menganalisa aplikasi *android* agar tidak masuk ke perangkat *android*.

   b) Perbedaan: menggunakan tool Dex2jar untuk menganalisis *malware* pada aplikasi *android.*

2.2 Android Application Identifier (APKiD)

Android Application Identifier (APKiD) adalah salah satu software yang dapat digunakan untuk membantu proses analisis *malware* pada file yang berekstensi Apk, fungsinya untuk mengetahui apabila file Apk berpotensi mengandung *malware. Android Application Identifier (APKiD)* dapat memberikan informasi tentang *compilers*, *packers*, *obfuscators* dan hal aneh lainnya. *Android Application Identifier (APKiD)* saat ini hanya dapat mengidentifikasi *compilers, packers, obfuscators* yang di pakai file Apk. Fungsi ini masih kurang efektif untuk mengetahui ada tidaknya potensi *malware* pada file Apk. Untuk mengembangkan proyek akhir ini *Android Application Identifier (APKiD)* di tambahkan *Tools VirusTotal dan Behavioral Analysis* untuk membuktikan file Apk sebagai *malware.*

2.3 Android

Android adalah sistem operasi yang dirancang untuk perangkat layar sentuh seperti *smartphone* dan tablet yang sekarang banyak sekali penggunanya diseluruh dunia. Dengan jumlah pengguna yang begitu banyak menjadikan sistem operasi *android* sebagai sasaran bagi para

pembuat atau penulis *malware* untuk menyebarkan *software* jahat yang dapat merugikan pengguna *android*.

2.4 Malware

Malware merupakan sebuah *software* jahat yang sengaja dibuat dengan berbagai tujuan tertentu oleh pembuatnya yang dapat mengakibatkan kerugian pada pengguna yang menjadi sasaran dari pembuat *malware*. Kata *malware* merupakan istilah umum yang digunakan untuk *software* atau *program* yang dibuat dengan tujuan untuk menyusup maupun merusak sebuah sistem secara diam-diam.

2.4.1 Contoh *Malware*

1. Virus
   Virus merupakan sebuah jenis *malware* yang menyerang *file exe* yang akan menyerang dan menduplikasi diri ketika *file exe* yang terinfeksi tersebut di jalankan. Virus akan menyebar dengan cara menyisipkan program jahatnya pada program atau dokumen yang ada dalam komputer.
2. Worm
   Worm merupakan sebuah program komputer yang dapat menduplikasi dirinya sendiri dalam sistem komputer. W*orm* dapat menduplikasi dirinya sendiri dengan memanfaatkan jaringan (LAN/WAN/Internet) tanpa perlu campur tangan dari pengguna itu sendiri. *Worm* memanfaatkan celah keamanaan yang memang terbuka yang dikenal dengan sebutan *vulnerability*.
3. Spyware
   Spyware merupakan program yang dibuat dengan tujuan untuk memata-matai dan untuk mengetahui kebiasaan pengguna komputer dan mengirimkan informasi tersebut ke pihak lain. Biasanya *spyware* digunakan oleh pihak pemasang iklan.
4. Adware
   Adware merupakan program jahat berbentuk iklan yang dimasukan tersembunyi oleh pembuat program, biasanya pada *adware* ini bersifat *freeware* untuk tujuan promosi atau iklan.
5. Trojan
   Trojan merupakan program jahat yang secara diam-diam dimasukkan ke sistem kita, kemudian menyediakan program lain seperti *virus*, *adware*, *spyware*, *keylogger* dan *malware* lainnya untuk masuk pada sistem, merusak sistem, memungkinkan orang lain mengontrol komputer dan mencuri informasi seperti *password* atau nomor kartu kredit.
6. Keylogger
   Keylogger merupakan sebuah program yang dibuat untuk memantau penekanan tombol pada *keyboard*, sehingga orang lain bisa mengetahui informasi apapun yang korban ketik bahkan *password*.
7. Rootkit
   Rootkit merupakan program yang dirancang untuk menyusup kedalam sistem komputer dengan menyamar sebagai bagian dari sistem. Kemudian mengontrol dan memantau kerja sistem yang disusupinya. *Rootkit* bisa mencuri data pada lalu lintas di jaringan, melakukan *keylogging*, mencuri data akun bank yang tersimpan pada browser dan lain-lain.

