BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Penelitian Terkait

Penelitian yang akan dilakukan tidak terlepas dari temuan penelitian sebelumnya yang dihasilkan sebagai bahan penelitian dan perbandingan. Penelitian yang digunakan sebagai perbandingan tidak terlepas dari topik penelitian yang dilakukan peneliti yaitu mengenai *malware*, *exploit*, dan *reverse engineering*. Berikut merupakan beberapa penelitian terkait dengan topik penelitian yang dilakukan peneliti:

1. **Aan Kartono pada tahun 2019 yang membahas mengenai "Membangun Sistem Pengujian Keamanan Aplikasi Android Menggunakan MobSF"**

   Latar belakang dari penelitian ini dikarenakan banyaknya aplikasi android yang telah disusupi oleh *malware* berbahaya sehingga dapat merusak ponsel itu sendiri. Pengaruh negatif dapat ditimbulkan dari adanya aplikasi android yang disusupi *malware*. Hal ini dikarenakan adanya pencuri data pengguna yang tersimpan pada ponsel tersebut. Hasil penelitian menunjukkan bahwa *malware* pada aplikasi android dapat dianalisis *Mobile Security Framework* (*MobSF*) melaksanakan analisis statis dan dinamis, serta memberikan laporan bahwasannya aplikasi android bebas dari adanya *malware* berbahaya yang ditimbulkan dari adanya campur tangan pihak tidak sah. Proyek akhir ini menggunakan *Mobile Security Framework* (*MobSF*) sebagai perangkat lunak untuk menganalisis aplikasi android. *Malware* android yang dapat terdeteksi adalah *Trojan*, *Ransomware*, *Adware*, dan sebagainya. Mobile Security Framework (MobSF) juga dapat memberitahu kepada peneliti *source code* berbahaya pada sebuah aplikasi android[9].

2. **Rizky Dwiananda Lukita Putra pada tahun 2019 yang membahas mengenai "Eksploitasi dengan Metode Reverse_TCP di Perangkat Android Menggunakan Metasploit**"

   Kesimpulan dari penelitian ini menyatakan bahwa android adalah sistem operasi yang disukai banyak orang pada *smartphone* yang berjalan pada *kernel linux*. Salah satu keuntungan dari sistem operasi android memungkinkan pengembang dapat mengakses dan memodifikasi kode sumber. Namun, dibalik keuntungannya terssebut juga memberikan dampak meningkatnya masalah keamanan. Sebagai contoh *exploit*, banyak penyerang melakukan serangan tersebut untuk mendapatkan informasi sensitif pengguna. Pada penelitian ini *tool* yang digunakan untuk melancarkan serangan *exploit* adalah *framework metasploit* dan percobaan ini dilakukan pada perangkat android yang menjadi sasaran serangan. Hasil yang diperoleh dari serangan *exploit* yakni pemahaman tentang proses melakukan *exploit attack*, menganalisis hubungan timbal-balik antara penyerang dengan korban, metode *exploit attack* dan waktu yang diperlukan untuk menyiapkan *payload* dengan *bash script shell*. Semua detail analisis akan memberi kesimpulan dan strategi untuk membuat sistem operasi android lebih aman[10].

3. **Aldy Putra Aldya pada tahun 2019 yang membahas mengenai "Reverse Engineering untuk Analisis Malware Remote Access Trojan**"

   Penelitian tersebut menjelaskan bahwa rekayasa balik merupakan solusi untuk melaksanakan analisis *malware.* Hal ini dikarenakan kode pada *malware* dapat diketahui dengan menggunakan teknik tersebut. Malware Flawed ammyy merupakan *software* yang disalahgunakan dari Ammyy admin versi 3 oleh hacker TA505. Tujuan dari penelitian ini untuk mengetahui jalan melakukan identikasi *malware* khususnya *malware RAT* dengan teknik *reverse engineering* serta *tools* dipakai. Penelitian ini menggunakan metodologi deskriptif.

Temuan penelitian menunjukkan bahwasannya alur untuk melakukan *reverse engineering* serta *tools* yang dapat dipakai[11].

