# *Security Assessment* pada Aplikasi *Mobile* Android XYZ dengan Mengacu pada Kerentanan OWASP *Mobile Top Ten* 2016

**Candra Kurniawan[1)], Nanang Trianto[2)]**
(1) *Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara, candra.kurniawan@student.poltekssn.ac.id*
(2) *Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara, nanang.trianto@poltekssn.ac.id*

Abstrak

*Aplikasi XYZ merupakan aplikasi unggulan pemerintah Provinsi X yang digunakan untuk pelayanan publik. Aplikasi ini menjadi penghubung antara pemerintah Provinsi dengan para Ketua RW. Pada penelitian ini dilakukan security assessment pada aplikasi XYZ untuk mengidentifikasi kerentanan dan dampak kerentanan, nilai kerentanan, serta memberikan rekomendasi keamanan pada kerentanan yang teridentifikasi. Kerentanan yang diidentifikasi mengacu pada kerentanan dari OWASP Mobile Top Ten 2016. Penelitian ini menggunakan metode security assessment berdasarkan SANS yang terdiri dari tiga langkah yaitu reviewing, examination, dan testing. Tahap reviewing dilakukan dengan mengumpulkan informasi terkait aplikasi XYZ, kebijakan Pemprov, dan terkait dengan NDA. Tahap examination disebut juga analisis statis, di sini dilakukan analisis statis otomatis menggunakan scanner MobSF dan MARA Framework. Kemudian tahap testing atau analisis dinamis dilakukan pengujian kerentanan dengan menjalankan aplikasi. Berdasarkan hasil security assessment teridentifikasi enam kerentanan pada aplikasi XYZ di mana enam kerentanan tersebut termasuk dalam lima kerentanan OWASP Mobile Top Ten 2016. Rincian kerentanan yang ditemukan insecure data storage (manipulatability backup dan aplikasi membuat file temp) kategori kerentanan medium, insecure communication (insecure implementation WebView) kategori kerentanan high, insufficient cryptography (static key) kategori kerentanan medium, client code quality (manipulatability activity) kategori kerentanan none, dan reverse engineering kategori kerentanan medium. Kerentanan yang ditemukan ini berdampak pada hilangnya aspek kerahasiaan seperti data sensitif pengguna, password default dan kunci konfigurasi aplikasi dengan server backend yang terdapat pada kode sumber. Berdasarkan kerentanan tersebut, diberikan rekomendasi keamanan berupa penerapan enkripsi data, penerapan teknik obfuscation, serta melakukan manajemen kunci untuk tujuan mengatasi kerentanan dan mencegah dampak yang terjadi.*

Kata kunci: Aplikasi android, OWASP Mobile Top Ten 2016, *security assessment*.

## 1. PENDAHULUAN

Negara Indonesia merupakan negara dengan jumlah pengguna Android terbesar keempat setelah India, Amerika Serikat, dan Brazil. Hal tersebut berdasarkan laporan yang dikeluarkan oleh Google pada Maret 2019 terkait keamanan dan privasi Android [1]. Berdasarkan laporan ptsecurity.com dalam laporan *Vulnerabilities and Threats in Mobile Application* 2019, disebutkan bahwa terdapat kerentanan dengan risiko tinggi sebanyak 43% pada aplikasi Android. Masalah keamanan yang umum ditemukan yaitu *insecure data storage* di mana pada laporan tersebut mencapai 76%. Sebagian besar masalah disebabkan karena kelemahan mekanisme keamanan sejak pembuatan aplikasi. Penggunaan *privileges* yang tidak tepat juga dapat dimanfaatkan penyerang untuk mendapatkan informasi finansial, data pribadi, *password* [2]. Besarnya risiko kerentanan tersebut berbanding lurus dengan jumlah aplikasi Android yang beredar di pasaran, Google mencatat hingga Maret 2018 terdapat 3,6 juta aplikasi pada Google Play Store di mana sebagian besar dari aplikasi tersebut dapat digunakan secara gratis.

Penggunaan aplikasi berbasis Android telah banyak digunakan di berbagai sektor, salah satunya sektor pemerintah. Aplikasi XYZ merupakan aplikasi berbasis Android milik Pemerintah Provinsi X yang memiliki fitur aspirasi, pelayanan publik, dan informasi. Dengan fitur aspirasi, masyarakat dapat menyampaikan aspirasi, saran, keluhan, dan fitur pelaporan ke pemerintah daerah. Fitur pelayanan publik meliputi pembayaran pajak kendaraan bermotor, perizinan dan administrasi, sedangkan fitur informasi meliputi informasi terkait harga pangan, lelang, dan nomor kontak darurat. Aplikasi ini merupakan produk unggulan Dinas Komunikasi dan Informatika Provinsi X yang digunakan secara luas di Provinsi X.

Aplikasi ini memerlukan *username, password* untuk bisa *login* serta terdapat data pribadi pada aplikasi. *Username, password*, data pribadi merupakan data penting yang harus dijaga kerahasiaannya [3], sehingga aplikasi harus memberikan keamanan yang optimal bagi pengguna. Hilangnya data dapat terjadi melalui jaringan Wi-Fi yang tidak aman, lemahnya kriptografi yang digunakan, serta tidak amannya penyimpanan [4].