2.5 Kali Linux

Kali Linux merupakan sistem operasi yang memiliki sistem keamanan tinggi yang kebal terhadap serangan virus dan *Kali Linux* memiliki lebih dari 300 *tools* di dalamnya dengan fungsi masing-masing. *Kali Linux* juga bersifat *Live CD* dan Instalasi manual. *Kali Linux* memiliki tampilan yang terbilang sederhana, tidak mencolok dan penggunaanya juga tergolong cukup mudah, sehingga sangat baik digunakan untuk para pemula dalam melakukan analisis atau deteksi *malware*. Karena hal tersebut *APKiD* sangat cocok dijalankan di sistem operasi *Kali Linux*.

2.6 Bahasa Pemrograman Python

Python merupakan bahasa pemrograman tinggi yang secara langsung dapat melakukan eksekusi sejumlah perintah multi guna. Python merupakan bahasa pemrograman yang mudah untuk dipahami karena *python* lebih menekankan pada keterbacaan sintaks. Oleh karena itu python menjadi bahasa pemrograman yang banyak digemari oleh para programer. *APKiD* menggunakan bahasa pemrograman *python* dalam konfigurasi dan penggunaannya.

2.7 VirtualBox

Virtualbox merupakan *virtual machine* gratis milik *Oracle* yang biasanya digunakan untuk mencoba berbagai sistem operasi di dalam sistem operasi utama. *APKiD* menggunakan *Virtualbox* agar dapat melakukan pengujian dengan aman.

2.8 Teknik Pengelabuan

Teknik pengelabuan atau obfuscation secara bahasa berasal dari istilah latin obfuscare (=to darken / menyamarkan). Dalam ilmu komunikasi obfuscation adalah cara untuk membuat sebuah pesan menjadi membingungkan dan susah untuk

dipahami. Teknik ini akan merubah sebuah program menjadi versi baru, tapi tetap berjalan dengan fungsi yang sama dengan code asli. Teknik ini digunakan oleh para pembuat malware agar malwarenya tidak dapat dideteksi oleh antivirus.

Terdapat beberapa cara untuk melakukan *obfuscation* pada *malware* diantaranya adalah:

1. Enkripsi

   Enkripsi adalah cara mengubah data menjadi data tidak dapat dibaca tanpa potongan kode (baca: key)-nya. Dengan begitu pesan tersebut hanya bisa dibaca oleh orang yang memiliki pesan yang integritasnya *valid.*

2. *Oligomorphic*

   Oligomorphic merupakan pengembangan dari teknik enkripsi. Pada *malware* yang dienkripsi terdapat *decryptor* yang bertugas untuk melakukan deskripsi *malware*. *Decryptor* menjadi titik lemah dari *malware* karena *antivirus* akan dengan mudah mendeteksi *decryptor*. Menyadari kelemahan ini penulis *malware* mengembangkan teknik *Oligomorphic.* Teknik ini dapat melakukan mutasi pada *decryptor* **s**ehingga *decryptor* memiliki berbagai variasi. Dengan melakukan mutasi *decryptor malware* akan *menggenerate* sampai ratusan variasi *decryptor.*

3. *Anti_vm*

   Anti_vm merupakan teknik anti *virtual machine* yang sering digunakan oleh penulis *malware* untuk menggagalkan upaya identifikasi atau analisis. Dengan teknik ini *malware* akan mencoba untuk mendeteksi apakah *malware* sedang dijalankan dalam *virtual machine* atau tidak. Jika *malware* mendeteksi *virtual machine, malware* akan bertindak berbeda atau tidak berjalan sama sekali.

4. *Compilers*

   Compilers adalah program khusus yang memproses pernyataan yang ditulis dalam bahasa pemrograman dan merubahnya menjadi bahasa mesin "*code"* yang digunakan olah komputer.