4. **Cholis Hanifurohman pada tahun 2020 membahas mengenai "Analisis Statis Menggunakan Mobile Security Framework Untuk Pengujian Keamanan Aplikasi Mobile E-Commerce Berbasis Android**"

   Penelitian ini bertujuan untuk memberikan pemahaman yang ditujukan kepada pengguna aplikasi *mobile e-commerce* terhadap celah keamanan aplikasi tersebut serta memberikan cara kerja dalam melaksanakan analisis statis menggunakan *mobile security framework (MobSF)* untuk melakukan pengujian keamanan terhadap aplikasi khususnya yang berbasis android. Analisis statis dilakukan dengan memeriksa kekurangan kriptografi (*weak crypto*), *SSL bypass*, penggunaan *dangerous permission*, *hardcode secret*, *root detection* dan *domain malware check*. Metode yang digunakan merupakan *mobile security framework.* Sistem ini memiliki tiga tingkatan yang di antaranya kebutuhan perencanaan, proses desain RAD dan implementasi. Hasil analisis keamanan keamanan yang dilakukan pada aplikasi *mobile e-commerce* yaitu SP, TP, LZ, BL dan SR yang merupakan lima teratas *mobile e-commerce* dengan basis android paling disukai di Indonesia menunjukkan bahwa beberapa celah keamanan masih terdapat dari di kelima aplikasi hasil tersebut yang perlu diketahui baik oleh pengguna maupun pengembang aplikasi[12].

5. **M. Alvian H Nasution pada tahun 2020 yang membahas mengenai "Investigasi Serangan Backdoor Remote Access Trojan (RAT) terhadap Smartphone**"

   Latar belakang dari munculnya penelitian ini dikarenakan adanya kemungkinan *smartphone* terutama pengguna android yang terkena tindak kejahatan ataupun serangan sehingga dapat membahayakan informasi pribadi atau data diri yang ada di dalamnya. Hal ini dikarenakan android dapat dengan mudah dilakukan penyisipan

malware dengan jenis *RAT* atau *Remote Access Trojan* yang sengaja diciptakan dengan alat ngrok bermacam ekstensi file seperti .jpg, .Mp4, ataupun .apk seperti yang dilakukan dan akan dibahas pada jurnal. Pada jurnal ini juga dilakukan uji coba penanaman RAT pada *smart*phone android yang menjadi sasaran, yang pada akhirnya akan memberi smartphone akses penuh ke semua direktori setelah penginstalan RAT yang disembunyikan dalam file.apk. Pengujian ini mengharapkan pengguna *sma*rtphone khususnya android supaya tidak melakukan *root* secara pribadi, karena memungkinkan celah keamanan terbuka dari *device* yang digunakan serta lebih mudah bagi individu yang tidak memiliki bertanggung jawab melakukan serangan[13].

6. **Deco Aprilliansyah pada tahun 2021 yang membahas mengenai "Analisis Remote Access Trojan Attack menggunakan Android Debug Bridge**"

   Celah keamanan pada sistem operasi android terkadang tidak disadari oleh pengguna seperti *malware* dan eksploitasi oleh pihak ketiga hingga akses jarak jauh. Penelitian ini dilakukan untuk mengidentifikasi kerentanan sistem operasi android dengan menggunakan *Ghost Framework*. Kerentanan *smartphone* android ditemukan dengan menggunakan *Android Debug Bridge* (*ADB*) dengan metode eksploitasi serta menganalisis hasil pengujian dan mengidentifikasi serangan trojan akses jarak jauh. Metode eksploitasi dengan beberapa langkah dari menyiapkan alat dan menghubungkan perintah pengujian ke perangkat pengujian telah dilakukan. Hasilnya menunjukkan bahwa android versi 9 dapat diakses dari jarak jauh dengan memasukkan eksploit melalui *ADB*. Beberapa informasi telah diperoleh oleh pihak ketiga, masuk dan mengubah isi direktori sistem dengan akses jarak jauh seperti berwenang melakukan aktivitas apapun pada perangkat seperti membuka layar kunci, memasuki sistem direktori, mengubah sistem, dan lain-lain[14].

