Jurnal Infosecure Vol.4 No.2, Oktober 2023
p-ISSN: 2087-1023
MTI-UNLA

Analsis Keamanan Pada Aplikasi *Android* Menggunakan Metode *Static Vulnerability Assessment and Penetration Testing* (VAPT)

Studi Kasus: Aplikasi Kependudukan Digital

Mochamad Taopik Haryanto[1], Aisyah Nuraeni[2], Ali Ahmadi[3]

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Langlangbuana[1,2,3]

[1]mochammad.taopik10@gmail.com

[2]aisyahnuraeni20@gmail.com

[3]kang.aliahmadi@gmail.com

Abstrak— Penelitian ini bertujuan untuk mengidentifikasi celah keamanan pada aplikasi Identitas Kependudukan Digital dengan metode analisis *Static Vulnerability Assessment and Penetration Testing* (VAPT), sesuai standar OWASP *Top 10 Mobile Risk*. Metode yang digunakan adalah kualitatif, dengan pengumpulan data melalui studi pustaka dan observasi. Hasil analisis menunjukkan adanya beberapa kerentanan, termasuk *M2 Insecure Data Storage, M3 Insecure Communication, M8 Code Tampering, dan M9 Reverse Engineering. Static vulnerability assessment* perlu divalidasi melalui *penetration testing* untuk memastikan ketepatan hasil. Rekomendasi untuk *M3 Insecure Communication* adalah penerapan *hashing* pada *body message* untuk mengamankan komunikasi klien-server, sedangkan untuk *M9 Reverse Engineering*, disarankan implementasi deteksi *root* atau *jailbreak* guna mencegah peretasan ilegal. Uji prototipe membuktikan *hashing* berhasil melindungi komunikasi dan aplikasi tidak berjalan pada perangkat yang di-*root*, sehingga menyulitkan *reverse engineering*. Penelitian ini berkontribusi dalam pengembangan aplikasi yang lebih aman dan meningkatkan kesadaran akan pentingnya keamanan aplikasi di era digital.

Kata kunci— Keamanan aplikasi, *Android*, VAPT, OWASP

I. PENDAHULUAN

Dalam era digital yang terus berkembang, penggunaan aplikasi mobile meningkat secara signifikan di berbagai sektor [1], termasuk sektor pemerintahan.

Pada aspek pemerintahan perkembangan teknologi informasi berdampak pada proses pelayanan publik yang berubah dari manual menjadi digital atau elektronik [4]. Sebagai studi kasus, Estonia telah berhasil membangun sistem data nasional yang terintegrasi melalui e-Estonia. Sistem ini memungkinkan berbagai instansi pemerintah untuk berbagi data secara efisien dan untuk meningkatkan pelayanan publik [5]. Salah satu inovasi dalam pelayanan publik adalah implementasi Aplikasi Identitas Kependudukan Digital (IKD) yang dirancang untuk mempermudah akses terhadap data kependudukan. Namun, pesatnya perkembangan teknologi ini juga menghadirkan tantangan besar terkait keamanan data. Serangan siber yang semakin canggih, seperti pencurian data melalui kerentanan aplikasi, menuntut penerapan sistem keamanan yang lebih ketat.

Aplikasi IKD, yang telah diunduh oleh jutaan pengguna, [3] berpotensi menjadi sasaran serangan siber yang memanfaatkan kerentanan aplikasi, terutama pada aspek komunikasi dan penyimpanan data yang tidak aman.

Berdasarkan penelitian [6] menunjukan bahwa Penelitian ini berhasil menganalisis infeksi *malware* Trojan Downloader pada aplikasi *Android* melalui teknik *reverse engineering*. Hasilnya menunjukkan adanya perubahan signifikan pada ukuran file, *hashing*, izin (*permissions*. Studi ini menggarisbawahi pentingnya deteksi malware secara dini untuk mencegah akses berbahaya ke perangkat dan data pengguna. Pada penelitian [7] membahas mengenai penggunaan *tools* MOBSF untuk mendeteksi isu kerentanan pada aplikasi *android*. Pada penelitian [8] menunjukan bahwa penggunaan OWASP *Top 10 Mobile Risk* sebagai parameter kerentanan, pada penelitian [9] menggunakan menggabungkan metode statis untuk proses *vulnerability assessment* dan metode dinamik. Pada penelitian [10] menunjukan bahwa penggunaan metode *hybrid* memberikan hasil yang lebih baik pada tahapan deteksi kerentanan. Pada penelitian [11] menguji pada dua kategori yaitu pada autentikasi dan penyimpanan berdasarkan OWASP MASTG. Pada Penelitian [12] melakukan analisis keamanan pada aplikasi *mobile payment* khususnya pada kebocoran informasi dan penyimpanan data *sensitive*. Pada penelitian [13] menganalisis aplikasi pemerintahan berbasis android dengan mengacu pada OWASP *Top 10 Mobile Risk*. Pada penelitian [14] melakukan pengujian pada aplikasi *video streaming* menggunakan metode statis. Pada penelitian [15] metode yang digunakan adalah pendekatan berbasis graf ketergantungan memberikan kerangka kerja yang sistematis dan komprehensif untuk pengujian kerentanan pada aplikasi perangkat lunak seluler.

