Seminar Nasional Teknologi dan Inovasi (SITASI)
Vol 1, Nomor 1, Juli 2025, hal. 255-262
P-ISSN: XXXX-XXXX | E-ISSN: XXXX- XXXX
Penerbit Institut Bisnis dan Teknologi Indonesia

Analisis Malware Android Menggunakan Pendekatan Analisis Hybrid

I Gede Adnyana[1*], I Nyoman Buda Hartawan[2], I Nyoman Arnawan[3], Mahesa Rama Aditya[4]

[1,2,3,4] *Institut Bisnis dan Teknologi Indonesia, Denpasar, Indonesia*
[1] *adnyana@instiki.ac.id*,* [2] *buda.hartawan@instiki.ac.id,* [3] *nyoman.arnawan@instiki.ac.id,* [4] *mahesaraditya19@gmail.com*

INFO ARTIKEL

Article history:
Received Juni 2025
Accepted Juli 2025
Published Juli 2025

ABSTRAK

Kejahatan siber di Indonesia terus mengalami peningkatan signifikan dalam beberapa tahun terakhir, seiring dengan meluasnya penggunaan perangkat digital dan konektivitas internet. Perangkat mobile, khususnya yang berbasis sistem operasi Android, menjadi sasaran utama karena sifatnya yang open-source dan memiliki ekosistem aplikasi yang luas, yang memungkinkan terbukanya berbagai celah keamanan. Penelitian ini menggunakan pendekatan analisis hybrid terhadap sejumlah sampel aplikasi berbahaya untuk mengidentifikasi pola dan teknik penyebaran malware. Hasil analisis hybrid menunjukkan bahwa sebagian besar malware Android meminta akses terhadap layanan sensitif seperti SMS, notifikasi, riwayat panggilan, serta lokasi pengguna. Informasi yang berhasil diakses mencakup kode OTP, data keuangan, dan informasi pribadi lainnya yang bernilai tinggi. Data tersebut kemudian dikirimkan ke bot Telegram pelaku, memanfaatkan fitur API dan enkripsi yang disediakan oleh platform tersebut. Teknik ini memberikan lapisan perlindungan tambahan bagi pelaku sekaligus menyulitkan proses pelacakan. Penelitian ini menegaskan pentingnya edukasi pengguna dan penguatan sistem deteksi dini untuk menghadapi ancaman malware secara proaktif dan efektif.
Kata Kunci: kejahatan siber, malware android, bot telegram, analisis hybrid, deteksi dini

ABSTRAK

Cybercrime in Indonesia has continued to increase significantly in recent years, in line with the widespread use of digital devices and internet connectivity. Mobile devices, particularly those based on the Android operating system, have become primary targets due to their open-source nature and extensive application ecosystem, which open up numerous security vulnerabilities. This study employs a hybrid analysis approach on a number of malicious application samples to identify patterns and techniques used in malware distribution. The hybrid analysis results indicate that most Android malware consistently request access to sensitive services such as SMS, notifications, call logs, and user location. The accessed information includes OTP codes, financial data, and other high-value personal information. This data is then transmitted to the attacker's Telegram bot, leveraging the platform's API integration and encryption features. This technique provides an additional layer of protection for the attackers while complicating the tracking process. The study emphasizes the importance of user education and the strengthening of early detection systems to proactively and effectively counter malware threats.
Keywords: cybercrime, android malware, telegram bot, hybrid analysis, early detection

©2025 Authors. Licensed Under CC-BY-NC-SA 4.0

1. Pendahuluan

Data kejahatan siber di Indonesia menunjukkan tren peningkatan yang signifikan. Menurut laporan dari Kementerian Komunikasi dan Informatika (Kominfo) dan Badan Siber dan Sandi Negara (BSSN), lonjakan serangan siber, terutama yang menggunakan malware pada perangkat mobile, semakin mengkhawatirkan. Peningkatan kasus kejahatan siber sebesar 35–40% dalam beberapa tahun terakhir dan urgensi peningkatan sistem keamanan informasi menjadi hal terpenting saat ini.(Butarbutar, 2023; Chintia et al., 2019)