7. **Imam Riadi pada tahun 2022 yang membahas mengenai "Mobile Device Security Evaluation using Reverse TCP Method"**

   Membahas mengenai pentingnya evaluasi keamanan pada perangkat android. Evaluasi keamanan ini bertujuan agar pengguna sistem operasi terlindungi dari serangan *malware* seperti *trojan* akses jarak jauh yang dapat mencuri data kredensial pengguna. Serangan *remote access trojan* (RAT) dapat diantisipasi dengan mendeteksi kerentanan pada aplikasi dan sistem. Penelitian ini mensimulasikan serangan *trojan* akses jarak jauh dengan mengeksploitasinya hingga penyerang mendapatkan akses penuh ke perangkat korban. Episode dilakukan dengan beberapa tahapan seperti membuat *payload*, menginstal aplikasi ke perangkat korban, menghubungkan pendengar, dan melakukan eksploitasi untuk mengambil informasi penting di perangkat korban. Materi pengujian menggunakan android 12, terjadi masalah saat mencoba menginstal aplikasi karena akan muncul peringatan berbahaya dari *play protect* akibat tidak menggunakan perlindungan privasi versi terbaru yang menyebabkan aplikasi terindikasi *malware* dan sejenisnya. Pada android 11, aplikasi yang diinjeksi dengan *backdoor* berhasil dipasang di perangkat dan berhasil diakses oleh penyerang. Penyerang juga mendapatkan informasi penting, termasuk informasi sistem, kontak, log panggilan, pesan, dan akses penuh ke direktori sistem perangkat korban. Berdasarkan penelitian ini, pengguna perangkat android diharapkan untuk selalu memperbarui versi android pada perangkat yang digunakannya[15].

Tabel 2.1 Penelitian Terdahulu

No	Penelitian	Studi Kasus	Tools	Metode	Vunarebility Malware	Hasil Penelitian	Perbedaan
1.	Membangun Sistem Pengujian Keamanan Aplikasi Android Menggunakan MobSF (2019) Aan Kartono, Anang Sularsa dan Setia Juli Irzal Ismail	Android Versi 0,99, 3,0 dan 11,5	Metasploit MobSF	PTES	Trojan	Hasil yang didapatkan dari *file* yang dianalisis berbeda-beda, tergantung dari sampel, metode analisis dan *tool*s yang digunakan.	Metode yang digunakan dan jenis malware yang digunakan berbeda begitupun dengan hasil yang didapatkan .
2.	*Exploitation with Reverse _tcp method on Android Device Using Metasploit* (2019) Rizky Dwiananda Lukita Putra dan Is Mardianto	Android Versi 7.1.2 Nougat	*Metasploit framework*	*Reverse tcp*	*Exploit*	Karakteristik serangan *exploit* memanfaatkan dua *bug* yang terdapat pada sistem operasi android, yaitu pada penyimpanan memori fisik dan servis binder yang berjalan pada *DRM (digital right management)* memungkinkan aplikasi mengelola konten yang dilindungi hak sesuai dengan batasan lisensi yang terkait dengan konten.	Versi android yang digunakan berbeda dab hasil yang didapatkan juga berbeda
3.	*Reverse Engi-neering* untuk Analisis *Mal-ware Remote Access Trojan* (2019)	*Malware Flawed Ammyy RAT*	Virus total	*Reverse Enginee ring*	*Remote Access Trojan*	Hasil penelitian menunjukan bahwa alur untuk melakukan *reverse*	Tools yang digunakan berbeda.

No	Penelitian	Studi Kasus	Tools	Metode	Vunarebility Malware	Hasil Penelitian	Perbedaan
	Aldy Putra Aldya, Nur Widiyasono dan Tesa Pajar Setia.					*engineering* dan *tools* yang dapat digunakan.	
4.	Analisis Statis Menggunakan *Mobile Security Framework* Untuk Pengujian Keamanan Aplikasi *Mobile E-Commerce Berbasis Android (2020) Cholis Hanifurohman dan Deanna Durbin Hutagalung*	*Mobile e-commerce*	*Mobile Security Framework*	Analisis Statis	*Kriptografi (weak crypto),*	Aplikasi *mobile e-commerce* yaitu SP, TP, LZ, BL dan SR menunjukkan bahwa beberapa celah keamanan masih terdapat dari kelima aplikasi hasil tersebut yang perlu diketahui baik oleh pengguna maupun pengembang aplikasi.	Jenis malware yang digunakan berbeda dengan penelitian ini.
5.	Investigasi Serangan *Backdoor Remote Access Trojan* (RAT) terhadap S*martphone (2020)* M. Alvian H Nasution dan Agung Tri Laksono	S*martphone*		*Dynamic Analisis Malware*	*Remote Access Trojan*	Serangan RAT dapat terjadi dan menyerang *system* dari sebuah android di mana pada serangan RAT *attacker* dapat melakukan akses terhadap *system* secara penuh.	Metode yang digunakan berbeda dengan penelitian ini, hasil yang didapatkan berbeda dengan penelitian ini.
6.	*Analysis of Remote Access Trojan Attack using Android Debug Bridge* (2021) Deco Aprilliansyah	Android	*Ghost Framework*	*Post-Exploitation*	*Android Debug Bridge (ADB)*	Android versi 9 dapat diakses dari jarak jauh dengan memasukkan eksploit melalui ADB.	Jenis tools yang digunakan berbeda dengan penelitian ini, metode