Untuk mengetahui keamanan pada aplikasi dapat dilakukan *security assessment* untuk mengidentifikasi kerentanan [5]. *Security assessment* merupakan kegiatan untuk mengidentifikasi keamanan dari sistem informasi serta memberikan rekomendasi untuk perbaikan [6]. Pada penelitian ini, *security assessment* yang dilakukan mengacu pada kerentanan umum yang terdapat di OWASP *Mobile Top Ten*

2016. OWASP merupakan standar keamanan yang bertujuan untuk meningkatkan kesadaran keamanan bagi pengembang aplikasi [7]. Dalam OWASP *Mobile Top Ten* 2016 terdapat 10 risiko teratas yang mungkin terjadi pada aplikasi *mobile* yaitu, penggunaan *platform* yang tidak benar, penyimpanan data tidak aman, komunikasi tidak aman, autentikasi tidak aman, lemahnya kriptografi, otorisasi tidak aman, kualitas kode klien, *code tampering*, *reverse engineering*, dan *extraneous functionality* [8].

Pada penelitian sebelumnya yang dilakukan Darvish pada tahun 2018, Darvish melakukan analisis statis otomatis dan analisis dinamis pada 26 aplikasi *mobile* keuangan. Analisis statis otomatis dilakukan dengan menggunakan aplikasi AndroBugs untuk mendapatkan kerentanan yang ada, sedangkan analisis dinamis dilakukan dengan cara *network intercepting* menggunakan Charles Poxy dan Burp Suite. Hasil penelitian dari Darvish menunjukkan dari 26 aplikasi yang di analisis, hanya 4 aplikasi memiliki keamanan relatif baik [9].

Berdasarkan penjelasan di atas, pada penelitian ini akan dilakukan *security assessment* aplikasi XYZ untuk mengidentifikasi kerentanan beserta nilai kerentanannya dengan menggunakan metode *security assessment* yang mengacu pada OWASP *Mobile Top Ten* 2016, dan memberikan rekomendasi keamanan berdasarkan dari hasil *security assessment* yang dilakukan.

## 2. LANDASAN TEORI

Bagian Landasan Teori ini membahas teori-teori yang terkait dengan pembahasan yang diperlukan.

### 2.1 *Security Assessment*

*Security assessment* dilakukan untuk mengidentifikasi keamanan dari sistem informasi serta memberikan rekomendasi untuk perbaikan yang memungkinkan untuk mengurangi risiko pada organisasi. *Security assessment* pada aset bergantung pada tiga metode yaitu *reviewing, examination, testing* [10]. *Security assessment* biasa digunakan untuk memperkirakan postur keamanan siber dengan mengidentifikasi kelemahan keamanan dan celah yang dapat digunakan oleh penyerang untuk melakukan serangan siber terhadap target. Secara umum *security assessment* menawarkan rekomendasi dan pedoman untuk mekanisme meningkatkan keamanan dan perlindungan untuk mengurangi risiko dan menghindari potensi ancaman keamanan [6].

### 2.2 OWASP *Mobile Top Ten* 2016

Perkembangan teknologi yang semakin maju memberikan dampak keamanan yang baru. Masalah keamaan pada aplikasi *mobile* berbeda dengan aplikasi web, banyak masalah muncul pada aplikasi *mobile* seperti masalah penyimpanan data, IPC, penggunaan API kriptografi, komunikasi jaringan yang tidak aman. Pada tahun 2015, OWASP melakukan survei untuk menganalisis dan mengategorikan kerentanan-kerentanan pada aplikasi *mobile* yang mana selanjutnya dimasukkan dalam OWASP *Mobile Top Ten* 2016, sehingga sepuluh kerentanan ini lebih berfokus pada aplikasi *mobile* bukan pada servernya. Sepuluh kerentanan yang masuk dalam daftar kerentanan teratas OWASP *Mobile Top Ten* 2016, yaitu [4] [8] [11]:

a. *Improper Platform Usage* [M1]

Kerentanan ini mencakup penyalahgunaan fitur *platform* atau kegagalan dalam menggunakan kontrol keamanan sehingga pengguna yang tidak berhak dapat mengakses fitur-fitur tertentu. Kerentanan ini mencakup penyalahgunaan dari *Touch ID, Keychain, Android intents*. Kerentanan ini biasanya dilakukan dengan eksploit *Android intent* yang merupakan objek pengiriman pesan dalam sistem operasi yang dapat berkomunikasi antar aktivitas termasuk mengakses data yang disimpan pada perangkat seluler atau server.

b. *Insecure Data Storage* [M2]

Kerentanan ini mencakup penyimpanan yang tidak aman dan kebocoran data. Ketika suatu aplikasi memiliki kerentanan ini maka penyerang dapat memperoleh informasi pribadi, memperoleh kredensial untuk *login* pada suatu aplikasi. Kerentanan ini berupa *compromise file system (database SQL, log files, manifest file, binary data stores, cookie stores, sd card, cloud synced).*

c. *Insecure Communication* [M3]

Kerentanan ini merupakan kerentanan yang umum terjadi pada aplikasi dengan struktur klien-server di mana data yang ditransfer tidak dienkripsi mengunakan SSL/TLS. Kerentanan ini dapat dilakukan dengan serangan man-in-the-middle atau intersepsi. Risiko dari *insecure communication* seputar integritas data, kerahasiaan data.