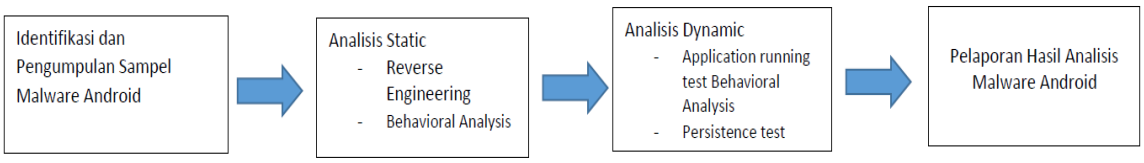
Platform Android, yang mendominasi pasar smartphone di Indonesia, menjadi target empuk bagi para pelaku kejahatan siber. Sifat sistem operasi Android yang bersifat open-source dan ekosistem aplikasinya yang beragam memberikan peluang bagi peretas untuk mengeksploitasi celah keamanan melalui teknik-teknik seperti obfuscation dan anti-analisis. Kerentanan ini kian diperparah oleh kurangnya penerapan standar keamanan yang memadai di sebagian besar aplikasi, sehingga meningkatkan risiko infeksi malware. (Ardita et al., 2022)

Berbagaii penelitian telah dilakukan dengan pendekatan beragam untuk menganalisis dan mendeteksi malware Android. Metode *reverse engineering* digunakan untuk membongkar struktur internal aplikasi dan memahami perilaku mencurigakan dari kode berbahaya (Saputro et al., 2020). Selain itu, analisis statis menjadi salah satu pendekatan yang banyak digunakan, seperti ditunjukkan dalam studi yang menyoroti pemeriksaan fitur-fitur seperti permission, API call, serta kode sumber untuk proses klasifikasi malware (Sinambela et al., 2020; Yuniati et al., 2022). Beberapa studi juga menyoroti penyebaran malware melalui aplikasi perpesanan instan dengan analisis berbasis reverse engineering (Adnyana; et al., 2024). Di sisi lain, pendekatan hybrid yang menggabungkan metode statis dan dinamis terbukti mampu memberikan pemahaman yang lebih menyeluruh terhadap tujuan pembuatan malware (Kusuma et al., 2020).

Untuk mengatasi tantangan dari meningkatnya kompleksitas teknik penyembunyian malware, pendekatan *hybrid analysis* menjadi solusi inovatif dengan menggabungkan keunggulan analisis statis dan dinamis secara simultan. Analisis statis memungkinkan identifikasi awal terhadap potensi kode berbahaya tanpa perlu mengeksekusi aplikasi (Himawan et al., 2022), sementara analisis dinamis mengamati perilaku aplikasi secara langsung dalam lingkungan terkontrol, sehingga pola-pola serangan yang kompleks dapat dikenali dengan lebih akurat (Karim et al., 2020). Dengan terus meningkatnya insiden kejahatan siber, penelitian yang mengembangkan metode deteksi malware berbasis hybrid diharapkan mampu menghasilkan sistem deteksi yang lebih adaptif, efisien, dan tangguh dalam menghadapi ancaman keamanan siber di era digital.

2. Metode Penelitian

Adapun langkah-langkahnya :



Gambar 1. Langkah-langkah Pendekatan Analisis Hybrid
(Sumber : Penulis, 2025)

a. Identifikasi dan Pengumpulan Sampel Malware Android

Pada tahap identifikasi, peneliti mengandalkan berbagai sumber terpercaya, seperti repositori malware, dan platform berbagi data riset guna mengidentifikasi varian malware yang relevan dan representatif.

Setelah identifikasi, pengumpulan sampel dilakukan dengan cara mendownload dan mengakuisisi file APK yang telah terverifikasi sebagai malware melalui proses otomatisasi maupun manual.

Tabel 1. Sampel Malware Android

No.	Nama File	Ukuran	Hash
1	Surat tilang elektronik.apk	5.626 MB	ee3012d3c71e51e6b33727b9e72f9915
2	UndanganPernikahan.apk	15.155 MB	746843e47430f29a04ef40a1012b66f4
3	Lihat Foto Paket.apk	5.761 MB	2a58e5989d9e248b918670382c3fb4e7
4	video viral.apk	5.890 MB	3da36d75040f8815843127b2aaa89e7f
5	buka_video.apk	5.834 MB	33ef8ef88905b6a526cf35feb99388bf
6	surat E-TILANG.apk	5.566 MB	26b3a9ba8b5fbd438db2ec67624f80b6
7	Undangan Hahal Bi Halal.apk	6.046 MB	ac2ad0bcca38a79a12d98731bd71ab2f
8	Undangan pernikahan cek lokasi detail.apk	5.726 MB	5a8d5e7399de9ef30d7e19642c916b26
9	UNDANGAN.PERNIKAHAN_.apk	5.368 MB	efeea09f142acfa4deaab412f1c52fe3

