ANALISIS MALWARE PADA SISTEM OPERASI ANDROID MENGGUNAKAN *PERMISSION-BASED*

MALWARE ANALYSIS IN ANDROID OPERATION SYSTEM USING PERMISSION-BASED

Anandika Nur Iman[1], Avon Budiyono, S.T., M.T.[2], Ahmad Almaarif, S.Kom., M.T.[3]

[1,2,3]Program Studi Sistem Informasi, Fakultas Rekayasa Industri, Universitas Telkom

[1]anandikanuriman@student.telkomuniversity.ac.id , [2]avonbudi@telkomuniversity.ac.id,
[3]ahmadalmaarif@telkomuniversity.ac.id

Abstrak

Malware telah berkembang pesat dan semakin banyak jenisnya. Salah satunya ada pada sistem operasi android, atau biasa disebut android malware. Android malware untuk mendapatkan informasi, kontrol dari perangkat, atau membuat kerusakan pada perangkat yang akan merugikan pemiliknya. Analisis malware pada android sangatlah dibutuhkan agar dapat mengetahui dampak, kategori, dan ciri-ciri lainnya dari malware tersebut. Hasil dari analisis yang dilakukan untuk mengetahui hal-hal tersebut. Analisis android malware dilakukan untuk mencari beberapa informasi, salah satunya adalah permission pada aplikasi tersebut. Permission merupakan salah satu fitur yang ada pada android yang digunakan untuk meminta izin untuk menggunakan sesuatu pada perangkat tersebut. Semakin banyak permission yang digunakan dan tidak berhubungan dengan kebutuhan aplikasi maka semakin besar juga kemungkinan bahwa aplikasi tersebut adalah malware. Dampak yang akan diterima pengguna atas permission yang tidak berhubungan tersebut sangat besar. Analisis permission ini menggunakan static analysis yang berupa reverse engineering. Reverse engineering dalam analisis malware berguna untuk ekstraksi data yang memuat informasi yang ada pada malware.

Kata kunci : ***malware, malware analysis, permission, reverse engineering.***

Abstract

Malware has grown rapidly and has more and more types. One of them is on the Android operating system, or commonly called Android malware. Android malware to get information, control from the device, or make damage to the device that will harm the owner. Malware analysis on android is needed to be able to find out the impact, categories, and other characteristics of the malware. The results of the analysis carried out to find out these things. Analysis of android malware is done to find some information, one of which is the permissions on the application. Permission is one of the features on Android that is used to request permission to use something on the device. The more permissions used and not related to application requirements, the greater the possibility that the application is malware. The impact that the user will receive on unrelated permissions is very large. This permission analysis uses static analysis in the form of reverse engineering. Reverse engineering in malware analysis is useful for extracting data that contains information on malware.

Keywords: ***malware, malware analysis, permission, reverse engineering***.

1. Pendahuluan

Perkembangan teknologi saat ini mempermudah manusia untuk saling terhubung menggunakan komputer yang terhubung dengan internet. Dibalik itu terdapat dampak negatif dari perkembangan teknologi ini, salah satunya adalah *cyber crime*. *Cyber crime* adalah perbuatan melawan hukum yang dilakukan dengan menggunakan komputer untuk memperoleh keuntungan ataupun tidak dengan merugikan pihak lain (Wisnubroto, 1999). *Cyber crime* yang marak dilakukan salah satunya adalah memanfaatkan kelemahan sistem jaringan komputer dengan menyusupkan program yang digunakan sebagai media untuk mencuri informasi dari sebuah sistem komputer, program ini disebut sebagai *malware*(Marthur, 2013).

Malware singkatan dari *malicious software* adalah *software* yang digunakan untuk melakukan aktivitas yang sifatnya merusak dan mengganggu (Zeltser, 2014). Pada umumnya penyebaran *malware* menggunakan berbagai cara, seperti *social engineering attack*, *email phising*, dan *downloader*. Tujuan penyebaran *malware* adalah untuk pencurian data rahasia, mencari username dan password, dan spam email (Zeltser, 2014). Selain itu, perangkat lunak yang dianggap sebagai perangkat perusak berdasarkan yang terlihat dan bukan berdasarkan ciri-ciri tertentu, mencakup *Virus Computer*, *Trojan Horse*, *spyware*, *adware crimeware* dan lainnya yang berniat jahat dan tidak diinginkan (G.Kaur, B. Nagpal, 2012).