d. *Insecure Authentication* [M4]

Kerentanan ini mencakup kelemahan dalam prosedur autentikasi dan pengaturan sesi. Autentikasi pada aplikasi *mobile* lebih ramping daripada untuk aplikasi web, hal ini karena terdapat aplikasi yang bekerja ketika *offline*. Autentikasi offline mudah untuk dieksploitasi sehingga penyerang dapat memperoleh kontrol penuh. Kerentanan ini dapat terjadi ketika penyerang mampu melewati protokol autentikasi.

e. *Insufficient Cryptography* [M5]

Kerentanan ini terjadi karena penggunaan kriptografi yang lemah. Selain itu dapat disebabkan juga karena algoritma kriptografi yang digunakan sudah tidak aman, sehingga penyerang dapat mendekripsi informasi yang diperoleh.

f. *Insecure Authorization* [M6]

Kerentanan ini mengacu pada kegagalan server untuk menerapkan identitas dan izin dengan benar. Kerentanan ini mencakup komunikasi otorisasi antara

aplikasi dan *server backend*, sehingga kerentanan ini berbeda dengan M4 di mana M4 mengacu pada pengguna yang mengelabuhi proses autentikasi dalam aplikasi.

g. *Client Code Quality* [M7]

Kerentanan ini terjadi karena kesalahan dalam pembuatan kode, sehingga dapat dimanfaatkan penyerang untuk mengeksploitasi logika dan berpotensi mem-*bypass* kontrol keamanan yang diterapkan pada perangkat. Contoh dari kerentanan ini yaitu terjadinya *buffer overflow*, kerentanan format string.

h. *Code Tampering* [M8]

Kerentanan ini mencakup tindakan modifikasi yang dilakukan oleh penyerang pada kode aplikasi. Dengan kerentanan ini memungkinkan penyerang untuk menginstal *backdoor* pada aplikasi, melakukan *re-sign* dan menerbitkan kembali aplikasi dalam versi jahat.

i. *Reverse Engineering* [M9]

Kerentanan ini digunakan untuk mencari informasi terkait algoritma yang digunakan untuk enkripsi, cara kerja server *back-end* yang seharusnya dilindungi.

j. *Extraneous Functionality* [M10]

Kerentanan ini terjadi karena adanya bug atau backdoor yang digunakan untuk memudahkan ketika siklus pengembangan namun ditinggalkan oleh pengembangnya setelah produksi, sehingga penyerang dapat masuk melalui jalan tersebut.

### 2.3 *Common Vulnerability Scoring System* (CVSS)

CVSS merupakan kerangka kerja yang digunakan untuk menentukan tingkat kerentanan pada perangkat lunak. CVSS terdiri dari tiga grup metrik yaitu *base metric, temporal*, dan *environmental*. *Base metric* digunakan untuk menentukan kualitas intrinsik kerentanan di mana tidak berubah berdasarkan waktu dan lingkungan. *Temporal metric* digunakan untuk menentukan karakteristik kerentanan yang berubah sepanjang waktu. *Environmental metric* digunakan untuk menentukan kerentanan yang berdasarkan pada lingkungan yang unik atau lingkungan tertentu [12].

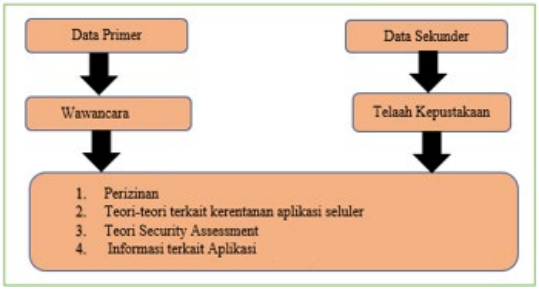
## 3. METODE PENELITIAN

Metode penelitian yang digunakan pada penelitian ini adalah metode kualitatif. Metode kualitatif merupakan metode yang dilakukan untuk mengumpulkan data, data yang didapatkan dianalisis secara kualitatif dengan didukung teori dan wawasan terhadap penelitian yang dilakukan untuk mendapatkan hasil yang jelas dan bermakna [13]. Pada penelitian ini, metode kualitatif digunakan untuk mengumpulkan data dan dokumen pendukung, teori terkait kerentanan aplikasi, teori terkait *security assessment*, pelaksanaan *security assessment*.

Teknik *Security assessment* dilakukan dengan menggunakan metode *security assessment*. Metode ini terdiri dari 3 tahap yaitu: *reviewing, examination*, dan *testing* [10].

a. Tahap I *Reviewing*

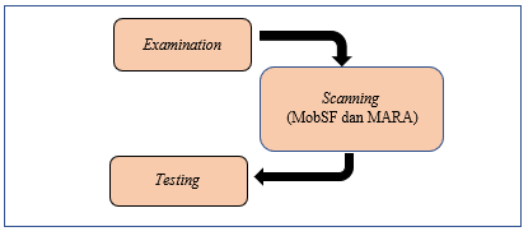
Tahapan ini meliputi teknik pengumpulan data dengan wawancara, telaah kepustakaan yang dilakukan secara manual serta terkait perizinan. Tahap ini membantu untuk mengevaluasi sistem, aplikasi, kebijakan, dan prosedur untuk menemukan kerentanan. Tahap ini disebut juga dengan tahap pengumpulan informasi terkait aplikasi seperti pada Gambar 1.