b. Analisis Static
   Analisis Statis dilakukan dengan teknik reverse engineering untuk membedah struktur APK dan meninjau kode sumber, serta behavioral analysis secara statis guna mengidentifikasi pola atau ciri khas yang menunjukkan potensi aktivitas berbahaya.
c. Analisis Dynamic
   Analisis Dinamis, sampel malware dijalankan di lingkungan terisolasi (sandbox) untuk melakukan application running test dan behavioral analysis secara langsung, termasuk persistence test untuk menilai kemampuan malware bertahan di sistem.
d. Pelaporan Hasil Analisis Malware Android
   Seluruh temuan dari kedua metode analisis tersebut didokumentasikan dalam Pelaporan Hasil Analisis Malware Android, yang menyajikan ringkasan pola serangan, teknik obfuscation, dan rekomendasi mitigasi untuk meningkatkan keamanan sistem.

3. Hasil dan Pembahasan

3.1 Analisis Static

Tahap analisis statis melibatkan pemeriksaan file AndroidManifest.xml, yang berisi daftar izin (permissions) yang diminta oleh aplikasi. Sebagai contoh, dalam file yang disediakan, aplikasi meminta izin untuk menerima dan mengirim SMS, mengakses internet, membaca SMS, dan lainnya. Izin-izin tersebut dapat menunjukkan potensi bahaya jika disalahgunakan untuk tujuan jahat.

Selanjutnya, analisis source code dilakukan dengan memeriksa struktur kode untuk menemukan bagian-bagian yang mencurigakan, seperti fungsi yang menangani data sensitif (misalnya kontak, pesan, atau informasi pribadi), kemudian menganalisis alur data untuk melihat bagaimana data dikumpulkan, diproses, dan dikirimkan. Proses ini dapat mengungkap aktivitas berbahaya seperti pencurian data atau pengiriman data ke server eksternal.

AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:versionCode="1" android:version
    <uses-sdk android:minSdkVersion="27" android:targetSdkVersion="32"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.ACCESS_NOTIFICATION_POLICY"/>
    <uses-permission android:name="android.permission.CALL_PHONE"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
    <application android:theme="@style/Theme.AppCompat.NoActionBar" android:label="Surat tilang elektronik"
        <activity android:name="com.example.ZzZZrrrrsd.HomeActivity" android:exported="false">
            <meta-data android:name="android.app.lib_name" android:value=""/>
        </activity>
```

Gambar 2. Potongan AndroidManifest.xml
(Sumber : Penulis, 2025)

Pada Gambar 2 ditampilkan, aplikasi ini meminta sejumlah izin yang tergolong sangat sensitif, termasuk akses lokasi (baik kasar maupun akurat), menerima dan membaca SMS, mengirim SMS, melakukan panggilan telepon, serta mengakses internet dan status jaringan. Selain itu, aplikasi juga meminta izin untuk tetap aktif di latar belakang (WAKE_LOCK), berjalan otomatis saat perangkat dinyalakan (RECEIVE_BOOT_COMPLETED), serta menjalankan layanan latar depan (FOREGROUND_SERVICE).
Selanjutnya dilakukan analisis file MainActivity untuk melihat alur data akan dikirimkan