Berkembangnya teknologi memicu berkembangnya *malware* baru, sehingga semakin banyak pihak yang dirugikan. Bahkan saat ini, *malware* dapat menjangkit hampir seluruh jenis sistem operasi, salah satunya yaitu sistem operasi android. Sistem operasi android adalah sistem operasi seluler *open source* berfitur lengkap pertama yang diciptakan untuk melayani pasar konsumen (Shaerpour, K dan Dehghantanha, A, 2013). *Malware* yang menyerang android disebut android *malware.* Android *malware* menyerang dengan berbagai cara dan salah satunya dengan menggunakan fitur *permission* yang ada pada aplikasi android. Banyak pengguna yang tidak mengerti apa arti dari setiap *permission* dan memberikan *permission* tersebut tanpa memikirkan resiko yang memungkinkan aplikasi tersebut mengakses informasi sensitif pengguna (Franklin, 2014).

Malware analysis adalah suatu aktivitas yang dilakukan untuk mendeteksi ada atau tidaknya *software* yang bertujuan jahat. Karena adanya perkembangan *malware* pada sistem operasi android sehingga *malware analysis* ini akan menjadi solusi untuk mendeteksi adanya *malware*, salah satunya dengan menggunakan *reverse engineering*. *Reverse engineering* dalam analisis *malware* berguna untuk ekstraksi data yang memuat informasi yang ada pada *malware.* Data yang diambil dari hasil *reverse engineering* untuk penelitian ini adalah *permission.* Keuntungan menggunakan *permission-based* ini adalah agar mengetahui kejanggalan dalam sebuah *software* saat meminta akses untuk menggunakan sesuatu didalam *smartphone* yang dimiliki. Pada penelitian ini akan mendapatkan hasil berupa jenis *permission* apa yang paling sering digunakan oleh *malware* agar pengguna android dapat mengetahui dan waspada terhadap jenis *permission* tersebut.

2. Dasar Teori

2.1 Sistem Operasi Android

Sistem operasi android adalah sistem operasi seluler open source berfitur lengkap pertama yang diciptakan untuk melayani pasar konsumen. Sifat open source android dan fitur-fiturnya yang menarik mengarah pada penerimaan global, dan pertumbuhan cepat yang memberikan dorongan besar bagi pasar aplikasi Android

Model keamanan Android mengandalkan sistem operasinya, salah satunya dengan menambahkan permission dalam file "Android Manifest.xml" dalam aplikasinya. Pada prosesnya, pengguna mendapat pertanyaan sebelum instalasi apakah ia ingin memberikan permission yang diminta oleh aplikasi tersebut. Permission tersebut digunakan untuk memastikan fungsi dari aplikasi, apakah sudah tepat atau belum

2.2 *Malicious Software* (*Malware*)

Malware adalah perangkat lunak yang digunakan untuk melakukan perusakan sistem, pencurian atau pengumpulan informasi, hingga mendapatkan akses terhadap suatu computer (Sikorsi & Honig, 2012).
Perangkat lunak yang dianggap sebagai perangkat perusak berdasarkan yang terlihat dari pencipta dan bukan berdasarkan ciri-ciri tertentu, mencakup Virus Computer, Trojan Horse, perangkat pengintai (spyware), perangkat iklan (adware) yang tidak jujur, perangkat jahat (crimeware) dan perangkat lunak lainnya yang berniat jahat dan tidak diinginkan (G.Kaur, B. Nagpal, 2012).
.

2.3 Android *Malware*

Android *malware* adalah perangkat lunak berbahaya yang dibuat untuk menyerang sistem operasi android pada smartphone yang dapat membuat pengguna android mengalami kebocoran informasi rahasia (Saxena, Shrivasta, Mourya, 2016). Android *malware* dapat didefinisikan sebagai sepotong kode yang ditulis dengan niat buruk untuk melakukan tugas-tugas pada perangkat, yang bertujuan untuk mengubah fungsionalitas OS Android tanpa izin dari pengguna, dan dapat membahayakan pengguna untuk kepentingan pembuat malware (N. S. Ismail, 2017).