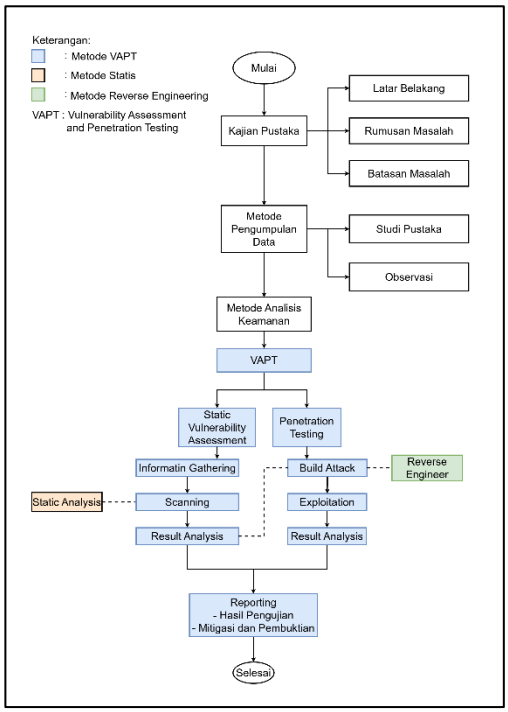
Berdasarkan hal tersebut, penelitian ini bertujuan untuk mengidentifikasi kerentanan keamanan pada aplikasi IKD, menggunakan metode *Static Vulnerability Assessment and Penetration Testing* (VAPT) dengan standar OWASP *Top 10 Mobile Risk.* Pendekatan ini menggabungkan analisis statis dan

dinamis untuk memastikan celah keamanan dapat diidentifikasi secara komprehensif.

Metodologi penelitian ini bersifat kualitatif, dengan data yang diperoleh melalui studi literatur dan observasi langsung terhadap aplikasi. Pengujian keamanan dilakukan untuk mengidentifikasi risiko dengan prioritas tinggi dan pada kerentanan yang paling umun terjadi seperti *M3 Insecure Communication* dan *M2 Insecure Data Storage*, serta memberikan rekomendasi mitigasi untuk meningkatkan perlindungan data pengguna.

II. METODE

Metode yang digunakan dalam penelitian ini menggunakan metode kualitatif dengan data yang diperoleh melalui studi literatur dan observasi langsung terhadap aplikasi [16]. Sedankan metode analisis keamanan menggunakan metode *static* [18] *vulnerability assessment and penetration testing* (VAPT) dengan mengacu pada standar OWASP *Top 10 Mobile Risk*. Yang dapat dilihat pada gambar 1



Gambar. 1 Kerangka Penelitian

Rincian mengenai gambar 1 dapat dijelaskan sebagai berikut:

A. Kajian Pustaka
Di tahap ini, peneliti mengkaji literatur dan penelitian sebelumnya yang relevan dengan topik, seperti keamanan aplikasi, VAPT, dan metode analisis keamanan.

B. Metode Pengumpulan Data
Tahapan ini dilakukan untuk mengumpulkan data yang diperlukan pada saat penelitian, pengumpulan data dilakukan dengan menggunakan studi pustaka dan observasi

C. Metode Analisis Keamanan
Tahapan analisis keamanan pada penelitian ini menggunakan metode *Static Vulnerability Assessment and Penetration Testing* (VAPT) yang memiliki beberapa tahapan *yaitu Vulnerability Assessment dan Penetration Testing dan Reporting* [2] yaitu:

1. *Vulnerability Assessment*
Pada tahapan ini ada tiga proses yang dilakukan yaitu:
   a. *Information Gathering* dilakukan dengan mengumpulkan informasu terkait aplikasi Identitas Kependudukan Digital
   b. *Scanning* dilakukan dengan menggunakan metode *static* dengan melakukan analisis terhadap *source code* aplikasi dan memindai *file* .apk. Adapun alat bantu atau *tools* yang akan digunakan untuk diantaranya, MOBSF [19] yang berpokus pada *code analysis*.
   c. *Result Analysis* pada tahapan ini hasil dari *Scanning* kemudian dianalisis untuk mengidentifikasi kerentanan yang paling kritis dan relevan. Tahap ini melibatkan peninjauan terhadap data yang dikumpulkan untuk menentukan tingkat risiko yang dihadapi oleh aplikasi
2. *Penetration Testing*
Pada tahapan ini ada tiga proses yang dilakukan yaitu:
   a. *Build Attack* dilakukan dengan merencanakan dan membangun serangan yang spesifik untuk menguji keamanan aplikasi. Tahap ini melibatkan pengembangan eksploitasi atau metode serangan yang akan digunakan untuk mencoba mengeksploitasi kerentanan yang ditemukan. Pada penelitian *build attack* yang akan dilakukan adalah *man in the midle* (MITM) atau *Intercept* pada kerenatana *M3 Insecure Commuication* dengan menggunakan metode *reverse engineering*. dan pada *M2 Insecure Data Storage* akan dilakukan pengecekan pada *file shared Preference* untuk melihat apakah ada data yang terekspos.
   b. *Explotation* pada proses ini dilakukan Pada tahap ini, serangan yang telah direncanakan diterapkan pada aplikasi untuk mencoba mengeksploitasi kerentanan yang ada. Tujuan dari tahap ini adalah untuk melihat apakah kerentanan tersebut benar-benar dapat dieksploitasi
   c. Result analysis
   Hasil dari eksploitasi dianalisis untuk memastikan efektivitas serangan dan kerusakan yang mungkin terjadi. Analisis ini membantu dalam memahami tingkat ancaman yang sebenarnya dan menentukan langkah-langkah perbaikan yang perlu diambil untuk mengamankan aplikasi.