No	Penelitian	Studi Kasus	Tools	Metode	Vunarebility Malware	Hasil Penelitian	Perbedaan
							dan jenis malware yang digunakan berbeda dengan penelitian ini.
7.	Evaluasi Keamanan Perangkat Seluler menggunakan *Metode Reverse TCP* (2022) Imam Riadi	Android versi 11 dan 12	*Metas-ploit frame-work*	*Android Security Lab-ware*	*Remote Access Trojan*	Sistem Android menggunakan *framework Metasploit* untuk memuat, mengeksploitasi, dan mengontrol perangkat korban sebagai trojan akses jarak jauh menggunakan android 11 dan versi 12. Sebelumnya penelitian sistem android menggunakan *metode exploit* dilakukan oleh android.	Fokus system operasi yang digunakan berbeda dengan penelitian ini, metode yang di gunakan juga berbeda dengan penelitian ini.

2.2 Landasan Teori

2.2.1 Android

Android adalah perangkat bergerak pada sistem operasi untuk telepon seluler berbasis *linux* yang di dalamnya terdapat sistem operasi, *middleware* dan aplikasi serta memiliki sifat *open source*[16]. Meskipun awal mulanya android hanya untuk telepon seluler namun sekarang juga dikembangkan untuk perangkat keras lain di antaranya kamera digital, jam

tangan, perangkat navigasi, televisi, dan kacamata pintar. Android adalah salah satu OS (*Operating System*) *mobile* yang tumbuh dan dikembangkan oleh Google Inc. Windows mobile, iphone operating system, Symbian merupakan contoh lain dari OS yang mengedepankan aplikasi inti yang dibangun sendiri dan mengabaikan potensi dari aplikasi pihak ketiga[17]. Dikarenakan adanya keterbatasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk platform developer. Android dipuji sebagai "platform mobile pertama yang lengkap, terbuka, dan bebas"[18]. Menurut pemikiran tersebut, bisa disimpulkan bahwa android merupakan sistem operasi berbasis linux yang populer di tengah OS lainnya.

2.2.2 Keamanan Sistem Android

Keamanan fisik atau *physical security* dikenal sebagai keamanan tingkat awal yang merupakan definisi dari keamanan level nol. Keamanan dari *smartphone* sangat bergantung dengan keamanan fisik. Keamanan level 1 mencakup keamanan *database*, data, dan keamanan perangkat. Saat membuat *database*, dapat dilihat apakah aplikasi yang digunakan telah diakui keamanannya. Desain *database* penting untuk diperhatikan dikarenakan wajib memikirkan kemungkinan keamanan dari database. *Device security* yaitu alat keamanan dari *database* tersebut. Keamanan level 2 adalah *patch* keamanan dari segi keamanan jaringan dan keseluruhan sistem. Patch level ini terdapat beberapa tambahan dari segi keamanan pada level selanjutnya dan memiliki ketergantungan, keamanan *patch* sendiri biasanya diberikan oleh para pengembang selama 3 tahun sejak *smartphone* tersebut dirilis. Kondisi ini tentu menjadi sebuah ancaman jika suatu saat terjadi peretasan pada *smartphone*, maka akan berimbas kepada patch keamanan level selanjutnya juga dengan dukungan hanya selama tiga tahun maka keamanan level 3 dan seterusnya akan mengikuti aturan patch keamanan level sebelumnya. Hal ini berarti jika batas support telah berakhir

