BAB II

DASAR TEORI

2.1 Kajian Pustaka

Terdapat penelitian yang melakukan analisis aplikasi Simpel Desa dengan menggunakan *Mobile Security Framework Static Analysis* untuk mengamankan *credential asset* seperti KTP. Pada penelitian didapatkan *risk score* pada aplikasi 6.2 dengan *Vulnerability* yang ditemukan yaitu *Weak Cypto, Dangerous Permission, Network Security, Hardcode Secret,* dan SSL *Pinning* [7].

Terdapat penelitian yang melakukan analisis aplikasi *game* dengan menggunakan *Mobile Security Framework Static Analysis* dan melakukan kombinasi pemanfaatan *background process* Android. Pada penelitian ini ditemukan *Vulnerability* yaitu *Dangerous Permission, Weak Cypto, Root Detection,* SSL *Bypass,* dan *Domain Malware Check* [8].

Terdapat penelitian yang melakukan analisis aplikasi video *steaming* menggunakan *Mobile Security Framework Static Analysis*. Pada penelitian ini ditemukan *Vulnerability* yaitu *Dangerous Permission, Weak Cypto, Root Detection,* dan SSL *Bypass* [9] .

Terdapat penelitian yang melakukan analisis aplikasi .apk menggunakan *Mobile Security Framework Static Analysis*. Pada penelitian ini ditemukan *Vulnerability* yaitu *Dangerous Permission, Weak Cypto, Root Detection,* dan SSL *Bypass* [10].

Terdapat penelitian yang melakukan analisis aplikasi XYZ menggunakan OWASP *Mobile Security Testing Guide*. Pada penelitian ini ditemukan *Vulnerability* yaitu *Local Storage for Sensitive Data, Local Storage for Input Validation, Keyboard Cache is Disabled, Sensitive Stored Data via* IPC [11].

Terdapat penelitian yang melakukan analisis aplikasi Android menggunakan *Mobile Security Framework Static Analysis*. Pada penelitian ini menggunakan *dynamic analysis* dan ditemukan *Vulnerability* yaitu *Sensitive Information at Local Storage, Improper Certificate Validation,* IDOR, *Weak Encoding for Password* [12].

Pada penelitian ini akan menganalisis *risk rating, vulnerability, threat actor, threat intelligence platform, TLS/SSL security test, speedtest* dan *command and*

control attack dengan atau tanpa *Virtual Private Network.* Skenario yang akan digunakan dalam penelitian ini yaitu skenario pertama akan melakukan analisis menggunakan metode SAST dan DAST pada aplikasi hasil buatan penulis menggunakan Metasploit yaitu *Creator*.apk dengan tujuan untuk dapat melakukan *C&C attack simulation* dengan atau tanpa *virtual private network*. Skenario kedua akan melakukan analisis menggunakan metode SAST dan DAST pada aplikasi Undangan Pernikahan.apk dan Cek Resi J&T.apk. Aplikasi ini terinfeksi *malware* dan beredar sangat *massive* di kalangan Masyarakat yang dapat berpotensi terjadinya kebocoran data. Berikut merupakan indikator perbandingan antara penelitian ini dengan penelitian sebelumnya dan kajian pustaka yang akan dijelaskan pada tabel 2.1 dan tabel 2.2.

Tabel 2.1 Indikator Perbandingan Penelitian

Indicator	***Previous Research***	***My Research***
Methodology	SAST	SAST, DAST
Application Quantity	1	3
Framework	MobSF	MobSF
Risk Rating Analysis	*Yes*	*Yes*
Vulnerability Analysis	*Yes*	*Yes*
Threat Actor Analysis	*No*	*Yes*
TIP Analysis	*No*	*Yes*
TLS/SSL Security Test	*No*	*Yes*
Speed test Analysis	*No*	*Yes*
C&C Attack Analysis	*No*	*Yes*