Gambar 1. Tahap *Reviewing*

b. Tahap II *Examination*

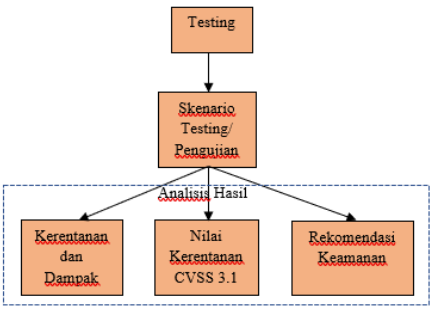
Tahapan ini merupakan tahap untuk mengidentifikasi kerentanan yang ada pada aplikasi atau sistem, termasuk melakukan pemindaian kerentanan seperti Gambar 2. Pada tahap ini akan dilakukan analisis statis otomatis untuk mengidentifikasi kerentanan yang ada pada aplikasi.




Gambar 2. Tahap *Examination*

c. Tahap III *Testing*

Tahapan ini merupakan tahap pengujian terhadap kerentanan yang ditemukan serta untuk membuktikan bahwa kerentanan yang ditemukan valid. Tahap ini merupakan tahap analisis dinamis yang dilakukan dengan cara menjalankan aplikasi XYZ seperti Gambar 3.




Gambar 3. Tahap *Testing*

Penilaian kerentanan dilakukan pada kerentanan yang valid dilakukan. Penilaian kerentanan menggunakan CVSS 3.1 dengan *base metric*. Digunakan *base metric* karena penilaian dilakukan oleh peneliti berdasarkan kemudahan dalam melakukan serangan *(exploitability metric)* dan dampak dari kerentanan yang dieksploitasi *(impact metric). Temporal metric* dan *environmental metric* tidak digunakan dalam penelitian ini. Penilaian *temporal metric* berkaitan dengan ketersediaan referensi kerentanan dan penilaian *environment metric* berkaitan dengan subyektif pemilik aset. Kemudian setelah dilakukan penilaian kerentanan selanjutnya diberikan rekomendasi keamanan untuk mengurangi atau mencegah dampak yang ditimbulkan dari kerentanan yang ada.

## 4. HASIL DAN PEMBAHASAN

### 4.1 *Reviewing*

Aplikasi XYZ merupakan produk unggulan Diskominfo Provinsi X dan digunakan secara luas di Provinsi X. Aplikasi ini digunakan untuk mempermudah warga untuk menyampaikan aspirasi, mendapatkan informasi, dan mendapatkan layanan publik. Aplikasi ini juga digunakan untuk koordinasi bantuan sosial Covid-19 di Provinsi X. Aplikasi ini telah diunduh lebih dari 10.000 kali, aplikasi ini memiliki beberapa fitur:

a. Aspirasi: fitur ini digunakan untuk menyampaikan aspirasi, saran, keluhan, dan ide dari masyarakat melalui poling, survei, serta fitur pelaporan ke pemerintah daerah.
b. Informasi: fitur ini digunakan untuk mendapatkan *broadcast* informasi terkait program pemerintah dan informasi penting, termasuk harga pangan, lelang, dan nomor kontak darurat.
c. Pelayanan publik: fitur ini digunakan untuk efisiensi pelayanan publik, termasuk hal pembayaran pajak kendaraan bermotor, perizinan, dan administrasi.

Selain fitur diatas, pada aplikasi ini juga terdapat data pribadi pengguna seperti *username*, pendidikan, jabatan, alamat *email*, alamat rumah, alamat media sosial, nomor telepon, tanggal lahir.

Aplikasi XYZ memiliki izin penggunaan aplikasi ketika proses penginstalan. Ketika menginstal Aplikasi XYZ pengguna akan diminta persetujuan terkait izin seperti berikut:

a. Kamera: izin untuk mengambil gambar atau video.
b. Lokasi: izin untuk mengakses lokasi akurat atau perkiraan lokasi.
c. Mikrofon: izin untuk merekam audio.
d. Penyimpanan: izin untuk mengubah atau menghapus isi penyimpanan USB, serta untuk membaca konten yang tersimpan di USB.
e. Jaringan: izin untuk mendapatkan akses jaringan secara penuh.

### 4.2 *Examination*

Berikut merupakan perbandingan hasil pemindaian kerentanan dengan menggunakan MobSF dan MARA.