2.9 Docker

Docker platform modern untuk inovasi High-Velocity. Satu-satunya platform kontainer idependen yang memungkinkan untuk membangun, mengemas, dan menjalankan aplikasi dimanapun dan menyederhanakan kofigurasi.

3. Analisis Dan Perancangan

3.1 Analisis

Pada bab ini menjelaskan mengenai proses analisis terkait dengan cara kerja sistem yang sedang berjalan, kemudian pada bab ini juga akan dibahas penjelasan mengenai gambaran umum sistem, dan diagram blok.

3.1.1 Analisis Kebutuhan Fungsional dan Non Fungsional

A. Kebutuhan Fungsional

Berikut ini adalah kebutuhan sistem yang diperlukan untuk menyelesaikan Proyek Akhir ini adalah sebagai berikut.

1. *Android Application Identifier (APKiD)* dibangun untuk dapat menganalisis *malware* pada aplikasi *android.*
2. *Python* adalah Bahasa pemrograman yang digunakan untuk konfigurasi *Android Application Identifier (APKiD)*.
3. Mesin virtual digunakan untuk analisis.
4. *Android* yang akan dijadikan sampel untuk uji coba didapatkan dari *Web*.

B. Kebutuhan Non Fungsional

Kebutuhan non fungsional pada sistem yang akan dibangun terdiri dari dua bagian yaitu *hardware* dan *software*, adapun rincian dari kedua bagian tersebut adalah sebagai berikut.

1. *Hardware*

Tabel 3.1 merupakan kebutuhan *hardware* atau perangkat keras yang dibutuhkan untuk membangun sistem adalah sebagai berikut.

Tabel Error! No text of specified style in document.**.1 Daftar Perangkat Keras *User***

No	*Hardware User*	Spesifikasi	Keterangan
1	*Laptop User*	*RAM* 8 GB, *Storage* 500 GB	*Hardware* yang digunakan untuk menjalankan sistem operasi Kali Linux 2019.4

2. *Software*

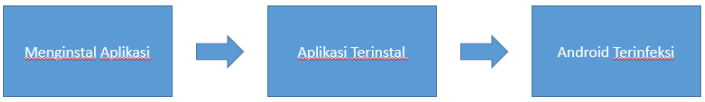
Tabel 3.2 merupakan kebutuhan *software* atau perangkat lunak yang dibutuhkan untuk membangun sistem adalah sebagai berikut.

Tabel Error! No text of specified style in document.**.2 Daftar Perangkat Lunak *User***

No	Software User	Fungsi
1	*Virtual Box*	Mesin yang digunakan untuk menjalankan analisis.
2	Kali Linux 2019.4	Sistem operasi yang digunakan untuk menjalankan *APKiD*
3	Virtualenv	Untuk membantu instalasi modul yang di butuhkan APKiD
4	*Python*	Bahasa Pemrograman *Python* digunakan untuk konfigurasi dalam penggunaan *Android Application Identifier (APKiD)*.
5	*Yara-Python*	Library *Python* digunakan untuk mencakup semua fitur yara.
6	*File* Apk	*File* aplikasi *android* yang didapat dari *Web* yang kemudian dijadikan sampel untuk menganalis *malware*.
7	*Android Application Identifier (APKiD)*	*Software* untuk membantu melakukan proses analisis malware pada file Apk

3.1.2 Diagram Blok

Berikut adalah diagram blok sistem saat ini ketika melakukan instalasi aplikasi *android* adalah seperti berikut.



Gambar Error! No text of specified style in document.**.1 Diagram Blok Sistem Saat Ini**

Gambar 3.1 merupakan tampilan dari diagram blok sistem saat ini, yang merupakan cara kerja sistem saat ini adalah umumnya orang mengunduh file Apk dari berbagai sumber padahal begitu banyak file Apk ini yang sudah disisipi malware.