Tabel 2.2 Kajian Pustaka

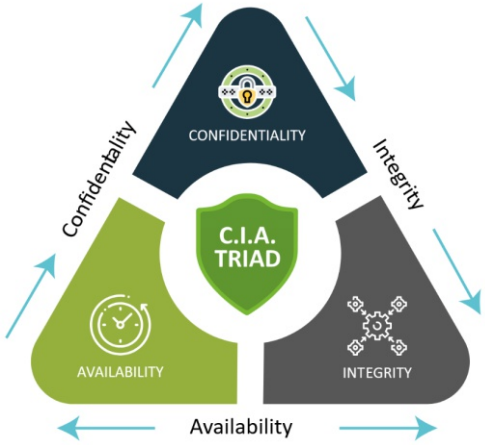
Year	***Author***	***Objective***	***Framework***	***Vulnerability***
2022	Khairunnisak Nur Isnaini, Didit Suhartono	Pada penelitian ini menggunakan metode *Static Analysis* untuk menganalisa aplikasi Simpel Desa dengan tujuan untuk mengamankan *credential asset* seperti KTP.	*MobSF*	*Weak Crypto, Dangerous Permission, Network Security, Hardcode Secret,* dan SSL *Pinning*
2022	Aris Rafael Tambunan, Trihastuti Yuniati, Yoso Adi Setyoko	Pada penelitian ini menggunakan metode *Static Analysis* untuk membaca informasi Malware dengan mengkombinasikan *background process* Android pada aplikasi *game*.	*MobSF*	*Dangerous Permission, Weak Crypto, Root Detection,* SSL *Bypass,* dan *Domain Malware Check*
2021	Fitri Nurindahsari, Bita Parga Zen	Pada penelitian ini menggunakan metode *Static Analysis* pada aplikasi *video streaming* Android untuk mengetahui aktivitas *Malware*.	*MobSF*	*Dangerous Permission, Weak Crypto, Root Detection,* dan SSL *Bypass*

Year	***Author***	***Objective***	***Framework***	***Vulnerability***
2022	Imam Himawan, Kevin Septianzah, Irawan Setiadi	Pada penelitian ini menggunakan metode *Static Analysis* pada aplikasi Sistem Pakar untuk mengetahui tingkat kerentanan aplikasi.	*MobSF*	*Dangerous Permission, Weak Crypto, Root Detection,* dan SSL *Bypass*
2021	Aldino Dika Pratama, Amiruddin	Pada penelitian ini menggunakan OWASP *Mobile Security Testing Guide* untuk mengetahui kerentanan pada aplikasi ABC.	*OWASP Mobile Security Testing Guide*	*Local Storage for Sensitive Data, Local Storage for Input Validation, Keyboard Cache is Disabled, Sensitive Stored Data via* IPC
2020	Fauzan Awanda Alviansyah	Pada penelitian ini menggunakan metode Dynamic Analysis untuk mengetahui kerentanan pada aplikasi Android.	*MobSF*	*Improper Certificate Validation, Unrestricted File Upload,* XSS, SQL *Injection, Weak Encoding,* IDOR

Year	*Author*	*Objective*	*Framework*	*Vulnerability*
2024	Yunanda Pratama	Pada penelitian ini menggunakan metode *Static and Dynamic Analysis* untuk mengetahui pengaruh *Virtual Private Network* dan kerentanan pada aplikasi Android terinfeksi Malware.	*MobSF*	*Use of a Broken or Risky Cryptographic Insufficient Cryptography Use Insufficiently Random Values Insufficient Cryptography Weakness in Mobile Application Improper Platform Usage*

2.2 Keamanan Informasi

Keamanan informasi adalah upaya melindungi informasi, baik data maupun infrastruktur, dari tindakan yang berbahaya atau tidak sah. Fokus dari keamanan informasi berupa upaya dalam mengamankan semua data dari tindakan yang berbahaya atau tidak sah. Pada tingkat individu, keamanan informasi berupa perlindungan data pribadi, seperti identitas, aktivitas, dan perangkat. Pada tingkat organisasi, keamanan informasi berupa perlindungan informasi organisasi, reputasi, dan pengguna [13]. Aspek-aspek keamanan yang berkaitan dengan keamanan informasi yaitu: *Confidentiality, Integrity*, dan *Availability* (CIA) (ISO/IEC 27001, 2018) [14]. Adapun untuk aspek keamanan informasi terdapat pada gambar 2.1.



Gambar 2.1 *Information Security* (CIA)

2.2.1 *Confidentiality*

Keamanan informasi dari aspek kerahasiaan *(confidentiality)* adalah memastikan informasi atau data yang dimiliki sistem atau pemilik informasi tersebut bersifat rahasia dan hanya bisa diakses oleh pemilik atau orang yang berhak.