3. Reporting
   Pada tahapan ini akan dijabarkan kembali hasil dari pengujian dengan metode *static vulnerability assessment and penetration testing* (VAPT) serta memberikan rekomendasi perbaikan atau mitigasi dan pembuktian mitigasi.

D. Kesimpulan
Di akhir penelitian, kesimpulan ditarik berdasarkan hasil analisis dan pengujian.

III. HASIL DAN PEMBAHASAN

Pengujian pada penelitian ini diimplementasikan dengan menggunakan metode *Static Vurlnerability Assessement and Penetration Testing* (VAPT) yang memiliki beberapa tahanpan yaitu: vulnerability *assessment (information gathering, scanning, resulty analysis), penetration testing (build attack, exploitation, result analysis)* dan *reporting*.

A. Vulnerability Assessment

Pada tahapan ini ada tiga tahapan yang dilakukan diantaranya:

1. Information Gathering

Pada tahap *information gathering*, dilakukan pengumpulan informasi mengenai aplikasi Identitas Kependudukan Digital yang didapatkan dari hasil analisis statis menggunakan *tools* MOBSF dan alat bantu lainnya yang dapat dilihat dari tabel 1

TABEL. 1
Information Gathering Aplikasi Identitas Kependudukan Digital

Informasi	Keterangan
Nama Aplikasi	Identitas Kependudukan Digital
Nama *Package*	gov.dukcapil.*mobile*_id
Main Activity	gov.dukcapil.*mobile*_id.MainActivity
Versi Aplikasi	1.2.2
Android Version Code	19
Minimal SDK	21
Target SDK	31
Jumlah *Install*	10,000,000+
Developer	DITJEN DUKCAPIL KEMENDAGRI
Developer Website	https://dukcapil.kemendagri.go.id
Developer Email	infosiak@dukcapil.kemendagri.go.id
Certificate/key	MD5,SHA1 dan SHA256

Berdasarkan tabel diatas, menunjukan bahwa aplikasi Identitas Kependudukan Digital dikembangkan oleh Direktorat Jenderal Kependudukan dan Pencatatan Sipil (Ditjen Dukcapil) Kementerian Dalam Negeri Republik Indonesia, bertujuan untuk memudahkan warga negara dalam mengakses dan mengelola identitas kependudukan mereka secara digital.

2. Scanning

Pada tahap ini dilakukan *scanning* terhadap aplikasi Identitas Kependudukan Digital, menggunakan bantuan *tools* MOBSF dengan metode statis, ditemukan beberapa kerentanan pada bagian *code analysis*, dengan tingkat risiko kerentanan tinggi dan kerentanan paling umun terjadi berdasarkan OWASP *Top 10 Mobile Risk* yang dapat dilihat pada tabel 2 dibawah.

TABEL. 2 Isu Kerentanan

No	Isu Kerentanan	Tingkat Risiko
1	*M2 Insecure Data Storage*	*Warning*
2	*M3 Insecure Communication*	*High*
3	*M5: Insufficient Cryptography*	*Warning*
4	*M7: Client Code Quality*	*Warning*

3. Result Analysis

Pada tahap ini, dilakukan analisis terhadap hasil temuan dari tahapan *vulnerability assessment* pada proses *scanning* menggunakan alat MOBSF dengan metode statis pada bagian *code analysis*. Hasilnya menunjukkan adanya kerentanan dengan tingkat risiko tinggi dan kerentanan paling umum terjadi dengan merujuk pada standar OWASP *Top 10 Mobile Risk*, Maka penelitian ini akan memprioritaskan pada kerentanan *M3 Insecure Communication* dengan tingkat kerentanan tinggi (*High*) dan *M2 Insecure Data Storage* kerentanan paling umum terjadi dengan tingkat kerentanan peringatan (*Warning*).

B. Penetration Testing

Pada tahapan ini ada tiga tahapan yang akan dilakukan diantaranya:

1. Build Attack

a. *M3 insecure communication*

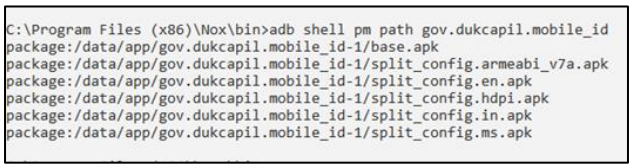
Berdasarkan *result analysis* pada bagian *vulnerability assessment* ditemukan isu kerentanan pada *M3 insecure communication* dengan tingkat risiko *high* berdasarkan OWASP *Top 10 Mobile Risk* yang berkaitan dengan komunikasi antara klien-server ini dapat dieksploitasi dengan melakukan *intercept* pada aplikasi Identitas Kependudukan Digital. Sebelum melakukan *intercept*, proses yang harus dilakukan adalah melakukan *bypass* SSL *pinning* pada aplikasi Identitas Kependudukan Digital yang pada prosesnya akan berkaitan dengan kerentanan M9 Reverse Engineer dan M8 Code Tampering berdasarkan OWASP Top 10 Mobile Risk, karena proses bypass SSL pinning akan dilakukan dengan metode reverse engineer dan akan dilakukan penandatanganan ulang aplikasi Identitas Kependudukan Digital yang termasuk kedalam kerentanan M8 code tampering berdasarkan OWASP Top 10 Mobile Risk.