maka keamanan level 3 dan 4 tidak akan terpenuhi mengingat syarat terpenuhinya keamanan level 3 dan 4 adalah patch keamanan level 2. Selanjutnya jika patch keamanan hanya diberikan selama 3 tahun maka setelah usai support patch tersebut secara otomatis information security yang meliputi data-data maupun file yang ada pada smartphone akan terancam. Keamanan level 3 adalah *information security*. Informasi tersebut di antaranya kata sandi yang dikirimkan ke teman atau file penting, karena dikhawatirkan adanya orang yang tidak sah. Sedangkan definisi dari keamanan level 4 merupakan keseluruhanan keamanan yang ada dari level 1 hingga level 3 saling terkait satu sama lain, jika satu dari keamanan tersebut tidak terpenuhi maka keamanan level juga tidak terpenuhi, begitu pula sebaliknya[19].

2.2.3 Baidu

Baidu browser merupakan aplikasi peramban yang berasal dari negara China diterbitkan oleh raksasa pencarian online Tiongkok, Baidu browser adalah browser web berdasarkan proyek open source chromium google. Dilengkapi dengan antar muka klasik, browser ini diperkaya dengan berbagai modul seperti klien pengunduhan BitTorrent yang terintegrasi. Sebelumnya disebut Spark Browser, perangkat lunak ini bekerja seperti browser standar yang telah ditambahkan berbagai ekstensi. Pada penelitian ini, baidu browser akan disuntik/disisipkan sebuah virus *metasploit* berjenis RAT[20].

2.2.4 Malware

Malicious software atau, lebih dikenal sebagai "*malware*", adalah kode yang sengaja diciptakan untuk merusak serta membahayakan sistem operasi atau data komputer[21]. Definisi tersebut dikemukakan oleh beberapa ahli seperti Siddiqui. *Malware* sendiri diciptakan untuk menyebabkan kerusakan dan kehancuran pada sistem serta jaringan komputer tanpa adanya izin dari pemiliknya. Pada umumnya kerusakan dan kehancuran sistem jaringan komputer ini menimbulkan kerugian bagi orang

lain dikarenakan adanya pengambilan data, penyalahgunaan informasi pribadi, serta pencurian identitas secara sepihak yang kemudian dimanfaatkan untuk tujuan kejahatan. Penyebaran *malware* ini biasanya melalui aplikasi, pelaku berpura-berpura menjadi oknum kurir pengiriman yang akan memberikan informasi kedatangan paket. Namun, untuk dapat menerima paket tersebut, calon korban diwajibkan mengunduh dan membuka lampiran file, maka pada saat file tersebut terunduh malware secara otomatis akan terunduh juga. Namun, tidak hanya berpura-pura menjadi kurir, pelaku biasanya juga melakukan penyebaran melalui aplikasi ilegal yang ada di internet. Adapun jenis-jenis malware terbagi menjadi beberapa di antaranya keylogger, worm, trojan, ransomeware, backdoor, rootkit, spyware, dan lain sebagainya[22].

2.2.5 TheFatRAT

TheFatRat merupakan sebuah alat untuk melakukan *exploiting* yang memungkinkan pembuatan *malware* dengan *payload* terkenal. *TheFatRat* dapat membuat sebuah *backdoor* untuk sistem operasi windows, linux, mac dan android (Kunwar, Sharma, & Kumar, 2009). *TheFatRat* mempunyai beberapa fungsi di antaranya membuat *backdoor* menggunakan *msfvenom*, *backdoor apk* di mana menggunakan file apk original seperti file apk instalasi Instagram, line, facebook, dan lain-lain. File original tersebut akan di decompile dan di inject backdoor sehingga korban tidak sadar bahwa aplikasi yang diinstal pada smartphone ternyata ada backdoor.