3.1.3 Gambaran Sistem Saat Ini

Adapun gambaran sistem saat ini dalam pembahasan proyek akhir ini, sebagai berikut:

Gambar Error! No text of specified style in document.**.2 Sistem Saat Ini**

Maksud dari Gambar 3.2 Merupakan gambaran sistem saat ini, untuk melakukan penginstalan sebuah aplikasi Umumnya orang mengunduh file Apk dari berbagai sumber, padahal begitu banyak file Apk ini yang telah disisipi *malware*. Ada juga kebiasaan orang yang tidak menginstalasi aplikasi dari google market resmi karena alasan aplikasi berbayar yang dapat membuat *malware* semakin cepat menyebar.

3.2 Perancangan

Perancangan menjelaskan mengenai proses cara kerja sistem yang akan dibuat, Kemudian juga akan dibahas penjelasan mengenai gambaran umum sistem usulan, diagram blok, cara kerja, dan analisis kebutuhan fungsional dan *non* fungsional.

3.2.1 Diagram Blok

Gambar Error! No text of specified style in document.**.3 Diagram Blok Sistem Usulan**

Keterangan dari diagram blok tersebut adalah sebagai berikut:

1. *Input*

Pada tahap ini dilakukan instalasi *Android Application Identifier* (*APKiD*) terlebih dahulu dengan cara mengunduh modul *Android Application Identifier* (*APKiD*) dari https://github.com/rednaga/APKiD.

2. *Process*

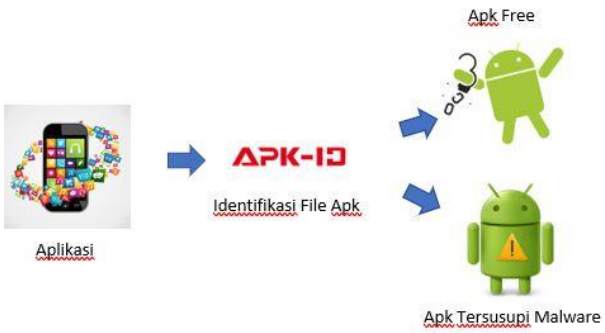
Pada tahap ini akan dilakukan konfigurasi terlebih dahulu pada *Android Application Identifier (APKiD)* dengan menginstal beberapa program yang diperlukan seperti *python, yara-python,* dan *Virtualenv* yang akan dijelaskan lebih lanjut pada bagian implementasi.

3. Output

Pada tahap ini akan dihasilkan *output* berupa *Android Application Identifier* siap digunakan untuk mengidentifikasi *malware* pada file yang berekstensi Apk.

3.2.2 Gambaran Sistem Usulan

Pada tahap ini akan dihasilkan *output* berupa *Android Application Identifier* siap digunakan untuk menganalisis aplikasi *android.*



Gambar Error! No text of specified style in document.**.4 *Gambaran* Sistem Usulan**

Gambar 3.4 merupakan gambaran sistem usulan dalam Proyek Akhir ini, sistem ini menggunakan *Android Application Identifier* (APKiD) untuk membantu melakukan analisis *malware* pada file yang berekstensi Apk. *Android Application Identifier* (APKiD) akan memberikan informasi tentang bagaimana Apk dibuat dan mengidentifikasi teknik pengelabuan (*obfuscation)* seperti berbagai *compilers, packers, obfuscators* dan hal aneh lainnya yang menjadi indikasi adanya *malware.*

3.2.3 Diagram Blok

Gambar Error! No text of specified style in document.**.5 Diagram Blok Cara Kerja**

Keterangan dari diagram blok tersebut adalah sebagai berikut:

1. *Input*

Pada tahap ini digunakan perangkat keras yaitu laptop yang menggunakan *sistem operasi Kali Linux 2019.4*. Kemudian *Android Application Identifier (APKiD)* dijalankan untuk melakukan identifikasi teknik *obfuscation*. Dengan menggunakan *docker* sebagai wadah untuk menjalankan *Android Application Identifier (APKiD)* dan masukkan sampel *file Apk* yang didapat dari *Web.*

