Uji Keamanan Aplikasi ABC Milik Instansi XYZ Menggunakan OWASP *Mobile Security Testing Guide*

Aldino Dika Pratama[1)], Amiruddin Amiruddin[2)]

(1) Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara, aldino.dika@student.poltekssn.ac.id

(2) Rekayasa Keamanan Siber, Politeknik Siber dan Sandi Negara, amiruddin@student.poltekssn.ac.id

Abstrak

Aplikasi ABC adalah sebuah aplikasi mobile berbasis Android yang digunakan oleh instansi XYZ. Aplikasi tersebut digunakan untuk layanan pengajuan daring pada bidang XXX. Seiring meningkatnya popularitas aplikasi berbasis Android, meningkat pula masalah keamanan yang perlu diperhatikan, seperti insecure data storage yang merupakan kerawanan tertinggi. Pada penelitian ini, dilakukan uji keamanan aplikasi mobile berbasis Android yang disebut ABC berdasarkan OWASP Mobile Security Testing Guide. Pengujian dilakukan dengan pendekatan grey box pada cakupan fokus area Data Storage on Android dan Local Authentication on Android di level 1. Pengujian terdiri dari delapan bagian pengujian. Berdasarkan hasil dari delapan pengujian yang telah dilakukan, ditemukan tiga kerentanan pada bagian Testing Local Storage for Sensitive Data, Testing Local Storage for Input Validation, dan Determining Whether the Keyboard Cache Is Disabled for Text Input Fields. Rekomendasi perbaikan yaitu memberikan pengamanan enkripsi di file shared preferences, melakukan validasi input pada akun pengguna, dan aplikasi tidak menyimpan cache.

Kata kunci: Android (1), *mobile* (2), OWASP MSTG (3), Uji Keamanan (4)

Abstract

ABC application is an Android-based mobile application used by XYZ agencie. The application is used for online submission services in the XXX field. Along with the increasing popularity of Android-based applications, there are also security issues that need attention, such as insecure data storage which is the highest vulnerability. In this study, a security test for an Android-based mobile application called ABC was conducted based on the OWASP Mobile Security Testing Guide. The test is carried out using a gray box approach in the focus area of Data Storage on Android and Local Authentication on Android at level 1. The test consists of eight test sections. Based on the results of the eight tests that have been carried out, three vulnerabilities were found in the Testing Local Storage for Sensitive Data, Testing Local Storage for Input Validation, and Determining Whether the Keyboard Cache Is Disabled for Text Input Fields. Recommendations for improvement are to provide encryption security in shared preferences files, validate input on user accounts, and application does not save cache.

Keywords: Android (1), *mobile* (2), OWASP MSTG (3), *Security Testing* (4)

1. PENDAHULUAN

Perkembangan perangkat *mobile* saat ini meningkat dengan pesat. Berdasarkan data App Annie pada tahun 2017, dijelaskan bahwa penduduk Indonesia menjadi salah satu pengguna aplikasi *mobile* paling aktif di dunia [1]. Beberapa aplikasi Android menyimpan data pribadi yang bersifat krusial milik pengguna seperti pada aplikasi perbankan, *e-commerce*, sosial media, dan lain sebagainya [2].

Menurut laporan dari securitytoday.com yang berjudul *Insecure Data Storage in Mobile Application Poses Security Issues 2019*, disebutkan bahwa terdapat kerentanan berisiko tinggi sebanyak 43% pada aplikasi Android [3]. Kerentanan yang sering dijumpai pada aplikasi *mobile* adalah *insecure data storage* pada laporan tersebut mencapai 76%. Kelemahan mekanisme keamanan sejak pembuatan aplikasi menjadi penyebab sebagian besar masalah tersebut dapat terjadi [3].

Kerentanan yang terdapat pada aplikasi Android dapat menjadi celah keamanan yang dapat disalahgunakan oleh penyerang dalam menyusup dan mengeksploitasi informasi sensitif pada aplikasi Android. Kerentanan pada aplikasi Android dapat diketahui dengan melakukan uji keamanan agar membantu pengembang dalam meningkatkan keamanan aplikasi dan mengurangi kemungkinan terjadinya eksploitasi [4] [2].