2.2.2 *Integrity*

Keamanan informasi dari aspek integritas *(Integrity)* adalah menjamin informasi atau data yang tersimpan dalam sistem dapat terjaga tanpa adanya perubahan atau modifikasi data serta memastikan informasi atau data tetap utuh dan benar adanya.

2.2.3 *Availability*

Keamanan informasi dari aspek ketersediaan *(Availability)* adalah memastikan sistem atau infrastruktur dapat menyediakan informasi atau data kepada pengguna saat diakses maupun digunakan. Ketersediaan informasi atau data tersebut harus selalu diperhatikan agar mendukung kinerja dari sistem dan proses bisnis dapat berjalan dengan baik.

2.3 *Static Application Security Testing (SAST)*

Static Application Security Testing (SAST) adalah metode untuk menemukan kerentanan keamanan dalam aplikasi yang dapat dieksploitasi oleh *malware*. SAST dilakukan di awal proses *Sytem Development Life Cycle (SDLC)*. *SDLC* adalah proses yang digunakan untuk mengembangkan perangkat lunak dari awal

hingga selesai. Dalam *SAST*, perangkat lunak tidak perlu dijalankan, tetapi hanya kode sumbernya. Hal ini memungkinkan pengembang untuk mengidentifikasi kerentanan pada tahap awal dan mempercepat proses perbaikan. *SAST* memberikan umpan balik kepada pengembang secara *real time* saat kode sumber masih dalam pengembangan. Pengembang kemudian dapat memperbaiki masalah tanpa harus melanjutkan ke fase *SDLC* berikutnya. Hal ini membantu mencegah masalah keamanan terkait penggunaan perangkat lunak. *SAST* juga memudahkan proses perbaikan kode karena *SAST* menunjukkan di baris kode mana kerentanan atau masalah terjadi [15]. Adapun untuk SAST *requirement* terdapat pada tabel 2.3.

Tabel 2.3 SAST *Requirement*

Static Application Security Testing (SAST)
Whitebox Testing
Requires Source Code
Earlier Detection
Doesn't find environment issues

2.4 *Dynamic Application Security Testing (DAST)*

Dynamic Application Security Testing (DAST) adalah proses pengujian aplikasi saat aplikasi sedang berjalan. Tujuan utama *DAST* adalah untuk mengidentifikasi dan mencari kerentanan keamanan dalam aplikasi Android, seperti *root bypass* yang memungkinkan penyerang untuk masuk tanpa otentikasi. *DAST* dapat diterapkan setelah aplikasi memasuki fase produksi atau *runtime*, setelah fase pengembangan awal. *DAST* menggunakan pendekatan pengujian *black box*, yang berarti bahwa pengujian dilakukan tanpa akses ke kode sumber atau informasi internal lainnya. Dalam pengujian *DAST*, hanya APK aplikasi yang diperlukan. *DAST* adalah bagian penting dari siklus *Secure Software Development Life Cycle (SSDLC)* [15] . Adapun untuk DAST *requirement* terdapat pada tabel 2.4.

Tabel 2.4 DAST *Requirement*

Dynamic Application Security Testing (DAST)
Black box Testing

Dynamic Application Security Testing (DAST)
Requires Web Application in Staging or Production
Later Detection
Finds Environment Issues

2.5 *Virus Total*

Virus Total adalah *Threat Intelligence Platform* yang digunakan untuk menganalisis *file,* URL, *domain, hash* dan alamat IP yang mencurigakan untuk mengetahui dan mendeteksi jenis *malware* atau ancaman apa saja yang terdapat pada *file,* URL, *domain,* hash dan alamat IP tersebut. *Virus Total* menyediakan berbagai informasi yang dikumpulkan dari berbagai *community* untuk dapat saling berbagi pengetahuan terkait jenis serangan yang ada pada suatu aspek yang sedang dilakukan analisis. API *Virus Total* memungkinkan untuk diintegrasikan pada *tools* keamanan untuk membuat sistem otomatisasi seperti pengecekan *bad* IP *reputation*.