2.4 Jenis-jenis *Malware*

Beberapa jenis malware android yang di kategorikan sebagai berikut:

a. *Trojan* adalah jenis *malware* yang menyamar sebagai aplikasi *benign* akan tetapi *malware* ini melakukan aktivitas berbahaya di sistem pengguna *smartphone* tanpa ada persetujuan dari pengguna (Karresand, 2002).
b. *Adware* adalah sebuah aplikasi yang diinstall yang berisi iklan yang tidak berbahaya, tetapi melakukan *pop up* terus menerus dan banyak sehingga menyebabkan beberapa gangguan kepada pengguna (Karresand, 2002).
c. *Spyware* adalah sebuah *malware* yang menembus *smartphone* melalui *email*, iklan, kunjungan situs atau aplikasi yang telah diunduh dan akan memonitor dan mencari informasi pribadi, kontak, pesan, atau aktifitas peramban internet (Thanh, 2013).
d. *Worm adalah sebuah malware yang membuat salinannya dan mendistribusikannya melalui jaringan*

seperti misalnya worm bluetooth yang menyebarkan ke perangkat yang terhubung dengan bluetooth tersebut (Schmidt et al., 2004).

e. *Backdoor* adalah sebuah malicious code yang dapat menginstalasi sendiri scriptnya secara otomatis pada suatu komputer. Pada umumnya backdoor menggunakan eksploitasi root untuk memberikan hak akses root ke malware dan memfasilitasi mereka untuk bersembunyi dari antivirus dan memungkinkan penyerang untuk melakukan koneksi pada komputer dengan sedikit bahkan tidak ada otentikasi dan mengeksekusi perintah pada sistem komputer lokal (Sikorsi & Honig, 2012).
f. *Botnets* adalah jaringan perangkat Android yang disusupi seperti *backdoor*. Botmaster, server jarak jauh, mengontrol botnet melalui jaringan C&C. C&C merupakan server yang digunakan sebagai media komunikasi malware dengan penyerang (Y. Zhou and X. Jiang, 2012).
g. *Ransomware* adalah sebuah *malware* yang mencegah pengguna untuk mengakses data mereka didalam perangkat dengan mengunci perangkat tersebut hingga jumlah tebusan yang diminta dibayarkan.

2.5 *Static Analysis*

Static analysis adalah teknik menganalisis *malware* tanpa mengeksekusinya. Teknik ini dapat diterapkan pada representasi yang berbeda dari sebuah program. Jika *source code* tersedia, *tools static analysis* dapat membantu menemukan cacat korupsi memori dan membuktikan kebenaran model untuk sistem yang diberikan (Savan Gadhiya & Kaushal Bhavsar, 2013). Di dalam *static analysis* terdapat satu teknik yang bernama *reverse engineering*. *Reverse engineering* dalam analisis *malware* berguna untuk ekstraksi data yang memuat informasi yang ada pada *malware.* Menggunakan *reverse engineering* merupakan salah satu solusi yang bisa digunakan saat ini.

2.6 Permission

Permissions adalah sebuah fitur yang ada pada aplikasi yang berfungsi untuk meminta hak akses agar aplikasi dapat mengakses beberapa informasi dari smartphone. *Permission* dapat dijadikan salah satu cara untuk menganalisis *malware.* Secara teori, *permissions analysis* dapat membantu pengguna untuk mengidentifikasi *malware*. Permintaan *permission* pada *malware* berbeda dengan *software*. Tujuannya adalah untuk menentukan apakah *permission* adalah alat yang berguna untuk mengidentifikasi praktik *malware*. Karakteristik *malware* yang paling banyak ditemukan adalah penggunaan *permission* SMS. *Malware* meminta kemampuan untuk mengirim pesan tanpa meminta konfirmasi pengguna. (K. A. Talha, D. I. Alper, dan C. Aydin, 2015).