Pemanfaatan aplikasi *mobile* juga terjadi pada instansi XYZ. Instansi XYZ merupakan instansi yang melakukan urusan di bidang XXX. Aplikasi ABC digunakan sebagai pengelola pelayanan daring yang memuat fitur layanan pengajuan XXX. Apabila data tersebut berhasil dicuri oleh pihak yang tidak bertanggung jawab maka akan membahayakan pengguna aplikasi. Oleh karena itu diperlukan adanya langkah untuk mengetahui kerentanan dari aplikasi ABC.

Informasi yang didapatkan dari salah satu pegawai pada instansi XYZ bahwa aplikasi ABC yang diluncurkan ke masyarakat pada bulan November tahun 2018 belum pernah dilakukan uji keamanan. Berdasarkan permasalahan tersebut, langkah yang

dapat dilakukan untuk mengetahui kerentanan pada aplikasi ABC adalah dengan melakukan uji keamanan.

Pada penelitian ini dilakukan uji keamanan aplikasi ABC menggunakan *Open Web Application Security Project (OWASP) Mobile Security Testing Guide.* Area yang diuji pada penelitian ini adalah *Data Storage on Android* dan *Local Authentication on Android* di level 1. Pada area tersebut ada delapan bagian pengujian, yaitu *Testing Local Storage for Sensitive Data, Testing Local Storage for Input Validation, Testing Logs for Sensitive Data, Determining Wheter Sensitive Data is Sent to Third Parties, Determining the Keyboard Cache is Disable for Text Input Fields, Determining Whether Sensitive Stored Data Has Been Exposed via IPC Mechanisms, Checking for Sensitive Data Disclosure Through the User Interface,* dan *Testing Confirm Credentials* [5]. Hasil pengujian ini diharapkan dapat menunjukkan kerentanan, dampak yang ditimbulkan dari kerentanan tersebut, dan dapat memberikan rekomendasi perbaikan keamanan terhadap aplikasi *mobile* ABC. Berdasarkan rekomendasi keamanan tersebut dapat digunakan sebagai acuan perbaikan bagi instansi XYZ dalam meningkatkan keamanan aplikasi *mobile* ABC.

2. LANDASAN TEORI

2.1. OWASP *Mobile Security Testing Guide*

Kemajuan teknologi memberikan masalah keamanan yang baru. Banyak masalah keamanan yang terdapat pada aplikasi *mobile* seperti masalah penyimpanan data, komunikasi antar aplikasi, penggunaan API kriptografi, dan komunikasi jaringan yang tidak aman. Pada tahun 2018, OWASP membuat panduan tentang pengujian keamanan pada aplikasi *mobile* yang dikemas dalam OWASP *Mobile Security Testing Guide*. OWASP *Mobile Security Testing Guide* merupakan standar keamanan yang digunakan pada aplikasi *mobile* disertai dengan panduan pengujian komprehensif yang mencakup proses, teknik, alat, dan studi kasus terkait pelaksanaan uji keamanan aplikasi *mobile* [5]. Pada penelitian ini, uji keamanan aplikasi ABC dilakukan mengikuti semua tahapan OWASP *Mobile Security Testing Guide.* Berikut tahapan pada OWASP *Mobile Security Testing Guide* [5], yaitu *preparation, intelligence gathering, mapping the application, exploitation,* dan *reporting.*

Kegiatan pengujian aplikasi *mobile* berdasarkan OWASP *Mobile Security Testing Guide* terdapat dua cara, yaitu menggunakan analisis statis dan analisis dinamis. Analisis statis dilakukan dengan cara mengidentifikasi aplikasi tanpa menjalankan aplikasi tersebut, seperti melakukan analisis hasil *source code* dan hasil *reverse engineering*. Analisis dinamis dilakukan dengan memeriksa aplikasi ketika dijalankan pada emulator atau perangkat yang asli. Pada OWASP *Mobile Security Testing Guide* terdapat delapan area pengujian, yaitu *Data Storage on Android, Android Cryptographic APIs, Local Authentication on Android, Android Network APIs, Android Platform APIs, Code Quality and Build Settings for Android Apps, Tampering and Reverse Engineering on Android, dan Android Anti-Reversing Defenses* [5]**.**