2.6 *Metasploit*

Metasploit adalah *platform* perangkat lunak keamanan yang sering digunakan untuk menguji keamanan sistem dengan mengeksploitasi kerentanan *software*. *Metasploit Project*, yang dikembangkan oleh Rapid7, menyediakan sistem yang berisi informasi tentang kerentanan keamanan komputer dan kode sumber yang memungkinkan jaringan untuk mengidentifikasi risiko keamanannya sendiri. *Metasploit* juga dapat digunakan sebagai alat serangan untuk mencari dan mengeksploitasi kerentanan sistem komputer, sehingga dapat memperoleh akses ke data dan fitur sistem. *Metasploit Framework*, yang merupakan produk paling terkenal dari proyek ini, dapat digunakan untuk mengembangkan, menguji, dan menjalankan eksploitasi.

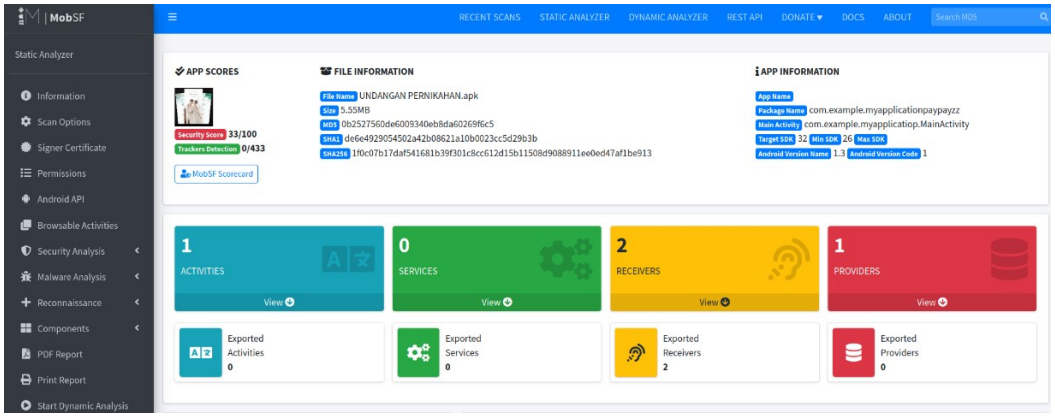
2.7 *Mobile Security Framework*

Mobile Security Framework (MobSF) adalah sebuah *framework* pengujian keamanan *mobile application* bersifat *open source*. MobSF bertujuan sebagai *vulnerability assessment* atau mencari informasi celah keamanan berdasarkan metode pengujian yang dilakukan. Metode pengujian yang dapat dilakukan oleh

MobSF adalah *Static Application Security Testing* (SAST) dan *Dynamic Application Security Testing* (DAST) [12]. MobSF akan menilai *severity level* pada aplikasi yang dilakukan analisis dengan beberapa informasi yang disajikan diantaranya: biru menandakan *activity* pada *code* yang *running* pada aplikasi, hijau menandakan *services* yang *running* pada saat aplikasi dijalankan, *orange* menandakan *receivers* atau kemampuan aplikasi mengirim dan menerima SMS, serta warna merah menandakan *inialization provider* atau inisialisasi otomatis dalam suatu proses [12]. Adapun untuk *risk calculation* yang ada pada *Mobile Security Framework* terdapat pada tabel 2.5 dan untuk tampilan *Mobile Security Framework* terdapat pada gambar 2.2.

Tabel 2.5 *Risk Calculation* MobSF

Security Score	***Risk***
0-15	*Critical*
16-40	*High*
41-70	*Medium*
71-100	*Low*



Gambar 2.2 *Mobile Security Framework*

2.8 *Genymotion*

Genymotion adalah salah satu jenis *emulator* Android yang sering digunakan oleh para pengembang Android. *Genymotion* memiliki antarmuka yang menarik dan *user-friendly*, sehingga memudahkan pengguna untuk menjalankan aplikasi Android. Pada *Genymotion*, pengguna dapat memilih jenis *smartphone* dan versi Android yang ingin dijalankan sesuai dengan pilihan yang telah disediakan oleh

Genymotion. *Genymotion* merupakan emulator yang lebih *flexible* untuk diintegrasikan dengan simulasi serangan dibandingkan dengan *Android Studio* [12].