```java
public void onReceive(Context context, Intent intent) {
    String stringExtra = intent.getStringExtra("package");
    String stringExtra2 = intent.getStringExtra("title");
    String stringExtra3 = intent.getStringExtra("text");
    intent.getStringExtra("id");
    HomeActivity.this.client.newCall(new Request.Builder().url("https://api.telegram.org/bot" + HomeActivity.this.token +
        @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("demo1", "OnResponse: Thread Id " + Thread.currentThread().getId());
            if (response.isSuccessful()) {
                response.body().string();
            }
        }
    });
    HomeActivity.this.client.newCall(new Request.Builder().url("https://api.telegram.org/bot" + HomeActivity.this.token +
        @Override // okhttp3.Callback
        public void onFailure(Call call, IOException iOException) {
            iOException.printStackTrace();
        }
```

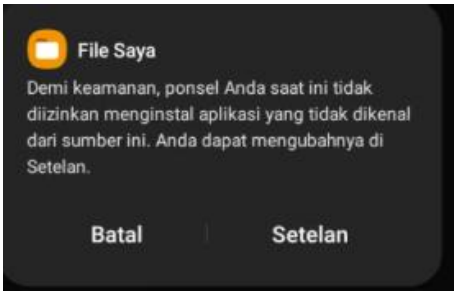
Gambar 3 Potongan MainActivity file apk
(Sumber : Penulis, 2025)

Berdasarkan hasil analisis kode pada file HomeActivity pada Gambar 3, terlihat bahwa aplikasi ini menginisiasi koneksi HTTP menggunakan pustaka okhttp3 dan mengirimkan data ke sebuah bot Telegram melalui URL https://api.telegram.org/bot. Penggunaan token bot Telegram yang tersimpan dalam variabel HomeActivity.this.token menunjukkan bahwa aplikasi secara aktif mengirimkan informasi ke kanal eksternal tanpa sepengetahuan pengguna

3.2 Analisis Dynamic

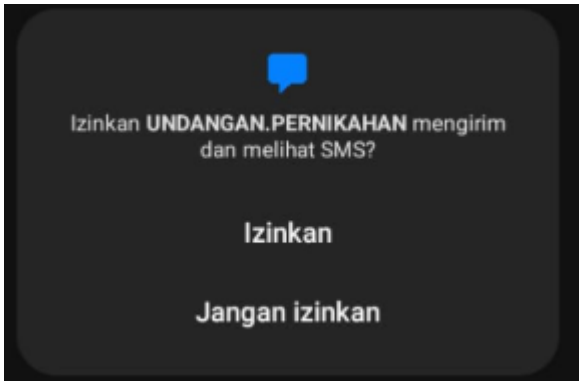
Analisis dinamis dilakukan dengan menginstal aplikasi yang dicurigai sebagai malware pada perangkat uji guna mengamati perilaku aktual saat aplikasi dijalankan. Setelah proses instalasi, aplikasi

menunjukkan gejala mencurigakan melalui notifikasi yang muncul secara tiba-tiba, seperti permintaan izin yang tidak wajar atau peringatan yang menyesatkan. Selain itu, melalui pemantauan jaringan dan log sistem, terdeteksi adanya aliran data yang dikirimkan ke server eksternal tanpa sepengetahuan pengguna. Sebelum dilakukan instalasi, dilakukan konfigurasi API telegram pada file apk untuk memudahkan menganalisis aliran data yang dikirimkan.



Gambar 4. Notifikasi peringatan saat mau instalasi apk
(Sumber : Penulis, 2025)

Gambar 4 menunjukkan peringatan keamanan dari sistem operasi Android ketika mencoba menginstal file APK yang tidak dikenal. Setelah itu, izin instalasi tetap diberikan agar proses instalasi dapat dilanjutkan. Hal ini dilakukan untuk keperluan analisis, guna mengamati perilaku aplikasi yang terindikasi sebagai malware serta melihat apakah aplikasi tersebut mengirimkan data sensitif. Setelah berhasil diinstal, selanjutkan akan muncul notifikai untuk meminta izin mengirimkan SMS seperti yang ditunjukkan pada Gambar 5.



Gambar 5. Notifikasi permintaan izin SMS
(Sumber : Penulis, 2025)

Setelah pengguna secara tidak sadar memberikan izin kepada aplikasi untuk mengakses dan mengirimkan SMS, malware yang tersembunyi di dalam aplikasi tersebut mulai beroperasi secara aktif. Setiap pesan SMS yang masuk ke perangkat korban secara otomatis akan dibaca dan kemudian diteruskan ke bot Telegram yang dikendalikan oleh pelaku. Proses ini berlangsung secara diam-diam tanpa sepengetahuan korban, memungkinkan pelaku untuk memantau komunikasi yang masuk, termasuk kode verifikasi, informasi pribadi, atau data sensitif lainnya.

Gambar 6. Konten SMS yang diterima oleh bot Telegram
(Sumber : Penulis, 2025)

Gambar 6 memperlihatkan bukti bahwa setiap konten SMS yang diterima oleh perangkat korban secara otomatis diteruskan ke bot Telegram milik pelaku. Mekanisme ini merupakan bagian dari aktivitas berbahaya malware yang bekerja di latar belakang tanpa diketahui oleh pengguna. Jika isi SMS tersebut mengandung data sensitif, seperti kode OTP, informasi akun, atau pesan rahasia lainnya, maka hal ini dapat menimbulkan kerugian besar bagi korban, baik secara finansial maupun dari segi privasi. Dengan mengakses konten SMS, pelaku dapat mengambil alih akun pribadi korban, melakukan penipuan, atau menjual informasi tersebut ke pihak lain