2.2.6 Remote Access Trojan (RAT)

Remote Access Trojan (RAT) adalah aplikasi malware dengan pintu belakang (*backdoor*) untuk kontrol administratif atas komputer target. Pintu belakang yang dimaksud adalah berupa port. *Backdoor* merupakan metode yang digunakan untuk melewati autetifikasi normal (*login*) dan berusaha tidak terdeteksi[23]. Malware merupakan salah satu bentuk kejahatan pada sebuah jaringan komputer. Salah satu jenis virus *trojan horse* adalah *Backdoor.* yang di mana virus ini dapat berkembang di dalam perangkat

yang terinfeksi, yang memungkinkan pencuri untuk memasuki sistem tanpa diketahui pemiliknya. Malware yang diinstal di backdoor biasanya disebut sebagai *remote access trojan*. (RAT)[13].

Remote Access Trojans (RAT) adalah kelas pintu belakang yang digunakan untuk memungkinkan remote control melalui mesin yang dikompromikan dan memberikan fungsi tampaknya bermanfaat bagi pengguna dan, pada saat yang sama, membuka port jaringan pada komputer korban. Setelah RAT dimulai, RAT berperilaku sebagai file yang dapat dieksekusi, berinteraksi dengan kunci registry tertentu yang bertanggung jawab untuk memulai proses dan terkadang membuat layanan sistemnya sendiri. Tidak seperti pintu belakang biasa, RAT menghubungkan diri ke sistem operasi korban dan selalu dikemas dengan dua file yang terdiri atas file klien dan file server. Server diinstal di mesin yang terinfeksi, dan klien digunakan oleh penyusup untuk mengontrol sistem yang dikendalikannya[23].

RAT digunakan untuk melakukan akses jarak jauh karena memiliki kemampuan untuk mengendalikan perangkat yang telah terinfeksi oleh malware yang telah di sisipkan backdoor, memungkinkan pencuri untuk melakukan autentikasi terhadap perangkat yang digunakan oleh korban[13]. RAT telah menjadi penting untuk semua jenis aktivitas penjahat dunia maya, digunakan oleh penjahat dunia maya, peretas negara, serta penguntit. Pasar telah matang. RAT telah berjalan jauh sejak NokNok menggunakan komputer windows dan meluncurkan babak baru dalam sejarah keamanan komputer ini[24].

2.2.7 Metasploit

Framework Metasploit merupakan sebuah *platform* untuk pengujian dan pengembangan. exploitasi yang menyediakan akses ke kode exploit untuk berbagai aplikasi, sistem operasi, dan platform yang bersifat open source. Sebagai bahasa berorientasi objek, Ruby adalah bahasa scripting yang digunakan. Selain itu, metasploit dianggap berfungsi pada sebagian

besar platform. variasi unix dan window[10]. Fasilitas yang diberikan oleh *framework metasploit* memudahkan untuk melakukan serangan yang bersifat legal maupun ilegal terhadap sebuah sistem komputer. Dengan demikian banyak profesional *penetration tester* ataupun *ethical hacker* yang memanfaatkan *metasploit* untuk melakukan pekerjaannya, namun di sisi lain tentunya adapula pihak-pihak yang tidak bertanggung jawab memanfaatkan *metasploit* untuk melancarkan upaya yang bersifat ilegal atau mengarah ke tindak kriminal berupa suatu tindakan peretasan memanfaatkan *exploit* tertentu menggunakan framework metasploit pada komputer(D.C. PRAKOSO - Google Scholar, n.d.).

Bagi pengguna baru *metasploit framework* sulit untuk digunakan karena tidak tersedianya *graphic user-interface* (GUI), meskipun menghasilkan cara kerja yang luar biasa. Oleh karena itu, supaya dapat menggunakan *metasploit* secara efektif diperlukan pemahaman terlebih dahulu mengenai *sintaks* dan perintah. Dalam *metasploit*, untuk sebagian besar serangan perlu mengikuti langkah-langkah dasar yang disebutkan di bawah ini.

1. Memilih dan mengkonfigurasi (kode yang memasuki sistem target dengan mengambil keuntungan dari salah satu bug tersebut)
2. Memilih dan mengkonfigurasi *payload* (kode yang akan dieksekusi pada sistem target jika berhasil masuk)
3. Memilih teknik pengkodean sehingga sistem intrusi pencegahan mengabaikan *payload*
4. Mengeksekusi *exploit*[10]