2.7 Metodologi Penelitian

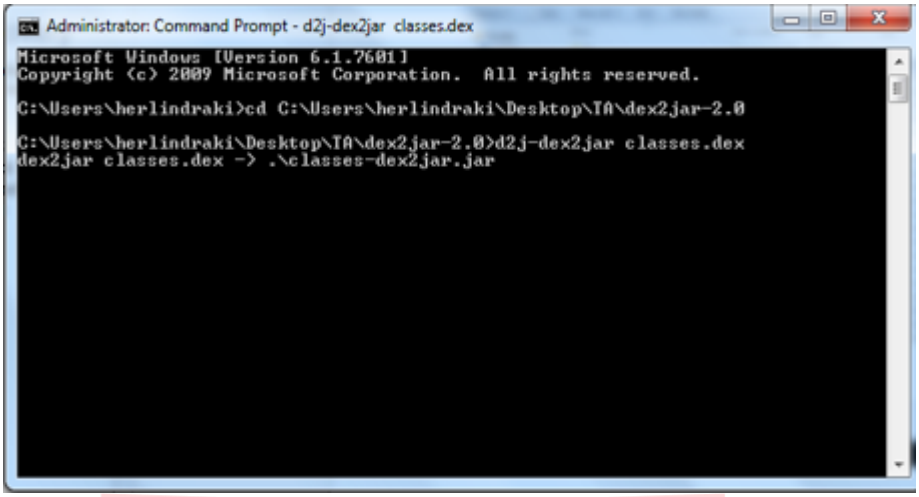
Metode yang digunakan dalam penelitian ini menggunakan model konseptual. Model konseptual adalah deskripsi singkat mengenai bagaimana suatu sistem diorganisasikan dan bekerja (Wiley & Sons, 2002). Fungsi utama dari model konseptual sangat erat hubungannya dengan teori referensi atau literatur yang digunakan. Dengan bantuan model konseptual, peneliti dapat menunjukkan bagaimana melihat fenomena yang ada pada penelitiannya. Konsep-konsep teoritis yang digunakan untuk membangun model konseptual memberikan perspektif atau sebuah cara untuk melihat fenomena empiris (Jonker, J.W, Pennink, & Wahyuni, 2011).

Permasalahan dalam penelitian yaitu dampak malware pada sistem operasi android, deteksi malware menggunakan permission analysis, dan kurangnya pengetahuan dalam android malware. Dengan permasalahan tersebut, terdapat peluang untuk menggunakan permission-based untuk menganalisis malware. Adapun dasar ilmu mengenai teori-teori dan metodologi yang mendukung penelitian ini, sehingga penelitian ini akan menghasilkan artefak berupa analisis malware pada android menggunakan permission-based yang dinilai dari hasil analisis.

3. Pengujian Sistem dan Analisis

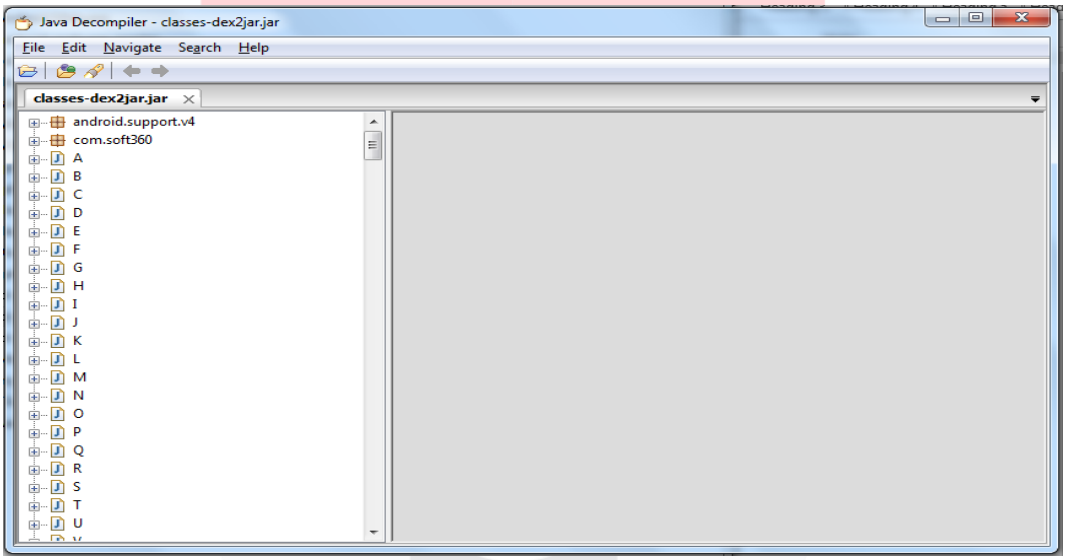
3.1 Pengujian Melakukan *reverse engineering*