Gambar. 2 BoringSSL in flutter

Selanjutnya pengujian dengan metode *Reverse Engineer* untuk *bypass* SSL *pinning* yang dijelaskan dibawah ini:

1) Ekstrak Aplikasi

Hasil dari ekstrak aplikasi didapatkan beberapa aplikasi yang terdiri dari Base.apk dan lainnya yang dapat dilihat pada gambar 5



Gambar. 3 Extrak Aplikasi Menggunakan ADB

Berdasarkan gambar 5 diatas, ekstrak aplikasi dengan menggunakan *tools* ADB menunjukan perintah yang dijalankan adalah adb shell pm path gov.dukcapil.*mobile*_id, yang digunakan untuk mengetahui lokasi *file* APK dari aplikasi Identitas Kependudukan Digital yang memiliki nama *package* gov.dukcapil.*mobile*_id. Hasil dari perintah ini menunjukkan beberapa *path* yang mengarah ke *file* APK utama dan *file* konfigurasi terkait yang diinstal di perangkat *Android*.

2) *Merge* Aplikasi

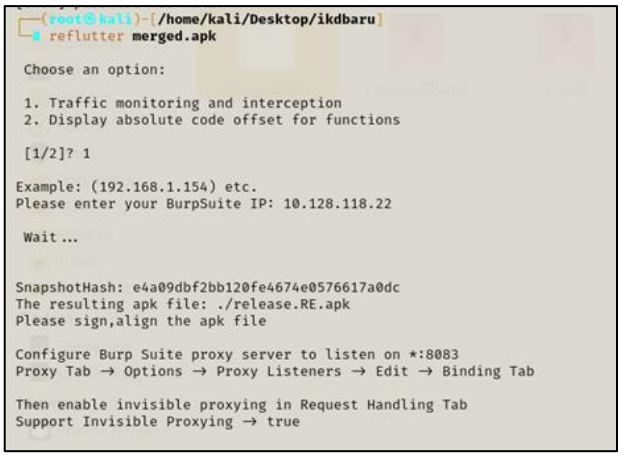
Berdasarkan gambar 6 menunjukan bahwa perintah pada command line menggunakan APK Editor, sebuah alat untuk memodifikasi *file* APK (*Android* Package Kit) yang digunakan untuk menginstal aplikasi di perangkat *Android*.

Gambar. 4 Proses Merge Aplikasi menggunakan apkeditor.jar

Input *file* APK yang akan digabungkan berada di direktori C:\ikd, dan output *file* hasil penggabungan akan disimpan sebagai merged.apk di direktori C:\v3.

3) *Rebuild* Aplikasi

Pada proses ini dilakukan dengan menggunakan tools khusus yaitu Reflutter untuk *reverse engineering* untuk melakukan modifikasi, karena aplikasi Identitas Kependudukan Digital dibangun dengan menggunakan flutter dengan bahasa pemograman dart. Berdasarkan gambar 7 dapat dilihat *command* yang dijalan adalah Reflutter merged.apk penguji memilih opsi pertama, yaitu pemantauan lalu lintas jaringan.



Gambar. 5 Proses Rebuild Menggunakan Reflutter

Penguji diminta untuk memasukkan alamat IP Burp Suite, yang dalam hal ini adalah 10.128.118.22. Setelah IP dimasukkan, Reflutter menghasilkan snapshot hash dan *file* APK hasil analisis disimpan dengan nama release.RE.apk. *File* ini kemudian perlu ditandatangani (*sign*) dan diatur ulang (*align*) agar dapat dipasang dengan benar.

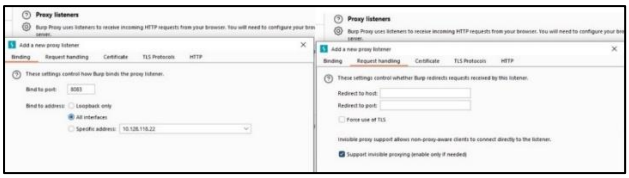
4) *Sign* ulang *file* .apk

Berdasarkan gambar 8 memperlihatkan proses penandatanganan dan verifikasi APK debug menggunakan uber-apk-signer.jar. yang mana proses penandatanganan ulang aplikasi ini termasuk ke kerentanan *M8 code tampering* berdasarkan OWASP *Top 10 Mobile Risk*. Proses ini memastikan APK bisa diinstal dan berjalan dengan baik di perangkat *Android* selama masa pengembangan.

Gambar. 6 Proses Sign ulang aplikasi menggunakan uber apk signer

5) *Setting* Burp Suite

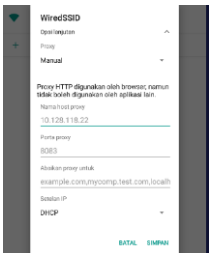
Berdasarkan gambar 9 instruksi untuk mengonfigurasi Burp Suite *Proxy* agar bisa mendengarkan lalu lintas pada port 8083, dengan langkah-langkah konfigurasi melalui tab *Proxy Listeners*. Setelah itu, pengguna diminta mengaktifkan fitur *Invisible Proxying* agar Burp Suite dapat menangkap lalu lintas secara diam-diam tanpa mengganggu koneksi aplikasi.



Gambar. 7 Proses Setting Burp Suite

6) *Install* .apk dan *setting proxy*

Berdasarkan gambar 10 menunjukan proses konfigurasi *proxy* di emulator *Android* NoxPlayer untuk menggunakan alamat IP dan port sesuai dengan intruksi dari Reflutter (10.128.118.22:8083), yang mungkin digunakan untuk memantau lalu lintas jaringan melalui alat seperti Burp Suite.