Tabel 1. Hasil *Examination*

| OWASP *Mobile Top Ten* 2016 | MobSF | MARA |
|---|---|---|
| *M1 Improper Platform Usage* | - | - |
| *M2 Insecure Data Storage* | √ | √ |
| *M3 Insecure Communication* | √ | - |
| *M4 Insecure Authentication* | - | - |
| *M5 Insufficient Cryptography* | √ | √ |
| *M6 Insecure Authorization* | - | - |
| *M7 Client Code Quality* | √ | √ |
| *M8 Code Tampering* | - | - |
| *M9 Reverse Engineering* | √ | - |
| *M10 Extraneous Functionality* | - | - |

Dari tabel 1 diketahui bahwa MobSF mengidentifikasi 5 isu kerentanan dan MARA mengidentifikasi 3 kerentanan yang mengacu pada OWASP *Mobile Top Ten* 2016. Kerentanan pada MobSF meliputi:

a. *Insecure Data Storage*

Isu kerentanan yang teridentifikasi pada aplikasi ini berupa aplikasi dapat menulis atau membaca ke penyimpanan eksternal dan aplikasi dapat membuat file temp untuk menyimpan informasi sensitif. Kerentanan ini memiliki kategori tinggi karena menyebabkan hilangnya data pengguna.

b. *Insecure Communication*

Isu kerentanan pada aplikasi ini berupa implementasi *WebView* yang tidak aman di mana tampilan web mengabaikan kesalahan sertifikat SSL dan menerima sertifikat SSL, sehingga aplikasi ini rentan terhadap serangan MITM. Isu kerentanan ini memiliki kategori tinggi karena menyebabkan komunikasi pengguna akan diketahui oleh pihak lain.

c. *Client Code Quality*

Isu kerentanan yang teridentifikasi dengan MobSF berupa adanya *database* yang menggunakan SQLite dan aplikasi mengeksekusi baris perintah kueri sql. Kerentanan ini memiliki kategori tinggi karena input pengguna yang tidak sesuai pada baris kueri sql dapat menyebabkan *sql injection*, dan jika informasi sensitif tidak dienkripsi maka akan diketahui pihak lain.

d. *Reverse Engineering*

Aplikasi ini dapat dilakukan *reverse engineering* sehingga memungkinkan akan didapatkan informasi seperti *usernames, password*, kunci, dan lain-lain.

e. *Insufficient Cryptography*

Isu kerentanan kriptografi yang teridentifikasi pada aplikasi ini berupa penggunaan *java hascode* yang lemah. Kerentanan ini hanya sebagai peringatan.

Kemudian untuk isu kerentanan yang teridentifikasi pada MARA meliputi:

a. *Insecure Data Storage*

Isu kerentanan pada aplikasi ini berupa adanya fitur pada aplikasi yang diizinkan untuk melakukan *backup* dan *restore* dengan menggunakan fitur *full backup*. Pada MARA kerentanan ini memiliki kategori kritis karena memungkinkan untuk menjadi sasaran serangan *backup* yang menyebabkan hilangnya data pengguna.

b. *Insufficient Cryptography*

Isu kerentanan yang teridentifikasi pada MARA berupa *Insecure cryptography: Non-Random Xor Cipher* dan *Insecure cryptography: Static keys*.

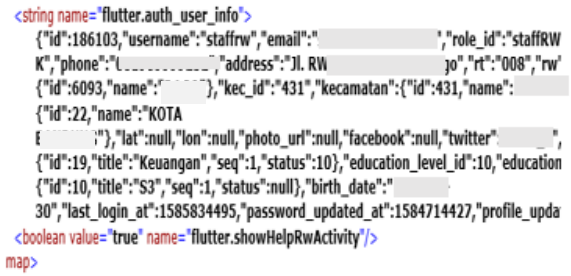
c. *Client Code Quality*

Isu kerentanan pada aplikasi ini berupa *manipulatable activity with private action names* dan *manipulatable broadcastreceiver with private action names*. Isu kerentanan ini terjadi ketika terdapat komponen Android yang diekspor yang dapat dimanfaatkan oleh pihak tertentu. Dengan adanya komponen yang diekspor memungkinkan aplikasi dipanggil dari sistem atau aplikasi lain. Kerentanan ini tergolong kategori tinggi karena dapat mem-*bypass* suatu *activity.*

### 4.3 *Testing*

#### 4.3.1 *Insecure Data Storage*

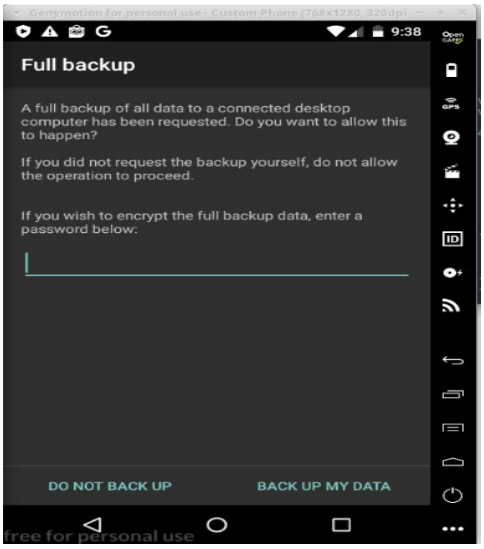
Dalam membuktikan isu kerentanan *insecure data storage* digunakan *tools* adb. Dengan menggunakan adb penguji dapat masuk ke folder paket aplikasi yang terinstal pada perangkat emulator. Pada pengujian ini diperoleh data pengguna seperti *username*, alamat *email*, nama, nomor telepon, alamat rumah, alamat media sosial, tanggal lahir seperti pada Gambar 4. Data ini ditemukan pada penyimpanan aplikasi di *path data/data/com.example.xyz.staging/shared_prefs*. Dalam folder *shared_prefs* terdapat 9 file namun hanya file dengan nama *FlutterSharedPreferences.xml* yang mengandung data tersebut. *Shared preferences* merupakan salah satu penyimpanan Android yang menggunakan format xml.