Reverse engineering adalah salah satu teknik pada *static analysis. Reverse engineering* dilakukan pada aplikasi yang telah didapatkan dan sudah terindikasi *malware* agar dapat mengetahui *permission* yang ada didalam aplikasi tersebut. Pertama menggunakan tools dex2jar pada file classes.dex yang ada pada aplikasi dan masukkan command d2j-dex2jar classes.dex seperti pada gambar 3-1.



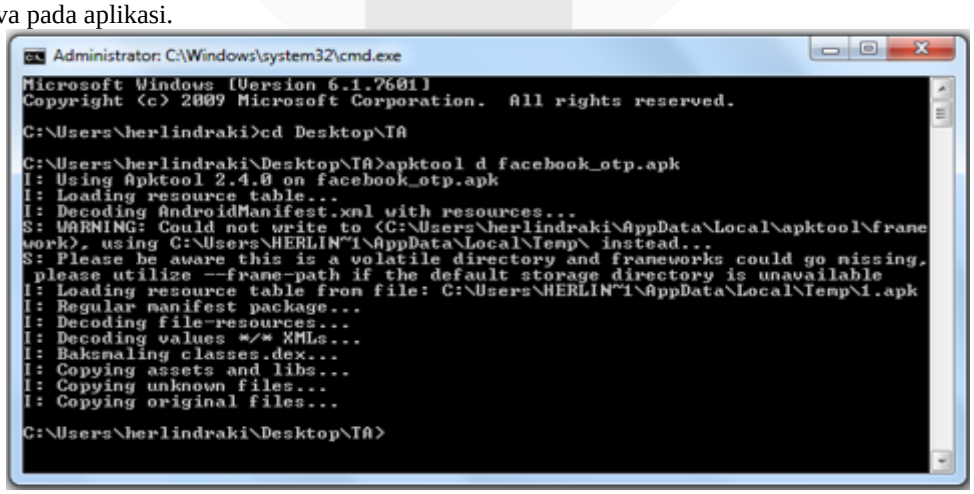
Gambar 1 Menggunakan dex2jar

Setelah melakukan sesuai dengan gambar 1, maka akan mendapatkan hasil berupa file yang telah berubah dari .dex menjadi .jar. Kemudian buka aplikasi jd_gui dan buka file classes.jar seperti pada gambar 2.



Gambar 2 membuka file.jar menggunakan jd_gui

Setelah membuka sesuai dengan gambar 2, simpan file tersebut. File tersebut akan menjadi src pada aplikasi yang telah dibongkar dan dapat dibaca. Setalah itu menggunakan apktool pada aplikasi guna menggenerate selain.java pada aplikasi.



Gambar 3 penggunaan apktool

Pada gambar 3 menjelaskan command yang digunakan apktool agar dapat menggenerate file selain java yang ditujukan pada file .apk secara langsung. Setalah digabungkan dengan src yang didapatkan pada tahap sebelumnya maka *reverse engineering* telah selesai dilakukan dan mendapatkan file yang bisa dibuka dan dalam bentuk source code.

3.2 Melakukan analisis pada *permission*

Analisis dilakukan dengan membuka file android manifest dari hasil *reverse engineering.* Didalam android manifest tersebut terdapat *permission* apa yang digunakan oleh aplikasi tersebut.