Gambar. 8 Proses Setting Proxy

7) *Intercept* Burp Suite

Pada proses ini dilakukan *intercept* dengan menggunakan Burp Suite [20] dan Nox Player yang telah terinstal aplikasi Identitas Kependudukan Digital untuk melihat bagaimana aplikasi berkomuikasi dengan *server* yang dapat dilihat pada tahapan *Exploitation*

b. *M2 Insecure Data Storage*

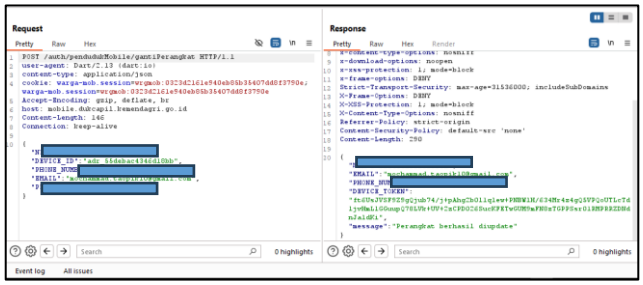
Kerentanan ini mencakup penyimpanan yang tidak aman dan kebocoran data. Ketika suatu aplikasi memiliki kerentanan ini maka penyerang dapat memperoleh informasi pribadi pada suatu aplikasi. Kerentanan ini berupa *compromise file system (database SQL, log files, manifest file, binary data stores, cookie stores, sd card, cloud synced)* [17]. Pada temuan ini pengujian akan berfokus pada data sensitif yang disimpan oleh aplikasi dengan mengidentifikasi file penyimpanan, selain itu akan dilakukan verifikasi terhadap data yang disimpan apakah bocor akibat serangan pada aplikasi. Dengan melakukan identifikasi terhadap penyimpanan internal dan eksternal untuk mendapatkan sumber jenis penyimpanan yang digunakan aplikasi dengan mengecek pada androidManifest.xml. selanjutnya dengan mengecek *shared Preference* pada bagian penyimpanan *file* berupa XML *file* (in /data/data/<package-name>/shared_prefs) untuk *file* sensitif yang dapat dilihat pada tahapan *Exploitation.*

2. *Exploitation*

Berdasarkan tahapan *build attack*, serangan yang telah direncanakan diterapkan pada aplikasi untuk mencoba mengeksploitasi kerentanan yang ada, diantaranya:

a. *M3 Insecure Communication*

Pada tahapan ini dilakukan sesuai dengan tahapan *build attack* yaitu dengan mencoba melakukan *bypass* SSL untuk melakukan *intercept* menggunakan Burp Suite pada aplikasi Identitas Kependudukan Digital, yang dapat dilihat pada gambar 11 dibawah.



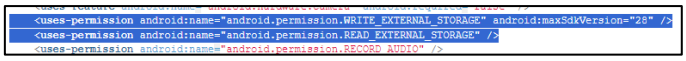
Gambar. 9 Proses Intercept Login

Pada gambar 11 menunjukan proses komunikasi yang terjadi pada aplikasi Identitas Kependudukan Digital dengan menggunakan *intercept* pada Burp Suite terlihat bahwa ada beberapa eksposur informasi sensitif dalam *Request* dan *Response*:

1) Data Sensitif di *Body Request*: Dalam *request*, informasi yang sangat sensitif seperti NIK (Nomor Induk Kependudukan), EMAIL, PHONE_NUMBER, dan PIN diekspos secara langsung. Informasi pribadi ini sangat rentan jika komunikasi tidak dienkripsi, karena dapat disadap oleh pihak ketiga.
2) Data Sensitif di *Response*: *Server* mengembalikan informasi yang sama seperti yang dikirimkan dalam *request*, termasuk NIK dan nomor telepon. Ini memperbesar risiko jika ada celah keamanan.

b. *M2 Insecure Data Storage*

Pada tahap ini dilakukan akan dilakukan identifikasi terhadap penyimpanan internal dan eksternal pada *file AndroidManifest.xml* pada aplikasi Identitas Kependudukan Digital yang dipasang pada Nox Emulator.



Gambar. 10 File AndroidManifest.xml

Berdasarkan Gambar 12, pada *source code* dalam *AndroidManifest.xml* terdapat izin untuk menulis dan membaca pada Penyimpanan Eksternal, yang memungkinkan penyimpanan data *cache*. Hal ini membuat *file* dalam penyimpanan tersebut dapat diakses oleh aplikasi lain di perangkat pengguna. Pada pengujian lanjutan dengan memeriksa *shared preferences* untuk *file* sensitif di *data/data/<package-name>/shared_prefs*, ditemukan file

FlutterSecureStorage.xml di *data/data/gov.dukcapil.mobile_id/shared_prefs* yang dapat diakses dan dibaca pada perangkat pengguna

Gambar. 11 capture file FlutterSecureStorage.xml

Pada gambar. 13 tersebut menunjukkan penggunaan *FlutterSecureStorage* untuk menyimpan data sensitif secara aman di perangkat. Data DEVICE_TOKEN disimpan dalam bentuk terenkripsi, tidak menunjukan adanya data atau informasi yang terekspos atau plain teks menunjukkan bahwa praktik keamanan yang baik dan sesuai untuk melindungi data dari serangan yang melibatkan akses fisik atau remote ke perangkat.