Gambar 4. Temporari File

Selain diperoleh data tersebut, aplikasi ini juga mengizinkan untuk melakukan *backup* secara penuh dengan menggunakan adb. Hasil *backup* ini berupa file-file *database* dari aplikasi. Ketika dilakukan perintah menggunakan adb, perangkat akan memberikan pilihan untuk mengenkripsi hasil *backup*dengan memberikan *password*, namun tanpa memberikan *password* tetap dapat dilakukan *backup*.

Kerentanan *insecure data storage* aplikasi membuat file temp memiliki nilai kerentanan 6,2 medium dengan vektor /AV:L/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N. Kerentanan *insecure data storage full backup* memiliki nilai kerentanan 5,5 medium dengan vektor /AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:N/A:N. Hal ini diperoleh dari perhitungan menggunakan CVSS 3.1.




Gambar 5. *Full Backup*

Berdasarkan dari kerentanan *insecure data storage* berupa aplikasi membuat file temp dan *full backup* pada Gambar 4 dan 5 yang telah dilakukan, kerentanan tersebut berdampak pada hilangnya kerahasiaan yang berupa dapat diketahuinya informasi pribadi pengguna. Seperti terlihat pada Gambar 4 bahwa terdapat data pribadi seperti nama, alamat *email*, alamat rumah, nomor ponsel, alamat media sosial, jabatan, pendidikan, dan tanggal lahir. Ketika data pribadi pengguna tersebar maka hal tersebut dapat disalahgunakan oleh pihak yang tidak bertanggung jawab untuk melakukan hal-hal terlarang seperti pelanggaran privasi, penipuan, merusak reputasi, serta hal yang menyebabkan kerugian material.

Berdasarkan CWE-312 *Cleartext Storage of Sensitive Information*, penyimpanan informasi sensitif aplikasi dapat dilindungi dengan menerapkan enkripsi pada data yang ada sehingga meskipun informasi diakses orang lain namun tidak mudah terbaca. Selanjutnya untuk mengatasi kerentanan *full backup* dapat dilakukan dengan cara mengatur atribut android:allowBackup ke *false* (android:allowBackup=”false”).

#### 4.3.2 *Reverse Engineering*

*Reverse engineering* dilakukan untuk mendapatkan kode sumber dari aplikasi. *Reverse* ini dilakukan dengan menggunakan *tools apktool* dan

MARA. Dengan melakukan *reverse* diperoleh folder *assets, build, kotline, lib, original, res, smali, smali_classes2, unknown*. Selain itu juga diperoleh file *AndroidManifest.xml*. Kode sumber aplikasi ini tidak di *obfuscation* sehingga kode-kode tidak terenkripsi dan lebih mudah untuk dibaca. Pada salah satu file dalam folder ditemukan kode-kode penting yang seharusnya tidak diperlihatkan secara tidak terenkripsi karena kode tersebut dapat dimanfaatkan oleh pihak yang tidak berhak. Adapun kode yang terlihat yaitu *default_web_client_id, firebase_database_url, defaultSenderID, google_api_key, google_app_id, project_id*. Kode tersebut berada pada folder *res/value/string.xml*.

Google *Api Key* yaitu *api key* untuk layanan *database firebase*. *Key* tersebut tercantum sebagai *string* yang dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab untuk mengakses *firebase*. Kerentanan ini berdampak pada hilangnya kerahasiaan di mana dapat terungkapnya data pada server *backend* yang mengakibatkan tersebarnya data-data penting pada server *backend*.

Berdasarkan penilaian kerentanan dengan menggunakan CVSS 3.1, Kerentanan *reverse engineering* memiliki nilai kerentanan 5,1 medium dengan vektor /AV:P/AC:L/PR:N/UI:N/S:C/C:H/I:N/A:N. Kerentanan ini termasuk dalam CWE-312 *Cleartext Storage of Sensitive Information*. Kerentanan ini dapat menyebabkan pencurian kekayaan intelektual dari informasi yang didapatkan, serta dapat dilakukan modifikasi kode aplikasi. Untuk mencegah hal tersebut, OWASP memberikan rekomendasi untuk menerapkan *obfuscation* pada kode aplikasi sehingga tidak mudah untuk dilakukan *reverse engineering* dan meskipun dapat dilakukan *reverse* namun kode tidak mudah dibaca.

### 4.3.3 *Insecure Communication*

Pada isu kerentanan ini dilakukan *testing* dengan menggunakan aplikasi Burp Suite. Berdasarkan dari *testing* tersebut diperoleh hasil bahwa pada fitur Info PKB di mana fitur tersebut diteruskan ke *website* Bapenda X, diketahui bahwa tidak menerapkan pengamanan saat transfer data atau komunikasi yang aman. Hal tersebut terlihat dari *testing* yang dilakukan di mana terdapat *request* berupa masukan nomor polisi yang dicobakan tertangkap Burp Suite secara *plaintext* beserta kode validasinya. Selain *request*, respon dari *request* juga tertangkap di Burp Suite berupa informasi kendaraan dan pajak. Kemudian pada masukan nomor polisi juga dapat dilakukan manipulasi masukan, sehingga hasil yang muncul pada tampilan aplikasi akan berbeda dari masukan awal.