2.2. Aplikasi Android

Android merupakan *platform open source* berbasis linux yang dikembangkan oleh Google sebagai sistem operasi pada *mobile device*. Saat ini platform Android banyak digunakan untuk berbagai teknologi modern, seperti *mobile device, tablet, smart television*, dan *smart device* lainnya [5]. Level terendah Android dibangun pada *kernel* linux, diatasnya terdapat *Hardware Abstraction Layer* (HAL). HAL digunakan untuk menghubungkan komponen *hardware* dengan *kernel* linux. Aplikasi Android biasanya menggunakan bahasa java dengan dukungan *Android Software Development Kit* (SDK), setelah itu dilakukan kompilasi ke Dalvik *bytecode*. Dalvik *bytecode* dibuat dengan cara melakukan kompilasi kode Java ke dalam direktori dengan ekstensi .class, setelah itu mengkonversi *Java Virtual Machine* (JVM) *bytecode* ke dalam *file* Dalvik dengan ekstensi .dex. Program ekstensi .dex dan *file* lainnya (gambar, XML) kemudian di*pack* menjadi *file* APK [6].

3. METODE PENELITIAN

Penelitian ini dilakukan dengan menggunakan metode kualitatif. Metode kualitatif digunakan untuk mengumpulkan data yang selanjutnya dianalisis dengan didukung oleh teori dan konsep terkait uji keamanan aplikasi *mobile* melalui berbagai sumber seperti buku, jurnal, *paper*, *report*, serta laman internet [7]. Selanjutnya dilakukan teknik analisis data berdasarkan OWASP *Mobile Security Testing Guide* yang meliputi lima tahapan yaitu *Preparation, Intelligence Gathering, Mapping the Application, Exploitation,* dan *Reporting* [5].

3.1. Teknik Pengumpulan Data

Teknik yang digunakan pada pengumpulan data meliputi telaah dokumen dan telaah kepustakaan. Telaah dokumen berguna untuk mendapatkan informasi tentang lokus penelitian, berupa mengumpulkan dan menganalisis dokumen peraturan tentang lokus penelitian. Telaah kepustakaan berguna untuk mendapatkan informasi tentang uji keamanan aplikasi *mobile* Android dari berbagai sumber seperti buku, jurnal, *internet browser*, dan media lain.

3.2. Teknik Analisis Data

Teknik analisis data digunakan sebagai proses untuk mencari pola yang berkaitan dengan proses

pengujian pada objek penelitian. Pada penelitian ini digunakan teknik analisis data menggunakan analisis statis dan analisis dinamis berdasarkan OWASP *Mobile Security Testing Guide*. Analisis statis dilakukan dengan cara mengidentifikasi aplikasi tanpa menjalankan aplikasi tersebut, seperti melakukan analisis hasil *source code* dan hasil *reverse engineering*. Analisis dinamis dilakukan dengan memeriksa aplikasi ketika dijalankan pada emulator atau perangkat yang asli. Analisis dinamis bertujuan untuk melakukan pengujian dan mengetahui informasi yang tidak bisa dilakukan pada analisis statis.

Pada penelitian ini dilakukan lima tahapan pengujian berdasarkan OWASP *Mobile Security Testing Guide* yaitu *Preparation*, *Intelligence Gathering*, *Mapping the Application*, *Exploitation*, dan *Reporting* [5].