2.9 *Docker Container*

Docker Container adalah paket perangkat lunak yang berisi semua dependensi yang diperlukan untuk menjalankan aplikasi tertentu. *Image Docker* adalah komponen yang berisi konfigurasi dan instruksi untuk memulai atau menghentikan *Container*. *Container* mencegah masalah kompatibilitas yang dapat terjadi saat menjalankan aplikasi di berbagai sistem yang berbeda. *Container* juga membuat perangkat lunak berjalan dengan cara yang sama di berbagai lingkungan. Setiap image yang dijalankan oleh pengguna akan secara otomatis membuat *Container* baru agar perangkat lunak tidak mengalami gangguan. *Docker* API atau *Command Line Interface* (CLI) membantu mengelola *Container* dengan lebih mudah. *Docker compose tool* dapat digunakan untuk mengontrol beberapa *Container*. Arsitektur *Docker* terdiri dari empat komponen utama, yaitu:

1. *Client Docker*: Komponen penting yang fungsinya untuk membuat, mengelola, dan menjalankan aplikasi pada container.
2. *Server Docker*: Sering disebut juga sebagai *Daemon Docker* ini menunggu permintaan REST API yang dibuat oleh client Docker juga mengelola image dan container.
3. *Image Docker*: Komponen yang menginstruksikan server *Docker* mengenai persyaratan tentang cara container *Docker* dibuat.
4. *Registry Docker*: Aplikasi sisi server *open-source* ini berfungsi untuk men-*ghosting* dan mendistribusikan *image Docker*.

Fungsi *Docker* adalah mencoba *software* baru, mempelajari CLI dan mengurangi risiko insiden yang mungkin terjadi [16].

2.10 JADX

JADX merupakan *decompiler* untuk bahasa pemrograman Java yang dapat digunakan untuk melakukan *reverse engineering* pada aplikasi Android. *Reverse engineering* adalah proses membongkar suatu aplikasi untuk mempelajari cara kerja *system* tersebut. JADX dapat digunakan untuk mendapatkan *source code* dari

aplikasi Android yang sudah terkompilasi. Berikut adalah beberapa manfaat menggunakan JADX untuk melakukan *reverse engineering* pada aplikasi Android:

1. Dapat mempelajari cara kerja aplikasi Android secara lebih *detail*.
2. Dapat menemukan *bug* atau kerentanan keamanan pada aplikasi Android.
3. Dapat membuat aplikasi Android yang serupa dengan aplikasi yang sudah ada.

2.11 *VirtualBox*

VirtualBox adalah perangkat virtualisasi yang memungkinkan pengguna untuk menjalankan sistem operasi tambahan di dalam sistem operasi utama. VirtualBox berfungsi untuk menvirtualisasi sistem operasi. *VirtualBox* juga dapat digunakan untuk membuat virtualisasi jaringan komputer sederhana. *VirtualBox* ditargetkan untuk penggunaan di *desktop*.

2.12 Ngrok

Ngrok adalah suatu program yang membuat alamat internet untuk aplikasi yang berjalan di komputer *local* atau *private*. Alamat internet ini akan diakses dari internet, sehingga aplikasi lokal bisa diakses dari mana saja.

Ngrok bisa digunakan untuk berbagai keperluan, misalnya:

1. Demo aplikasi tanpa harus *deploy* ke *server hosting*. Dengan Ngrok, aplikasi lokal bisa diakses dari internet, sehingga bisa didemokan ke klien tanpa harus *deploy* ke *server hosting*;
2. Mempermudah pengujian aplikasi di perangkat *mobile*. Dengan Ngrok, aplikasi lokal bisa diakses dari perangkat *mobile*, sehingga bisa diuji tanpa harus mengubah konfigurasi jaringan.

2.13 Sistem Operasi Android

Android adalah sistem operasi berbasis Linux yang dikembangkan oleh Google untuk perangkat mobile, seperti smartphone. Android pertama kali dirilis pada tahun 2005, dan kemudian diakuisisi oleh Google. Saat ini, Android telah mencapai versi 14. Android merupakan perangkat lunak sumber terbuka, sehingga dapat dikembangkan oleh siapa saja. Aplikasi Android didistribusikan dan diinstal dalam format berkas *Application Package* (APK). APK adalah berkas yang berisi aplikasi Android. Struktur berkas APK mirip dengan berkas .EXE yang digunakan

di Windows untuk menjalankan aplikasi. Seperti halnya berkas zip atau rar, APK dapat diekstraksi untuk melihat struktur aplikasi Android [12].