3. *Result Analysis*

Pada tahapan *Exploitation* menunjukan bahwa isu kerentanan yang ditemukan pada tahapan *Vulnerability Assessment* dan dilakukan analisis lebih lanjut pada tahapan *Penetration Testing* menunjukkan bahwa:

a. M3 inscure communication dengan tingkat kerentanan tinggi (*high*)

Berdasarkan OWASP *Top 10 Mobile Risk*, dari hasil tahapan ini dapat dibuktikan bahwa kerentanan *M3 Insecure Communication* tersebut *True Positive* benar adanya. Dimana dalam proses uji coba *login* pada aplikasi Identitas Kependudukan Digital yang telah dilakukan *bypass* SSL menunjukan pada proses komunikasi antara klien-server ada beberapa eksposur data sensitif yaitu *password* atau *pin* yang dikirim pada *body* ditampilkan dalam bentuk *plain text* pengujian ini terdeteksi ketika melakukan *intercept* menggunakan Burp Suite.

b. M2 Insecure Data Storage dengan tingkat kerentanan peringatan (*warning*)

Berdasarkan OWASP *Top 10 Mobile Risk*, dari hasil tahapan ini dapat dibuktikan bahwa kerentanan *M2 Insecure Data Storage* tersebut *False Positive*. Dimana pada proses pengujian menunjukkan tidak adanya data sensitif yang terekspos. Dengan Penggunaan *FlutterSecureStorage* untuk menyimpan data sensitif secara aman di perangkat. Data DEVICE_TOKEN disimpan dalam bentuk terenkripsi, yang menunjukkan praktik keamanan yang baik dan sesuai untuk melindungi data dari serangan yang melibatkan akses fisik atau remote ke perangkat.

C. Reporting

Pada tahapan ini akan dijabarkan kembali hasil dari pengujian dengan metode *static vulnerability assessment and penetration testing* (VAPT) serta memberikan rekomendasi perbaikan atau mitigasi dan pembuktian mitigasi, diantaranya sebagai berikut:

1. Hasil Pengujian

Dari tahapan pengujian yang telah dilakukan dengan menggunakan metode *static vulnerability assessment and penetration testing* (VAPT) pada aplikasi Identitas Kependudukan Digital ditemukan beberapa kerentanan diantaranya dapat dilihat pada table 3 berikut.

Tabel. 3 Hasil Pengujian

No	Isu Kerentanan	*Vulnerability Assessment*	*Penetration Testing*	Status
1	*M2 Insecure Data Storage*	Ya	Tidak	*False Positive*
2	*M3 Insecure Communication*	Ya	Ya	*True Positive*
3	*M8 Code Tampering*	Tidak	Ya	*False Negative*
4	*M9 Reverse Engineering*	Tidak	Ya	*False Negative*

Berdasarkan table. 3 menujukan bahwa isu kerentana *M2 Insecure Data Storage* pada tahapan *Vulneralbility Assessment* terdetaksi adanya isu kerentanan tetapi pada tahapan *Penetration Testing* tidak dapat dibuktikan maka isu kerentanan ini berstatus *False Positive*, isu kerentanan *M3 Insecure Communication* pada tahapan *Vulneralbility Assessment* terdetaksi adanya isu kerentanan dan pada tahapan *Penetration Testing* dapat dibuktikan maka isu kerentanan ini berstatus *True Positive*, Isu kerentanan *M8 Code Tampering* pada tahapan *Vulneralbility Assessment* tidak terdetaksi adanya isu kerentanan tetapi pada tahapan Penetration Testing dapat dibuktikan adanya kerentanan maka isu kerentanan ini berstatus False Negative, dan isu kerentanan M9 Reverse Engineering pada tahapan Vulneralbility Assessment tdiak terdeteksi adanya isu kerentanan tetapi pada tahapan Penetration Testing tidak dapat dibuktikan ini maka isu kerentanan ini berstatus False Positive

2. Mitigasi dan Pembuktian Mitigasi

Dari hasil pengujian yang telah dilakukan pada untuk meningkatkan kualitas keamanan pada aplikasi identitas kependudukan Digital dapat menerapkan beberapa rekomendasi untuk mengatasi kerentanan, diantaranya:

a. M2 Insecure Data Storage

1) Hindari penggunaan local storage untuk menyimpan data sensitif atau informasi karena memungkinkan untuk rentan terhadap serangan *cross-site scripting* (XSS)

2) Gunakan enkripsi jika harus menyimpan data pada *local storage*

b. M3 Insecure Communication

1) Gunakan Protokol Enkripsi yang Kuat Pastikan semua data yang dikirim melalui jaringan dienkripsi dengan protokol yang kuat, seperti TLS (*Transport Layer Security*) versi terbaru (TLS 1.2 atau 1.3).
2) Hindari Pengiriman Data Sensitif melalui Koneksi yang Tidak Aman
3) Pastikan data dikirim di dalam *body message* tidak dalam bentuk *plain text*.

c. M8 Code Tampering

1) Implementasi *obfuscate* kode aplikasi Flutter. Proses ini mengacak nama variabel, metode, dan kelas sehingga lebih sulit dipahami oleh orang yang mencoba untuk membongkar aplikasi.
2) Cegah aplikasi dijalankan pada perangkat yang sudah di-*root* atau di-*jailbreak*

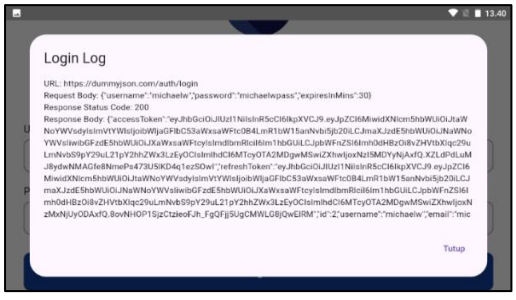
d) *M9 Reverse Engineering*

1) *Obfuscation* mengacak kode sumber setelah dikompilasi untuk mempersulit pembacaannya. Nama variabel, kelas, dan metode diacak menjadi karakter acak yang tidak mudah dimengerti, sehingga membuat reverse engineering lebih sulit.
2) *Reverse engineering* sering dilakukan di perangkat yang telah di-root atau di-jailbreak. Implementasikan deteksi *jailbreak* (iOS) atau *root* (*Android*) untuk mencegah aplikasi berjalan di perangkat yang telah di-*root*.