a. *Preparation*
   Pada tahapan ini bertujuan untuk menentukan cakupan pada OWASP *Mobile Security Testing Guide*, ditentukan pendekatan yang dilakukan menggunakan *grey box*, selanjutnya metodologi yang digunakan adalah kualitatif meliputi teknik pengumpulan data dan teknik analisis data. Selain itu dilakukan perizinan kepada pihak lokus dalam melakukan pengujian.
b. *Intelligence Gathering*
   Pada tahap ini dilakukan pengumpulan informasi terkait *environmental* (lingkungan) dan *architectural* (arsitektur) target. Tahapan ini berfungsi untuk mengumpulkan informasi sebanyak-banyaknya dari aplikasi untuk digunakan pada tahapan *exploitation*. Pada tahap ini pengumpulan terkait informasi target menggunakan *tool scanning.* Selain itu juga dilakukan pengumpulan informasi terkait *environmental* (lingkungan) dan *architectural* (arsitektur) dengan melakukan wawancara terhadap lokus.
c. *Mapping the Application*
   Tahapan untuk melakukan pemetaan dan penilaian kerentanan yang didapatkan dari *vulnerability assessment*. Pemetaan dilakukan berdasarkan kerentanan yang didapatkan. Kerentanan juga dilakukan penilaian agar penguji memprioritaskan dalam pengujian. Selanjutnya digunakan *scanner* untuk melakukan pemindaian kerentanan menggunakan aplikasi *scanner* dengan dilakukan teknik analisis statis otomatis.
d. *Exploitation*
   Sebagai tahapan pengujian kerentanan yang telah teridentifikasi dari tahap *examination* untuk membuktikan bahwa kerentanan tersebut telah valid. Tahap ini dilakukan dengan teknik analisis dinamis karena pengujian dilakukan dengan menjalankan aplikasi pada perangkat yang telah terkondisikan [5]. *Tool* yang digunakan yaitu Nox Emulator, Burp Suite, Drozer, Android Debug Bridge, dan Wireshark.
e. *Reporting*
   Reporting sebagai tahap terakhir pada pengujian aplikasi *mobile*. Tahap ini dilakukan pelaporan kepada pemilik sistem, berdasarkan hasil dari pengujian kerentanan dan rekomendasi perbaikan keamanan yang difokuskan pada aspek teknis untuk menjaga aplikasi dari *attacker*.

3.3. Objek dan Lokus Penelitian

Objek pada penelitian ini adalah aplikasi ABC. Pada penelitian ini telah dilakukan uji keamanan terhadap aplikasi *mobile* ABC berbasis Android berdasarkan panduan OWASP *Mobile Security Testing Guide*. Aplikasi yang akan diuji adalah aplikasi ABC milik instansi XYZ. Aplikasi ABC merupakan aplikasi yang mendukung layanan secara daring bagi masyarakat terkait bidang XXX. Sedangkan lokus pada penelitian ini adalah instansi XYZ sebagai pengelola bidang XXX melalui aplikasi ABC.

4. HASIL DAN PEMBAHASAN

4.1. *Preparation*

a. Profil Instansi XYZ
   XYZ merupakan sebuah instansi yang berfungsi sebagai unsur pelaksana yang menjadi kewenangan di daerah memiliki tugas membantu pimpinan dalam melaksanakan urusan XXX yang menjadi kewenangan daerah dan tugas perbantuan lain. Berdasarkan Peraturan No. XXX Tahun XXX tentang Kedudukan, Susunan Organisasi, Tugas dan Fungsi, serta Tata Kerja pada instansi XYZ, kedudukan instansi XYZ adalah sebagai instansi yang memiliki tugas pokok melaksanakan urusan di bidang XXX. Oleh karena itu, pada penelitian ini dilakukan pengujian keamanan pada aplikasi ABC berkaitan dengan instansi XYZ yang menyelenggarakan layanan XXX.
b. Aplikasi *Mobile* ABC Berbasis Android
   Aplikasi *mobile* Android ABC sebagai aplikasi layanan bidang XXX. Aplikasi ini digunakan untuk mempermudah dalam pengajuan layanan bidang XXX di suatu daerah.
c. Ruang Lingkup dan Pendekatan
   Pengujian yang dilakukan menggunakan OWASP *Mobile Security Testing Guide*. Peneliti melakukan pengujian pada ruang lingkup *Data Storage on Android* dan *Local Authentication on Android* di level 1 dengan delapan bagian pengujian. Pengujian dilakukan dengan menggunakan pendekatan *grey box.* Pendekatan *grey box* merupakan jenis pendekatan dimana penguji belum memiliki informasi secara keseluruhan terhadap objek yang diuji.
d. OWASP MASVS
   OWASP MASVS digunakan sebagai *requirements* berupa *checklist* yang dipakai pada