Kerentanan ini juga mengungkap data terkait nomor rangka kendaraan bermotor yang seharusnya berdasarkan tampilan pada aplikasi disamarkan dengan xxx, namun tertangkap pada Burp Suite tidak disamarkan dengan xxx sehingga terlihat secara plain text. Nomor rangka kendaraan yang terekspos tersebut seperti pada Gambar 6.

Berdasarkan hasil *insecure communication* ini diketahui bahwa aplikasi ini tidak menerapkan pengamanan ketika transfer data sehingga menyebabkan terungkapnya *request* dan *respon,* dapat dilakukan manipulasi masukan pada fitur info PKB, dan adanya nomor rangka kendaraan yang terekspos. Kerentanan ini dapat berdampak pada hilangnya kerahasiaan dan integritas data yang dapat menyebabkan menurunnya tingkat kepercayaan pengguna aplikasi. Kemudian nomor rangka kendaraan merupakan suatu identitas fisik kendaraan bermotor yang digunakan sebagai legitimasi operasional dari pemilik kendaraan, seharusnya informasi ini harus disamarkan dengan xxx seperti pada gambar 7 sesuai di tampilan aplikasi guna menjamin kepercayaan pengguna aplikasi dan pemilik kendaraan bermotor, namun pada Gambar 6 diketahui bahwa nomor rangka tidak disamarkan.

Gambar 6. Nomor Rangka


Gambar 7. Nomor Rangka XXX

Berdasarkan penilaian kerentanan, *insecure communication* memiliki nilai kerentanan 7,5 *high* dengan vektor /AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N. Kerentanan ini terdapat pada CWE-311 *Mission Encryption of Sensitive Data*. Jika terdapat seseorang yang melakukan intersepsi pada jaringan maka akan berdampak pada data kepemilikan kendaraan pengguna aplikasi yang dapat diketahui orang lain. Di

mana data saat ditransfer tidak dienkripsi terlebih dahulu sehingga informasi yang dikirim melalui jaringan tertangkap di Burp Suite. Oleh karena itu perlu untuk menerapkan pengamanan pada komunikasi tersebut. Tindakan pengamanan dapat dilakukan dengan menerapkan enkripsi pada masukan pengguna sebelum data ditransfer melalui jaringan, begitu juga sebaliknya saat server merespon permintaan juga harus menerapkan enkripsi untuk mencegah terjadinyan MITM.

#### 4.3.4 *Insufficient Cryptography*

Isu kerentanan *static key,* setelah dilakukan analisis pada *source code* aplikasi ditemukan file *environment.dart* dan *FirebaseConfig.dart* yang mengandung *static key.* Kedua file tersebut masing-masing berada pada folder *lib/environment* dan *lib/constants*. File *environment.dart* merupakan *backed* aplikasi berupa informasi server dan *database* di mana pada file tersebut terdapat *string defaultpassword* dan *googleApikey.* Kemudian pada file *FirebaseConfig.dart* terdapat *string secretKeys*.

Kerentanan ini akan berdampak pada hilangnya kerahasiaan di mana pihak yang tidak bertanggung jawab dapat melakukan pengambilan informasi sensitif dari aplikasi atau dari server *backend*.

Berdasarkan penilaian kerentanan, kerentanan ini memiliki nilai 4,6 medium dengan vektor /AV:P/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:N.
Kerentanan tersebut dijelaskan dalam CWE 798 *Use of Hard-coded Credential*. Disebutkan pada CWE ini berupa aplikasi mengandung kredensial *hardcode* seperti *password* atau kunci kriptografi yang digunakan untuk komunikasi ke komponen eksternal dari sistem frontend ke layanan *backend*. Untuk mencegah terjadinya hal yang tidak diinginkan maka perlu menyimpan kunci dan kredensial diluar file konfigurasi atau menerapkan enkripsi (*key management*). Jika aplikasi memang memerlukan kredensial untuk dapat berhubungan dengan komponen lain maka diberlakukan teknik obfuscation pada *source code* sehingga kredensial tidak mudah terbaca [14].

#### 4.3.5 *Client Code Quality*

Isu kerentanan *manipulatable activity with private action name* terjadi karena adanya ekspor *activity* yang diatur ke *true* sehingga memungkinkan seseorang untuk dapat menjalankan aplikasi tanpa perlu menjadi pengguna yang terautentikasi. *Activity* yang diekspor yaitu *activity* `com.example.xyz.MainActivity`.
Kerentanan ini tidak berdampak pada kerahasiaan, integritas, dan ketersediaan sesuai dengan tujuan keamanan karena tidak mengungkap data serta tidak berpengaruh pada data sensitif. Kerentanan *activity* ini akan berdampak pada kerahasiaan, integritas, dan ketersediaan. Namun jika *activity* yang diekspor merupakan *activity* terkait *database* atau terkait dengan pembayaran pajak kendaraan, di mana hal tersebut dapat diketahui oleh pihak yang tidak berhak.