Investigator digital forensik menghadapi tantangan dalam menemukan bukti digital pada komputer korban dalam kasus kriminal yang menggunakan *framework metasploit*. Serangan exploit pada komputer korban harus dianalisis untuk menemukan barang bukti digitalnya. Komputer korban memiliki RAM yang mencatat semua prosesnya, jadi serangan *exploit* yang dilancarkan menggunakan metasploit dapat menemukan barang bukti di dalam RAM dikarenakan RAM komputer

memiliki sifat *volatile*. Ini berarti data yang tersimpan di dalamnya akan hilang jika sistem komputer mati. Oleh karena itu, forensik digital harus dilaksanakan dengan cara *live forensics*, yang berarti komputer harus dinyalakan. Akibatnya, perlu dilakukan penelitian mengenai cara melakukan investigasi RAM komputer menggunakan sistem yang menyala pada Windows 8 dan Windows 10, serta cara melakukan simulasi serangan menggunakan metasploit untuk mengetahui karakteristik artefak barang bukti digital yang ditemukan pada sistem operasi tersebut setelah melakukan simulasi serangan menggunakan metasploit(D.C. PRAKOSO - Google Scholar, n.d.)

2.2.8 Analisis Statik

Analisis statis merupakan cara kerja untuk dapat melaksanakan analisis perangkat lunak tanpa mengeksekusinya. Selama analisis statis aplikasi dipecah dengan menggunakan alat teknik rekayasa balik, sehingga membangun kembali kode sumber dan algoritma yang telah dibuat oleh aplikasi. Analisis statis dapat dilakukan melalui program analisis, *debugger* dan *disassembler*[26]. Berbagai teknik analisis statis adalah sebagai berikut :

1. Teknik Pendeteksian *Heuristic*

   Pada umumnya teknik tersebut disebut dengan teknik proaktif yang hampir sama dengan teknik berdasarkan *signature* tertentu dalam kode, detektor *malware* sekarang mencari perintah atau intruksi yang tidak ada dalam program aplikasi. Hasilnya, varian *malware* baru yang belum ditemukan lebih mudah ditemukan di sini[26]. Teknik analisis *heuristik* yang berbeda adalah sebagai berikut:

   a. *File Based Heuristic Analysis*

      Analisis *heuristik* berdasarkan *file* disebut dengan analisis *file*. Teknik tersebut diawali dengan menganalisis *file* secara menyeluruh mulai dari isi, tujuan, pengerjaan file, apabila

terdapat perintah untuk menghapus atau merusak *file* lain, file tersebut dianggap berbahaya.

b. *Based Heuristic Analysis*

Analisis dilakukan dengan melakukan ekstrasi aturan yang mengidentifikasikan aplikasi. Aturan-aturan ini kemudian disesuaikan dengan aturan sebelumnya, apabila tidak terdapat kecocokan antar aturan, maka aplikasi tersebut tedapat *malware.*

c. *Generic Signature Analysis*

Varian *malware* berarti, *malware* itu berbeda dalam perilakunya tetapi memiliki keluarga yang sama seperti "kembar identik". Teknik ini menggunakan definisi antivirus yang ditetapkan sebelumnya, untuk menemukan varian baru *malware*[26].

2.2.9 Reverse Engineering

Reverse engineering terdiri atas dua kata dasar yaitu *reverse* dan *engineering*. *Reverse* merupakan kebalikan atau kebalikan dari kondisi sebelumnya atau normal. Apabila dilihat maka *reverse engineering* ialah metode untuk menemukan dan mengidentifikasi sistem, fungsi, dan operasi yang bekerja pada desain, komponen, atau objek melalui proses analisis menyeluruh pada setiap komponen struktur dari desain atau objek yang hendak diteliti[27]. Penggunaan metode *reverse engineering* dalam analisis *malware* sangat diperlukan karena dapat mengekstraksi data yang memuat informasi di dalam *malware*[11]. Selain itu, penggunaan metode ini juga dapat untuk mengidentifikasi ancaman vektor serangan, cara penularan dan penyebaran *malware*, upaya pencegahan serta penyelesaian yang efektif terhadap ancaman tersebut. Metode rekayasa balik atau *reverse engineering* memberikan manfaat bagi korban untuk dapat menghindari deteksi dan penyebaran *malware* karena dapat meniru strategi yang digunakan oleh penyerang[27]. Dengan demikian, hal ini memungkinkan untuk mengambil tindakan pencegahan yang sesuai guna mendeteksi dan mencegah kemungkinan malware yang dapat terjadi di masa depan. Dengan kata lain,

rekayasa balik atau reverse engineering bukan hanya sekadar alat untuk menganalisis serangan yang sudah terjadi, tetapi juga merupakan langkah proaktif dalam memperkuat pertahanan keamanan cyber dengan mengantisipasi potensi ancaman yang akan datang.