saat dilakukan *test case* menggunakan OWASP MSTG. OWASP MASVS mendefinisikan model keamanan aplikasi *mobile* dan menampilkan persyaratan umum untuk aplikasi *mobile*. OWASP MASVS dapat digunakan oleh penguji, *developers*, dan konsumen dalam mengetahui kualitas aplikasi *mobile* yang aman. Terdapat dua level pada *checklist* OWASP MASVS, yaitu L1 dan L2. Pengujian berdasarkan L1 dijadikan sebagai standar minimal keamanan suatu aplikasi, sedangkan L2 berupa standar keamanan pada suatu aplikasi secara lebih mendalam [5]. Pada pengujian ini dilakukan pengujian aplikasi ABC di L1 untuk mengetahui persyaratan minimal keamanan pada aplikasi.

4.2. *Intelligence Gathering*

Pada tahap ini dilakukan pengumpulan informasi terkait *environmental* (lingkungan) dan *architectural* (arsitektur) dari target. Tahapan ini berfungsi untuk mengumpulkan informasi awal agar mendapatkan pemahaman dari aplikasi ABC untuk digunakan pada tahapan *exploitation*. Informasi yang digunakan pada tahap *exploitation* seperti nama aplikasi, *package name, main activity*, versi aplikasi, minimal versi Android, dan lain sebagainya. Pengumpulan informasi tersebut dilakukan dengan menggunakan bantuan *tool Mobile Security Framework* (MobSF). Penggunaan *tool* MobSF pada aplikasi ABC dilakukan dengan cara meng-*upload* APK ABC ke *environment tool* MobSF yang sudah disiapkan. Hasil dari *tool* MobSF ditunjukkan pada **Error! Reference source not found.** 1. Pada tahap ini juga dilakukan pengumpulan informasi *environmental* (lingkungan) dan *architectural* (arsitektur) target dengan melakukan wawancara terhadap lokus.

Tabel 1. Informasi Aplikasi ABC berdasarkan MobSF

Info	Keterangan
Nama aplikasi	Aplikasi ABC
Package Name	id.go.xxx.
Main Activity	id.go.xxx.SplashActivity
Versi Aplikasi	1.2.1
Target SDK	29
Minimal SDK	19
Minimal Versi Android	Android KitKat 4.4
Android Version Code	5
Install	50.000
Developer	XYZ
Developer website	https://xyz.xxx
Developer email	xyz@gmail.com
Certificate/Key	SHA-256
Alamat IP	103.94.xxx.xxx
Lokasi server	Jakarta, Indonesia
Lattitude	-6.21462
Longtitude	106.845131
Deskripsi	Pelayanan daring bagi masyarakat di bidang XXX pada instansi XYZ

Tahap *environmental information*, bahwa Aplikasi ABC milik instansi XYZ ini dikelola oleh Bidang WWW, sedangkan Bidang VVV hanya sebagai pemrakarsa inovasi pada aplikasi ABC. Data yang di-*input*-kan oleh pengguna dalam hal ini masyarakat selanjutnya diproses oleh Unit Pelaksana Teknis Daerah (UPTD) tergantung jenis pelayanan dan alamat pemohon. *Input* data dari masyarakat diproses pertama kali oleh *front office*, jika lolos atau telah memenuhi syarat kemudian akan dilanjutkan oleh operator masing-masing jenis pelayanan, seperti pendaftaran XXX, dan lain sebagainya sampai selesai percetakan. Setelah siap diambil, dokumen kemudian diberikan kepada bagian pengambilan untuk ditandai bahwa dokumen tersebut siap diambil oleh pemohon. Semua data terkait bidang XXX ada di *server* instansi XYZ. Pembuatan aplikasi tersebut pada awalnya diberikan oleh Pemerintah Provinsi ZYX, setelah itu dikembangkan oleh instansi XYZ dalam bentuk aplikasi *mobile* Android ABC.