2. *Process*

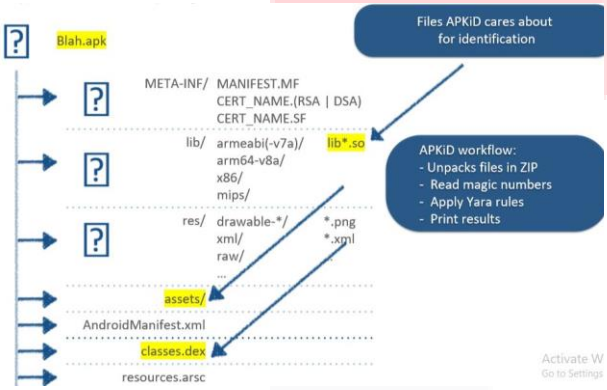
Pada tahap ini akan dilakukan proses identifikasi dengan masuk kedalam direktori sampel *file* aplikasi *android* yang sudah disediakan sebelumnya yang kemudian diidentifikasi dengan *Android Application Identifier (APKiD)*.

3. *Output*

Pada tahap ini akan dihasilkan *output* berupa informasi tentang bagaimana Apk dibuat dan berbagai *compilers*, *packers*, *obfuscators* dan hal aneh lainnya.

3.2.4 Cara Kerja

Berikut ini adalah cara kerja sistem yang akan dibuat dalam Proyek Akhir ini adalah sebagai berikut:



Gambar Error! No text of specified style in document.**.6 Cara Kerja APKiD**

1 *Android Application Identifier (APKiD)* mengidentifikasi file Apk, membaca *MANIFEST.MF, lib dan res*.

2 *Android Application Identifier (APKiD)* melakukan *Unpacks* pada file Apk.

3 *Android Application Identifier (APKiD)* melakukan deteksi aktifitas folder *Manifest, lib, assets* dan *classic.dex*.

4 *Android Application Identifier (APKiD)* membandingkan dengan signature yara.

5 *Android Application Identifier (APKiD)* mengeluarkan hasil identifikasi file Apk.

4. Pengujian

Pengujian 30 File Apk menggunakan APKiD

Tabel Error! No text of specified style in document.**.3 Hasil Pengujian 30 File Apk Menggunakan APKiD**

No	Nama File	Size File	Processing Time	Hasil Berpotensi Malware
1.	com.fdhgkjhrtjkjbx.model.apk	2.4 MB	2.91 Detik	Ya
2.	com.c101421042723.apk	282 KB	1.33 Detik	Tidak
3.	Ram Memory Booster 64GB_v1.0_apkpure.com.apk	3.5 MB	5.10 Detik	Ya
4.	org.benews.apk	1.13 MB	2.24 Detik	Tidak
5.	brother.apk	26.9 KB	1.17 Detik	Tidak
6.	mcpef.apk	55.1 KB	0.74 Detik	Tidak
7.	BadNews.A.apk	3.2 MB	1.85 Detik	Ya
8.	Bios.NativeMaliciousCode.apk	6.2 MB	2.60 Detik	Ya
9.	com.parental.control.v4.apk	921 KB	2.15 Detik	Tidak
10.	com.parental.control.v4-dexguarded.apk	814 KB	1.09 Detik	Ya
11.	com.tinker.gameone(Cowboy Adventure).apk	20.7 MB	3.45 Detik	Tidak
12.	com.tinker.jumperchess(Jump Chess).apk	10.1 MB	2.57 Detik	Tidak
13.	Omigo.apk	584 KB	1.07 Detik	Ya
14.	SmsThief.apk	172 KB	1.08 Detik	Ya
15.	SmsWorker.apk	1.79 MB	1.27 Detik	Ya
16.	FakeBank.B.apk	224 KB	1.26 Detik	Ya
17.	FakeCMCC.A.apk	212 KB	1.16 Detik	Ya