Table 1 data analisis

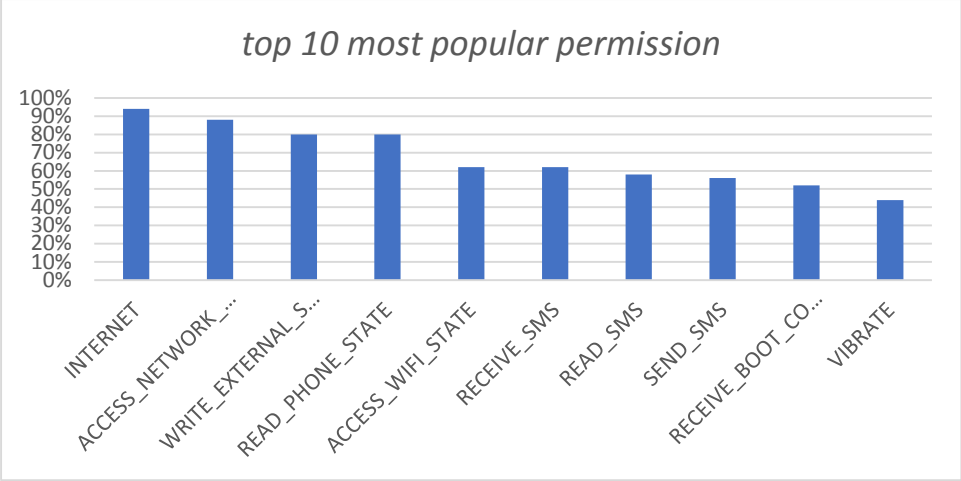
Sample	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Sample1	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓											
Sample2	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓			✓	✓	✓		✓	✓						✓
Sample3	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓											
Sample4	✓		✓	✓																					
Sample5	✓																								
Sample6	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓											
Sample7	✓	✓	✓	✓	✓	✓	✓	✓													✓			✓	
Sample8	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓		✓	✓											
Sample9	✓	✓																							
Sample10	✓	✓		✓	✓																				

Pada tabel 1 menunjukkan *permission* yang digunakan pada salah satu sample *malware*. Dan pada tabel 2 menunjukkan banyaknya jenis *permission* beserta jumlah dari keseluruhan sample.

Tabel 2 hasil seluruh analisis permission

No	Nama permission	Jumlah	%	Deskripsi
1	android.permission.INTERNET	47	94%	Mengizinkan aplikasi untuk membuka socket jaringan
2	android.permission.ACCESS_NETWORK_STATE	44	88%	Mengizinkan aplikasi untuk mengakses informasi mengenai jaringan
3	android.permission.WRITE_EXTERNAL_STORAGE	40	80%	Mengizinkan aplikasi untuk menulis pada external storage
4	android.permission.READ_PHONE_STATE	40	80%	Mengizinkan aplikasi untuk mengakses kondisi ponsel dengan cara membaca. seperti nomor telepon perangkat, akun perangkat yang telah terdaftar dan lain-lain
5	android.permission.ACCESS_WIFI_STATE	31	62%	Mengizinkan aplikasi untuk mengakses informasi mengenai Wi-Fi
6	android.permission.RECEIVE_SMS	31	62%	Mengizinkan aplikasi untuk menerima pesan SMS
7	android.permission.READ_SMS	29	58%	Mengizinkan aplikasi untuk membaca pesan SMS
8	android.permission.SEND_SMS	28	56%	Mengizinkan aplikasi untuk mengirimkan pesan SMS
9	android.permission.RECEIVE_BOOT_COMPLETED	26	52%	Mengizinkan aplikasi menerima Intent.ACTION_BOOT_COMPLETED
10	android.permission.VIBRATE	22	44%	Mengizinkan aplikasi untuk mengontrol getaran

Jenis *permission* yang paling banyak digunakan pada analisa ini adalah internet, *access network state, write external storage, read phone state, access wifi state, receive sms, read sms, send sms, receive boot completed,* dan *vibrate*. Dan gambar 4 menunjukkan grafik dari *top* 10 *most popular permission* pada *malware* yang telah di analisis.