3.3 Hasil Analisis Malware Android

Hasil analisi statis menunjukkan struktur malware meminta izin untuk mengakses layanan perangkat yang telah terinfeksi malware seperti akses SMS, notifikasi, melakukan panggilan bahkan meminta akses lokasi. Pada tabel 2, dapat dilihat bahwa sebagian besar file APK berbahaya (malware) meminta izin akses terhadap SMS, yang menjadi target utama untuk memperoleh informasi sensitif pengguna, seperti kode OTP, informasi transaksi, dan data pribadi lainnya. Beberapa file juga meminta izin tambahan seperti lokasi dan akses telepon, yang mengindikasikan upaya pengumpulan data lebih luas untuk profiling korban atau pelacakan. Izin notifikasi yang diminta pada beberapa file berpotensi digunakan untuk menyembunyikan aktivitas jahat, seperti intercepting pesan masuk dan mencegah pengguna melihat notifikasi asli dari aplikasi sah.

Tabel 2. Permintaan Perijinan dan Pengiriman Data Malware

No.	Nama File	Perijinan yang diminta	Pengiriman Data
1	Surat tilang elektronik.apk	Lokasi, SMS, Notifikasi, Telpon	Bot Telegram
2	UndanganPernikahan.apk	SMS	Bot Telegram
3	Lihat Foto Paket.apk	SMS	Bot Telegram
4	video viral.apk	SMS	Bot Telegram
5	buka_video.apk	SMS	Bot Telegram
6	surat E-TILANG.apk	Lokasi, SMS, Notifikasi, Telpon	Bot Telegram
7	Undangan Hahal Bi Halal.apk	SMS	Bot Telegram
8	Undangan pernikahan cek lokasi detail.apk	SMS	Bot Telegram
9	UNDANGAN.PERNIKAHAN_.apk	SMS	Bot Telegram

Selain perizinan, pola pengiriman data dari malware ini cenderung seragam, yaitu menggunakan Bot Telegram sebagai media komunikasi. Hal ini menunjukkan bahwa pelaku memanfaatkan Telegram sebagai command-and-control (C2) server karena kemudahan integrasi API dan sifatnya yang terenkripsi.

Dengan mengirimkan data curian ke akun bot Telegram, pelaku dapat mengakses informasi korban secara real-time tanpa perlu mengatur server sendiri, yang sekaligus mempersulit pelacakan oleh pihak berwenang. Strategi ini menunjukkan tingkat efisiensi dan adaptabilitas tinggi dalam operasi malware modern.

Hasil analisis dinamis menunjukkan adanya kesesuaian dengan temuan pada analisis statis, di mana setelah proses instalasi berhasil, aplikasi secara langsung meminta sejumlah izin akses penting dari sistem Android. Izin tersebut mencakup akses untuk membaca dan mengirim SMS, mengelola notifikasi, melakukan panggilan telepon, serta mengakses lokasi perangkat. Permintaan akses ini menjadi indikasi awal bahwa aplikasi mencoba mengeksploitasi fitur-fitur sensitif pada perangkat untuk tujuan yang tidak sah. Pada tahap ini, pengguna yang tidak waspada mungkin akan memberikan izin tanpa menyadari potensi risikonya, terutama jika aplikasi menyamar sebagai aplikasi yang sah atau fungsional.

Ketika semua izin tersebut diberikan, malware mulai menjalankan fungsinya secara aktif. Salah satu perilaku yang diamati adalah pengiriman otomatis pesan SMS yang masuk ke perangkat korban menuju bot Telegram milik pelaku. Alur ini menunjukkan bahwa aplikasi telah dirancang dengan kemampuan untuk menyadap komunikasi pribadi dan menyalurkannya ke server eksternal tanpa diketahui pengguna. Aktivitas semacam ini tidak hanya melanggar privasi, tetapi juga dapat dimanfaatkan untuk aksi kejahatan seperti pencurian identitas, penyalahgunaan informasi rahasia, hingga pengambilalihan akun-akun penting yang menggunakan verifikasi berbasis SMS

4. Kesimpulan