2.14 *Virtual Private Network*

Virtual Private Network (VPN) adalah perangkat lunak yang memungkinkan para pengguna untuk tersambung ke layanan internet secara pribadi. VPN memberikan akses secara aman melalui koneksi server dengan menyembunyikan jejak data pribadi pengguna. Layanan koneksi VPN memberikan keamanan pengguna saat akan mengakses website dengan mengubah jalur koneksi dengan server dan menyembunyikan pertukaran data. Beberapa fungsi dari VPN untuk aktivitas sehari-hari meliputi:

1. Menyembunyikan aktivitas *browsing*

   Ketika melakukan *browsing* dan mengakses konten penting yang tidak boleh dilihat oleh orang lain. VPN dapat membantu untuk menutupi dan menyembunyikan aktivitas internet yang sedang dilakukan.
2. Mengamankan transaksi *online*

   VPN dikenal berpengaruh pada aktivitas bisnis seperti transaksi *online*. Karena dapat berguna saat seseorang yang memiliki bisnis sering melakukan transaksi secara *online* lewat jaringan internet. VPN menjadi salah satu fitur yang wajib diaktifkan saat kita sedang berada di luar kota dan sedang menggunakan akses internet publik seperti *WiFi*. Pencurian data sering terjadi ketika kita menggunakan sumber internet yang bersifat *public* [17].

2.15 Cek Resi J&T.apk

Cek Resi J&T.apk adalah aplikasi yang mengandung *malware*. Aplikasi ini dengan sengaja dibuat oleh *Hacker* menyerupai aplikasi pengecekan resi pada J&T untuk mendapatkan informasi sensitif seperti: data keuangan, pesan pribadi, lokasi geografis atau rincian kontak. Cek Resi J&T.apk memungkinkan penerusan *input* tidak dipercaya ke sistem termasuk dalam kategori "*Moderate"* yang berdampak pada kerusakan reputasi bisnis secara pencurian kekayaan intelektual [18].

2.16 Undangan Pernikahan.apk

Undangan Pernikahan.apk adalah aplikasi yang mengandung *malware*. Aplikasi ini dengan sengaja dibuat oleh *Hacker* menyerupai aplikasi undangan pernikahan

digital pada umumnya untuk mendapatkan informasi *sensitive* dengan tujuan mengakuisisi akun *banking* korban. Ketika diinstal oleh korban, aplikasi akan meminta akses *permissions* SMS, tujuan dari akses ini adalah untuk mendapatkan OTP yang nantinya akan digunakan sebagai informasi untuk melakukan *reset password* dan transaksi [19].

2.17 *Creator*.apk

Creator.apk adalah aplikasi yang dibuat oleh penulis mengandung *malware* menggunakan Metasploit. Aplikasi ini dengan sengaja dibuat oleh penulis untuk melakukan simulasi serangan pada *virtual machine* sehingga mengetahui efektivitas penggunaan *Virtual Private Network* ketika terjadinya serangan *command and control.* Aplikasi ini sama seperti Cek Resi J&T dan Undangan Pernikahan yaitu akan meminta akses *permissions* kepada korban untuk dapat melakukan *controlling.*

2.18 *Speedtest by* Ookla

Speedtest by ookla adalah suatu aplikasi yang memiliki fungsi untuk menguji kecepatan dan performa koneksi internet dan parameter *jitter, download speed, upload speed* dan *packet loss* [20]. *Jitter* merupakan perubahan *latency* dari *delay* atau variasi waktu kedatangan paket. *Packet loss* merupakan parameter yang menggambarkan suatu kondisi dari jumlah total paket yang hilang pada suatu jaringan paket yang disebabkan oleh *collision,* penuhnya kapasitas jaringan dan penurunan paket [21]. Adapun untuk standar *packet loss* terdapat pada tabel 2.6.

Tabel 2.6 Standar *Packet Loss*

Packet Loss Category	*Packet Loss*	*Index*
Poor	>25%	1
Medium	12-24%	2
Good	3-14%	3
Perfect	0-2%	4