Gambar 4 *top* 10 *most popular permission*

4. Kesimpulan

4.1 Dalam melakukan analisis malware, pada system operasi android guna mengetahui dampak dari *permission* dalam sebuah aplikasi, dilakukan dengan menggunakan *static analysis*. *Static analysis* yang digunakan adalah *reverse engineering* yang menggunakan beberapa aplikasi agar dapat membuka aplikasi tersebut dalam bentuk source code.

4.2 *Permission* dapat digunakan untuk melakukan analisis *malware* dan menentukan bahwa *software* tersebut *benign* atau tidak.

4.3 *Permission* yang banyak digunakan pada sample yang telah dianalisi adalah internet, access network state, write external storage, read phone state, dan access wifi state.

Daftar Pustaka:

[1] Zeltser, L. (2014). What is Malware. The SANS Institute

[2] Sikorski, M. & Honig, A. (2012). Practical Malware Analysis. San Francisco, USA.

[3] Shaerpour, K. and Dehghantanha, A. (2013). Trends in android malware detection. Retrieved September 28, 2018, from http://usir.salford.ac.uk/33894/

[4] Wiley, J., & Sons. (2002). Interaction Design. In Sharp, Roger, & Preece, Beyond Human-Computer 2nd edition (p. 22). Retrieved from http://tambunan.staff.telkomuniversity.ac.id/files/2015/09/03-MI3482Model-Konseptual.pdf. Diakses pada 17 Oktober 2018.

[5] Jonker, J., J.W., B., Pennink, & Wahyuni, S. (2011). Metodologi Penelitian. Panduan Untuk Master Ph.D di bidang Manajemen. Jakarta: Salemba Empat.

[6] Grace, M., Zhou, Y., Zhang, Q., Zou, S., & Jiang, X. (2012). RiskRanker: scalable and accurate zero-day Android malware detection. The 10th International Conference on Mobile Systems, Applications, and Services (MobiSys'12), Low Wood Bay, Lake District, United Kingdom.

[7] Zhou, Y., Wang, Z., Zhou, W., & Jiang, X. (2012). Hey, you, get off of my market: Detecting malicious apps in official and alternative Android markets. Proceedings of the 19th Annual Network and Distributed System Security Symposium, 5-8 February 2012, San Diego, California, USA.

[8] Franklin Tchakount (December 2014). Permission-based Malware Detection Mechanisms on Android: Analysis and Perspectives. JOURNAL OF COMPUTER SCIENCE AND SOFTWARE APPLICATION, 1, 2nd ser., 63-77. Retrieved September 24, 2018.

[9] Yunan Zhang & Qingjia Huang, (2017). Based on Multi-Features and Clustering Ensemble Method for Authomatic Malware Categorization.

[10] Zarni Aung & Win Zaw (march 2013). Permission-Based Android Malware Detection. INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH, 2(3), 228-234. Retrieved October 24, 2018.

[11] Hendra Saputra, Setio Basuki & Mahar Faiqurahman (Mei 2018). Implementasi Teknik Seleksi Fitur Pada Klasifikasi Malware Android Menggunakan Support Vector Machine. Fountain of Informatics

Journal, 3, 1st ser., 12-18. doi:https://dx.doi.org/10.21111/fij.v3i1.1875

[12] Google Developer. (2019, 5 8). *Permissions overview*. Retrieved from developer.android.com: https://developer.android.com/guide/topics/permissions/overview

[13] Google Developers. (2019, Mei 8). App Manifest Overview. Retrieved from developer.android.com: https://developer.android.com/guide/topics/manifest/manifest-intro

[14] Mylonas, A., Theoharidou, M., & Gritzalis, D. (2014). Assessing Privacy Risks in Android: A User-Centric Approach. Lecture Notes in Computer Science, 21–37. doi:10.1007/978-3-319-07076-6_2

[15] Wang, Y., Zheng, J., Sun, C., & Mukkamala, S. (2013). Quantitative Security Risk Assessment of Android Permissions and Applications. Data and Applications Security and Privacy XXVII, 226–241. doi:10.1007/978-3-642-39256-6_15