Dari beberapa mitigasi yang telah disebutkan diatas, penguji akan melakukan beberapa pembuktian dari rekomendasi mitigasi tersebut diantanya:

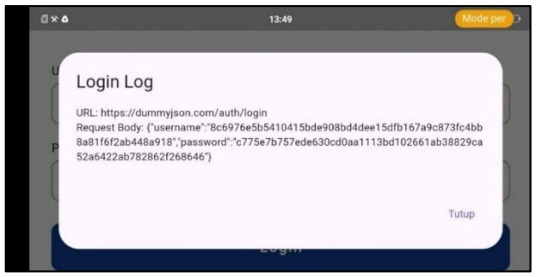
a) Pembuktian Mitigasi *M3 Insecure Communication*

Pembuktian mitigasi untuk menutup celah kerentanan yang sama terhadap aplikasi Identitas Kependudukan Digital yaitu pada bagian *text body message* menampilkan data atau infromasi yang terekspos yang dapat dilihat pada gambar dibawah ini.



Gambar. 12 Prototype Login 1

Pada gambar 14 menunjukan menu login prototye ini menggunakan server *dummy public* yaitu https://dummyjson.com/auth/login yang diterapkan pada *prototype* untuk melakukan *request login* yang menunjukan bahwa data yang dikirim pada *body message* berupa *plain text.*

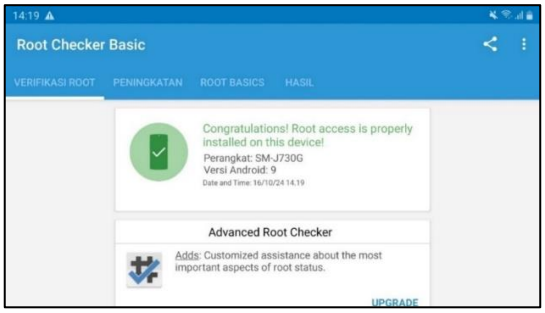


Gambar. 13 Prototype Login 2

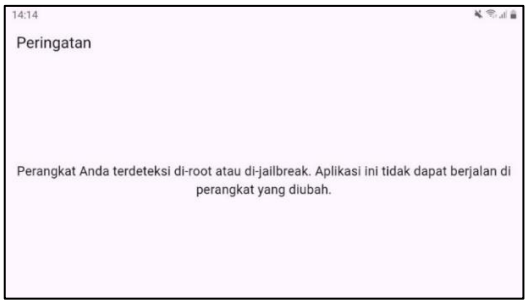
Berdasarkan gambar 15 diatas menunjukan perubahan *body message* yang dilakukan pengembang untuk menutupi dengan menabahkan *hashing* pada *username* dan *password* pengguna, hal ini dapat dipastikan bahwa jika *username* dan *password* pengguna dalam keadaan aman dan tidak terekspos secara *plain text*.

b) Pembuktian Mitigasi *M9 Reverse Engineering*

projek *prototype* dapat berjalan pada perangkat yang sudah *diroot*.



Gambar. 14 Root detection 3



Gambar. 15 Root detection 4

Pada gambar 16 menunjukan bahwa perangkat yang digunakan untuk menjalankan projek *prototype* telah di *root* dan pada gambar 17 mencoba menjalankan projek *prototype* tidak dapat digunakan dan menampilkan pemberitahuan bahwa perangkat telah *diroot* atau *jailbreak*.

IV. SIMPULAN

A. Simpulan

Berdasarkan analisis keamanan yang telah dilakukan pada bab sebelumnya mengenai "Analisis Keamanan Analisis Keamanan Pada Aplikasi *Android* Menggunakan Metode Static *Vulnerability Assessment And Penetration Testing* (VAPT) Studi Kasus: Aplikasi Identitas Kependudukan Digital ", penulis dapat menyampaikan beberapa kesimpulan diantanya:

1. Berdasarkan pada hasil penelitian, teridentifikasi bahwa ada beberapa kerentanan yang ditemukan pada aplikasi Identitas Kependudukan Digital berdasarkan *OWASP Top 10 Mobile Risk* yaitu *M2 Insecure Data Storage*, *M3 Insecure Communication*, *M8 Code Tampering dan M9 Reverse Engineering*
2. Dengan melakukan analisis keamanan menggunakan metode *static vulnerability assessment and penetration testing* (VAPT) menunjukkan bahwa kerentanan yang didapatkan pada proses *static vulnerability assessment* harus divalidasi kembali dengan proses *Penetration testing* untuk memastikan bahwa kerentanan tersebut benar (*true postif*) atau tidak (*false negatif*)
3. Berdasarkan hasil *reporting* didapatkan rekomendasi untuk menutup celah keamanan pada *M3 Insecure Communication* dengan menerapkan *hashing* pada body message untuk mengamankan komunikasi antara klien-server dan pada M9 Reverse Engineering dengan menerapkan *root* atau *jailbreak detection* dapat menghambat proses *reverse engineering* atau peretasan secara ilegal
4. Pada tahapan pembuktian mitigasi menunjukkan bahwa mitigasi keamanan yang disebutkan pada tahap *reporting* dapat dibuktikan dengan menggunakan projek prototype *M3 insecure Communication* menerapkan *hashing* pada *body message* dan *M9 reverse engineer* menerapkan *root detection*.