Berdasarkan hasil analisis statis dan dinamis, dapat disimpulkan bahwa malware Android yang dianalisis memiliki pola perilaku berbahaya yang konsisten, baik dari sisi struktur izin maupun aktivitas pasca-instalasi. Malware ini secara agresif meminta akses terhadap layanan penting seperti SMS, notifikasi, panggilan, dan lokasi, yang bertujuan untuk mengumpulkan data sensitif pengguna. Target utama dari serangan ini adalah pesan SMS, terutama yang berisi informasi rahasia seperti kode OTP dan data transaksi, serta potensi penyalahgunaan izin lain untuk pelacakan dan penyamaran aktivitas jahat. Penggunaan bot Telegram sebagai media komunikasi utama antara malware dan pelaku menunjukkan adanya strategi yang canggih dan efisien. Dengan memanfaatkan platform yang terenkripsi dan mudah diintegrasikan, pelaku dapat secara real-time menerima informasi curian tanpa perlu membangun infrastruktur server sendiri, yang juga menyulitkan proses pelacakan. Temuan ini menegaskan pentingnya kewaspadaan pengguna terhadap aplikasi yang meminta akses berlebihan, serta perlunya penguatan sistem deteksi dini terhadap aplikasi berpotensi berbahaya di perangkat Android.

5. Ucapan Terima Kasih

Dengan penuh rasa hormat dan apresiasi, kami mengucapkan terima kasih yang sebesar-besarnya kepada Direktorat Riset dan Pengabdian kepada Masyarakat (DRPM) Institut Sains dan Teknologi Indonesia (INSTIKI) atas dukungan pendanaan yang telah diberikan melalui skema IRDP Tahun 2025.

Daftar Pustaka

Adnyana;, I. G. A., Nugraha;, P. G. S. C., & Nugroho, B. R. A. (2024). Reverse Engineering for static analysis of Android Malware in Instant Messaging. *Journal of Computer Networks, Architecture and High Performance Computing*, *6*(3), 1460–1469. https://jurnal.itscience.org/index.php/CNAPC/article/view/4417/3353

Ardita, I. K. A. O., Anom Cahyadi Putra, I. G. N., Kustiadie, M. R., Dika Varuna, G. N. M., & Eka Prananda, M. Y. (2022). Analisis Keamanan Aplikasi Android Dengan Metode Vulnerability Assessment. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*. https://doi.org/10.24843/jlk.2022.v10.i03.p04

Butarbutar, R. (2023). Kejahatan Siber Terhadap Individu : Jenis , Analisis , Dan Perkembangannya. *Journal Technology and Economiacs Law*, *2*(2), 297–316.

Chintia, E., Nadiah, R., Ramadhani, H. N., Haedar, Z. F., Febriansyah, A., & Rakhmawati S.Kom., M.Sc.Eng, N. A. (2019). Kasus Kejahatan Siber yang Paling Banyak Terjadi di Indonesia dan

Penanganannya. *Journal of Information Engineering and Educational Technology*. https://doi.org/10.26740/jieet.v2n2.p65-69

Himawan, I., Septianzah, K., & Setiadi, I. (2022). Analisis Keamanan Informasi Malware Terhadap Aplikasi Apk Dengan Metode Static Analysis Menggunakan Mobsf. *JRKT (Jurnal Rekayasa Komputasi Terapan)*. https://doi.org/10.30998/jrkt.v2i02.6734

Karim, A., Chang, V., & Firdaus, A. (2020). Android botnets: A proof-of-concept using hybrid analysis approach. *Journal of Organizational and End User Computing*. https://doi.org/10.4018/JOEUC.2020070105

Kusuma, S. D., Akbi, D. R., & Risqiwati, D. (2020). Analisis Malware Berdasarkan Tujuan Pembuatan Dengan Menggunakan Metode Hybrid Pada Android. *Jurnal Repositor*. https://doi.org/10.22219/repositor.v2i8.764

Saputro, B. A., Alfitra, L. I., & Oktaviaji, R. B. (2020). Analisis Malware Android Menggunakan Metode Reverse Engineering. *Jurnal Repositor*. https://doi.org/10.22219/repositor.v2i10.1061

Sinambela, S., Pangestu, A. R., & Feriyanto, R. (2020). Analisis Aplikasi Malware pada Android dengan Metode Statik. *Jurnal Ilmiah ILKOMINFO - Ilmu Komputer & Informatika*. https://doi.org/10.47324/ilkominfo.v3i2.101

Yuniati, T., Tambunan, A. R., & Setyoko, Y. A. (2022). Implementasi Static Analysis Dan Background Process Untuk Mendeteksi Malware Pada Aplikasi Android Dengan Mobile Security Framework. *LEDGER : Journal Informatic and Information Technology*. https://doi.org/10.20895/ledger.v1i2.848