Berdasarkan penilaian kerentanan, kerentanan ini memiliki nilai 0,0 *none*. Kategori kerentanan *none* atau tidak rentan terjadi karena tidak ada aspek kerahasiaan, integritas dan ketersediaan yang terungkap, namun meskipun begitu kerentanan ini dapat dicegah dengan mengatur ekspor *activity* com.example.xyz.MainActivity ke *false.*

## 5. KESIMPULAN

Berdasarkan penelitian ini dapat disimpulkan kerentanan yang ditemukan yaitu:

a. *Insecure data storage*
Berupa kerentanan pada fitur *full backup* dan adanya file temp. Kerentanan ini termasuk dalam kategori medium. Kerentanan ini berdampak pada hilangnya aspek kerahasiaan berupa terungkapnya data sensitif pengguna dan ter-*backup*nya data pada aplikasi. Rekomendasi keamanan dari kerentanan ini dapat menerapkan enkripsi pada file temp sehingga data tidak berbentuk plaintext dan untuk mencegah *full backup* dapat mengatur pada Android *manifest* dengan menerapkan *allow backup* ke *false*.

b. *Reverse engineering*
Kerentanan ini termasuk dalam kategori kerentanan medium. Kerentanan ini berdampak pada hilangnya aspek kerahasiaan berupa terungkapnya data sensitif berupa *google api key* dan *project id* yang ditemukan pada *string*. Rekomendasi keamanan dari kerentanan ini dapat menerapkan teknik *obfuscation* pada kode sumber aplikasi sehingga ketika dilakukan *reverse engineering string* tidak berbentuk *plaintext.*

c. *Insecure communication*
Kerentanan ini termasuk dalam kategori kerentanan *high*. Kerentanan ini berdampak pada hilangnya aspek kerahasiaan berupa terungkapnya data kendaraan bermotor serta *request* dan *respon* yang tidak diamankan. Rekomendasi keamanan dari kerentanan ini dapat menerapkan enkripsi data sebelum ditransfer sehingga ketika dilakukan MITM data tidak berbentuk *plaintext.*

d. *Insufficient cryptography*
Kerentanan ini termasuk dalam kategori kerentanan medium. Kerentanan ini berdampak pada hilangnya aspek kerahasiaan berupa penggunaan password default dan kunci konfigurasi server *backend* yang tercantum pada kode sumber. Rekomendasi keamanan dari kerentanan ini dapat menerapkan manajemen kunci dan jika kunci konfigurasi harus berada pada kode sumber maka dapat menerapkan enkripsi atau *obfuscation* pada kode sumber.

e. *Client code quality*
Kerentanan ini termasuk dalam kategori kerentanan *none* atau tidak rentan karena tidak berdampak pada aspek kerahasiaan, integritas, atau ketersediaan.

**REFERENSI**


[1] "Android Security & Privacy 2018 Year in Review," Google, 2019.

[2] ptsecurity.com, "Vulnerabilities and threats in mobile applications," ptsecurity.com, 2019.

[3] B. Custer, A. M.Sears, F. Dechesne, I. Georgieva, T. Tani dan S. van der Hof, EU Personal Data Protection in Policy and Practice, Netherland: Asser Press, 2019.

[4] G. Basatwar, "OWASP Mobile Top 10: A comprehensive guide for mobile developers to counter risks," 23 January 2020. [Online]. Available: https://www.appsealing.com/owasp-mobile-top-10-a-comprehensive-guide-for-mobile-developers-to-counter-risks/.

[5] A. A. Ali dan M. Z. Murah, "Security Assessment of Libyan Government websites," dalam *Cyber Resilience Conference 2018*, Malaysia, 2018.

[6] Q. Qassim, N. Jamil, M. Daud, A. Petel dan N. Ja'affar, "A Review of Security Assessment Methodologies in Individual Control Systems," *Information and Computer Security,* pp. 47-61, 2019.

[7] OWASP.org, "OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risk," The OWASP Foundation, 2017.

[8] B. Mueller, S. Schleier dan J. Williemsen, Mobile Security Testing Guide, The OWASP Foundation, 2018.

[9] H. Darvish dan M. Husain, "Security Analysis of Mobile Money Applications on Android," dalam *International Conference on Big Data*, 2018.

[10] A. Abdel-Aziz, "Scoping Security Assessment- A Project Management Approach," SANS Institute, 2011.

[11] K. Mokris, "Building bloc for secure mobile development: Testing for the OWASP Mobile Top 10," 13 October 2016. [Online]. Available: https://www.nowsecure.com/blog/2016/10/13/secure-mobile-development-testing-owasp-mobile-top-10/.

[12] FIRST, "Common Vulnerability Scoring System version 3.1 Rev 1," FIRST, 2019.

[13] Sugiyono, Metode Penelitian Kuantitatif Kualitatif dan R&D, Bandung: Alfabeta, 2016.

[14] CWE, "CWE-798: Use of Hard-coded Crendential," 25 June 2020. [Online]. Available: https://cwe.mitre.org/data/definitions/798.html. [Diakses 7 July 2020].

[15] M. Zhang dan H. Yin, Android Application Security, Switzerland: Springer, 2016.

[16] "Android Security & Privacy 2018 Year in Review," Google, 2019.