REFERENSI

[1] Lukman, J. P., & Sakir, A. R. (2024). Transformasi Digital dalam Administrasi Publik: Peluang dan Tantangan. MULTIPLE: Journal of Global and Multidisciplinary, 2(1), 1042-1049.

[2] Shinde, P. S., & Ardhapurkar, S. B. (2016, February). Cyber security analysis using vulnerability assessment and penetration testing. In *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)* (pp. 1-5). IEEE.

[3] Play, G. (2024). Google Play Store - Download Free Android Apps, Googleplay. Retrieved

[4] Permadi, I. B., & Rokhman, A. (2023). Implementasi Identitas Kependudukan Digital Dalam Upaya Pengamanan Data Pribadi. *JOPPAS: Journal of Public Policy and Administration Silampari*, *4*(2), 80-88.

[5] Gabriel, A. (2024, August). PERLINDUNGAN HUKUM ATAS DATA PRIBADI DALAM KASUS KEBOCORAN DATA PUSAT DATA NASIONAL SEMENTARA (PDNS) DALAM PERSPEKTIF HUKUM PIDANA. In *Seminar Nasional-Hukum dan Pancasila* (Vol. 3, No. 3, pp. 18-26).

[6] Putra, A. D., Santoso, J. D., & Ardiansyah, I. (2022). Analisis Malicious Software Trojan Downloader Pada Android Menggunakan Teknik Reverse Engineering (Studi Kasus: Kamus Kesehatan v2. apk). *Building of Informatics, Technology and Science (BITS)*, *4*(1), 69-79.

[7] Kartono, A., Sularsa, A., & Ismail, S. J. I. (2019). Membangun Sistem Pengujian Keamanan Aplikasi Android Menggunakan Mobsf. *eProceedings of Applied Science*, *5*(1).

[8] Priambodo, D. F., Hasbi, M., & Malacca, M. S. (2022). Security Assessment Aplikasi Mobile E-Kinerja dengan Acuan OWASP Top 10 Mobile Risks. *JEPIN (Jurnal Edukasi dan Penelit. Inform., vol. 8, no. 3, pp. 560—571*.

[9] Arditaa, I. K. A. O., Putraa, I. G. N. A. C., Kustiadie, M. R., Varuna, I. G. N. M. D., & Prananda, M. Y. E. (2022). Analisis Keamanan Aplikasi Android Dengan Metode Vulnerability Assessment. *Jurnal Elektronik Ilmu Komputer Udayana p-ISSN*, *2301*, 5373.

[10] Tansen, E., & Nurdiarto, D. W. (2020). Analisis dan Deteksi Malware dengan Metode Hybrid Analysis Menggunakan Framework MOBSF. *Jurnal Teknologi Informasi*, *4*(2), 191-201.

[11] Anwar, C., Hady, S., Rahayu, N., & Kraugusteeliana, K. (2023). The Application of Mobile Security Framework (MOBSF) and Mobile Application Security Testing Guide to Ensure the Security in Mobile Commerce Applications. *Jurnal Sistim Informasi dan Teknologi*, 97-102.

[12] Archibong, E. E., Stephen, B. U. A., & Asuquo, P. (2024). Analysis of Cybersecurity Vulnerabilities in Mobile Payment Applications. *Archives of Advanced Engineering Science*, 1-12.

[13] Kurniawan, C., & Trianto, N. (2021). Security Assessment pada Aplikasi Mobile Android XYZ dengan Mengacu pada Kerentanan OWASP Mobile Top Ten 2016. *Info Kripto*, *15*(1), 11-18.

[14] Nurindahsari, F., & Zen, B. P. (2022). Analisis Statik Keamanan Aplikasi Video Streaming Berbasis Android Menggunakan Mobile Security Framework (Mobsf). *Cyber Security dan Forensik Digital*, *4*(2), 63-80.

[15] Antonishyn, M. (2020). Mobile applications vulnerabilities testing model. *Collection" Information Technology and Security"*, *8*(1), 49-57.

[16] Pratama, A. D. (2021). Uji Keamanan Aplikasi ABC Milik Instansi XYZ Menggunakan OWASP Mobile Security Testing Guide. *Info Kripto*, *15*(3), 113-121.

[17] Chandra, R., Turnip, T. N., Sinambela, E. S., Panjaitan, G. H. A., Tampubolon, V. N., Turnip, A. M., & Siahaan, D. (2023, November). Mobile Application Security in the Health Care and Finance Sector with Static Analysis (Tools: MobSF). In *2023 29th International Conference on Telecommunications (ICT)* (pp. 1-7). IEEE.

[18] Hanifurohman, C., & Hutagalung, D. D. (2020). Analisis Statis Menggunakan Mobile Security Framework Untuk Pengujian Keamanan Aplikasi Mobile E-Commerce Berbasis Android. *Sebatik*, *24*(1), 22-28.

[19] Yankson, B., Hung, P. C., Iqbal, F., & Ali, L. (2021, April). Security assessment for Zenbo robot using Drozer and mobSF frameworks. In *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)* (pp. 1-7). IEEE.

[20] Zhang, L., Wang, B., Shen, Q., Song, Y., Guo, N., & Xie, L. (2021, April). A MITM Based Penetration Test Efficiency Improvement Approach for Traffic-Encrypted Mobile Applications of Power Industry. In *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)* (pp. 743-747). IEEE.