18.	GinMaster.Z.AdvancedObfuscation.apk	12.9 MB	3.09 Detik	Ya
19.	Agent.apk	17.7 KB	0.85 Detik	Tidak
20.	Blatantsms.apk	4.56 MB	2.60 Detik	Tidak
21.	FakeValidation.apk	12 MB	2.91 Detik	Ya
22.	Fobus.apk	121 KB	1.06 Detik	Tidak
23.	Opfake.apk	50.6 KB	0.95 Detik	Tidak
24.	Vietcon.apk	5.39 MB	2.45 Detik	Tidak
25.	JiFake.A.apk	1.25 MB	2.33 Detik	Ya
26.	NotCompatible.A.apk	22.3 KB	0.87 Detik	Tidak
27.	Obad.A.apk	82.3 KB	0.97 Detik	Ya
28.	Oldboot.A.apk	52.4 KB	1.10 Detik	Tidak
29.	Samsapo.A.apk	463 KB	0.97 Detik	Ya
30.	XTaoAd.A.apk	729 KB	0.88 Detik	Ya

Kesimpulan yang di dapat dari pengujian:

a. File apk yang terdeteksi menggunakan *compilers Dexlib* kemungkinan file apk telah dirusak, jika file telah dirusak atau di crack kemungkinan file adalah malware.

b. File apk yang terdeteksi menggunakan *packers* maka file berpotensi sebagai *malware*.

c. File apk yang terdeteksi menggunakan teknik obfuscation seperti anti_vm berpotensi sebagai malware.

5 Kesimpulan

Proyek akhir ini menyimpul APKiD mengidentifikasi 30 file apk, dari 30 file apk, 53% file apk terdeteksi menggunakan teknik obfuscation seperti *dexlib*, *packers*, dan *anti_vm* yang menjadi ciri bahwa file apk berpotensi sebagai malware.

Daftar Pustaka

[1] W. K. Pertiwi, "Awas Virus Joker, Segera Hapus 24 Aplikasi Ini dari Ponsel Android Anda," kompas.com, 10 September 2019. [Online]. Available: https://tekno.kompas.com/read/2019/09/10/09080007/awas-virus-joker-segera-hapus-24-aplikasi-ini-dari-ponsel-android-anda. [Accessed 31 Desember 2019].

[2] staf, "rednaga/APKiD," github.com, [Online]. Available: https://github.com/rednaga/APKiD. [Accessed 31 Agustus 2018].

[3] A. Kartono, A. Sularsa and S. J. I. Ismail, "MEMBANGUN SISTEM PENGUJIAN KEAMANAN APLIKASI ANDROID MENGGUNAKAN MOBSF," vol. 1, p. 146, 2019.

[4] J. Brigs, Python For Kids, San Francisco: william Pollock, 2013.

[5] belajarpython, "Pengertian Python," belajarpython.com, [Online]. Available: https://belajarpython.com/tutorial/apa-itu-python. [Accessed 10 September 2018].

[6] Julismail, "Malware Obfuscation," julismail.staff.telkomuniversity.ac.id, 23 Mei 2015. [Online]. Available: https://julismail.staff.telkomuniversity.ac.id/malware-obfuscation/. [Accessed 31 Desember 2019].

[7] Soeleman, "Apa itu Encoding, Obfuscation, Hashing dan Encryption," codepolitan, 15 Maret 2017. [Online]. Available: https://www.codepolitan.com/apa-itu-encoding-obfuscation-hashing-dan-encryption-58bfb7eee3215. [Accessed 31 Desember 2019].

[8] R. Novrianda, Y. N. Kunang and P. Shaksono, in *ANALISIS FORENSIK MALWARE PADA PLATFORM ANDROID*, Makassar, 2014.

[9] M. Habibi, S. J. Ismail and A. Sularsa, "IMPLEMENTATION OF MALWARE DETECTION SERVICE ON ANDROID," vol. 3, p. 1839, 2017.

[10] M. K. MASDUQI, in *ANALISIS INFEKSI MALWARE JENIS UAPUSH MENGGUNAKAN METODE STATIC DAN BEHAVIOR PADA ANDROID*, 2017.

[11] ashishb, "Android Malware," github.com, 24 Agustus 2019. [Online]. Available: https://github.com/ashishb/android-malware. [Accessed 2 Januari 2020].