2.2.10 JADX

JADX merupakan sebuah *tools* yang digunakan untuk melakukan penelitian secara mendalam dari sebuah file aplikasi android yang telah dilakukan injeksi *metasploit* dan pada aplikasi yang belum dilakukan injeksi *metasploit* dengan tujuan untuk mengetahui perbedaan kode dan perbedaan *anomali* yang ada pada aplikasi original, yang kemudian hasil dari analisis yang didapatkan dari melakukan *decompile* file .apk akan dibandingkan temuannya dengan hasil yang di dapatkan dari scaning secara otomatis melalui *MobSF, JADX* memiliki antarmuka berbasis Graphical User Interfaces yang sangat interaktif, alat ini memberikan sebuah output dalam bahasa java dari ekstraksi file berekstensi.apk[28]. Pada penelitian ini *JADX* dipilih sebagai tools analisis manual karena penggunaannya yang mudah serta dapat mencakup file - file mencurigakan yang tidak semua dapat terbaca dalam *MobSF*.

2.2.11 Mobile Security Framework (MobSF)

*Mobile Security Framework (*MobSF) adalah *framework* pengujian otomatis yang memiliki *open*-source. MobSF sendiri dapat melakukan analisis statis dan dinamis serta menampilkan hasil yang berupa laporan. *Mobile Security Framework* (MobSF) adalah *framework* yang digunakan untuk pengujian *exploitasi* terhadap aplikasi seluler (android/iOS/windows) otomatis yang mampu melakukan analisis statis, dinamis, dan *malware.* MobSF mendukung kedua binari (*Android Package Kit* (APK), *iPhone Application* (IPA) & APPX yang merupakan *format file windows store*) dan kode sumber zip. MobSF dapat melakukan pengujian aplikasi dinamis saat *runtime* untuk aplikasi android dan memiliki kemampuan *fuzzing API* Web yang didukung oleh *CapFuzz,* pemindai keamanan khusus Web

API.Fuzzing merupakan suatu metode mencari kesalahan piranti lunak dengan menyediakan *input* yang tidak diduga lalu mengamati hasilnya[29].

MobSF dirancang untuk membuat integrasi *Continuous Integration/ Continuous Delivery* (CI/CD) atau *Development, Security*, *and Operations* (Dev-SecOps) secara bagus[30]. Metode pengiriman aplikasi ke pelanggan yang dilakukan teratur dengan memasukkan otomatisasi ke fase pengembangan aplikasi disebut dengan *CI/CD. DevSecOps* merupakan Gambaran kerja kerja sama yang memperluas pengaruh *Development and Operations* (*DevOps*) dengan memasukkan praktik keamanan ke proses pengembangan dan pengiriman perangkat lunak[31]. *Mobile Security Framework* merupakan kerangka kolaborasi yang dikembangkan dengan *Phyton* dan dapat melaksanakan analisis statis serta dinamis dari sebuah aplikasi.. Untuk menghasilkan laporan MobSF menggunakan *html.* Ini menjalankan server lokal melalui baris perintah di komputer *host*[32]. Semua alat analisis menggunakan *DroidMon-Dalvik Monitoring Framework* dan *Xposed Module Repository* untuk melaksanakan analisis. *Xposed* merupakan kerangka kerja modul yang memungkinkan terjadinya perubahan perilaku sistem dan aplikasi tanpa menyentuh APK. Hal ini berarti modul mampu bekerja dengan versi yang tidak sama dan bahkan ROM tanpa perubahan apa pun (berlaku apabila kode asli tersebut tidak mengalami perubahan yang signifikan)[33].

Backdoor adalah program yang dibuat agar dapat melewati autentifikasi normal (*login)* atau disebut dengan mengakses dari pintu belakang secara tidak sah, ketika pintu belakang sudah masuk ke dalam sistem maka dapat dengan mudah untuk mengambil alih komputer.