Tahap *architectural information*,versi Android saat ini melakukan eksekusi Dalvik *bytecode* pada *Android runtime* (ART). Dilakukan kompilasi *Ahead-of-time* (AOT) untuk meningkatkan kinerja ART. Kompilasi AOT dilakukan pada saat sebelum aplikasi dijalankan untuk pertama kalinya. Kode *Precompiled Machine* yang telah dikompilasi akan dipakai di setiap eksekusi aplikasi. Sehingga dengan adanya AOT maka kinerja perangkat akan terus meningkat dan dapat menghemat konsumsi daya. Aplikasi Android tidak memiliki akses secara langsung terhadap *resource hardware*, sehingga memerlukan *sandbox* agar dapat bekerja masing-masing. Sehingga aplikasi dan perangkat tersebut dapat terkontrol, misalnya ketika ada kerusakan atau *error* tidak mempengaruhi terhadap aplikasi lain meskipun mengakses *resource* dan *hardware* yang sama. Dalam melakukan komunikasi dengan Sistem Operasi, aplikasi Android berinteraksi dengan layanan sistem melalui Android *Framework*. *Framework* merupakan lapisan abstraksi yang memiliki Java API dengan level yang tinggi. Aplikasi ABC menggunakan jenis *framework* Codelgniter (CI), kemudian aplikasi ABC ini menggunakan bahasa pemograman PHP. Selain itu untuk penyimpanan *database* menggunakan mysql. Aplikasi ABC hanya dapat dijalankan oleh perangkat dengan minimal versi Android KitKat 4.4 atau di atasnya. Ketika komunikasi data aplikasi ABC menggunakan pengamanan *Transport Layer Security* (TLS) berupa *Secure Socket Layer* V2, sedangkan *certificate/key* yang digunakan adalah algoritma SHA-256.

4.3. *Mapping the Application*

Tahapan ini adalah untuk memetakan dan melakukan penilaian kerentanan berdasarkan hasil *vulnerability assessment*. *Vulnerability assessment* digunakan untuk mengetahui kerentanan apa saja pada aplikasi tersebut. Selanjutnya dilakukan penilaian agar penguji dapat memprioritaskan dalam melakukan pengujian. Pengkategorisasian aplikasi ABC dijelaskan berdasarkan *tool Mobile Security Framework* (MobSF). Kategorisasi kerentanan yang

didapatkan berdasarkan OWASP MASVS pada cakupan *Data Storage on Android* dan *Local Authentication on Android* di level 1 seperti pada Tabel 2.

Tabel 2. Kategorisasi kerentanan hasil *scanning*

No	Jenis Pengujian	Temuan Kerentanan
1	*Testing Local Storage for Sensitive Data* (MSTG-STORAGE-1 and MSTG-STORAGE-2)	Ada kerentanan
2	*Testing Local Storage for Input Validation* (MSTG-PLATFORM-2)	Tidak ada kerentanan
3	*Testing Logs for Sensitive Data* (MSTG-STORAGE-3)	Tidak ada kerentanan
4	*Determining Whether Sensitive Data is Sent to Third Parties* (MSTG-STORAGE-4)	Tidak ada kerentanan
5	*Determining Whether the Keyboard Cache Is Disabled for Text Input Fields* (MSTG-STORAGE-5)	Tidak ada kerentanan
6	*Determining Whether Sensitive Stored Data Has Been Exposed via IPC Mechanisms* (MSTG-STORAGE-6)	Tidak ada kerentanan
7	*Determining Whether Sensitive Stored Data Has Been Exposed via IPC Mechanisms* (MSTG-STORAGE-6)	Tidak ada kerentanan
8	*Testing Confirm Credentials* (MSTG-AUTH-1)	Tidak ada kerentanan

Berdasarkan hasil yang telah diperoleh, ternyata ditemukan kerentanan pada aplikasi ABC di OWASP-MSTG-STORAGE-2 yang termasuk dalam *Data Storage on Android* di bagian *Testing Local Storage for Sensitive Data* (MSTG-STORAGE-1 and MSTG-STORAGE-2). Kerentanan yang ditemukan berupa OWASP-MSTG-STORAGE-2 keterkaitannya dengan aplikasi ABC yaitu aplikasi memungkinkan untuk *read*/*write* ke penyimpanan eksternal. Selain itu aplikasi ABC dapat membuat temp *file* untuk menyimpan informasi, dikhawatirkan informasi yang disimpan bersifat sensitif yang seharusnya informasi bersifat sensitif tidak boleh ditulis ke dalam *file* sementara (temp).

4.4. *Exploitation*

Pada tahap ini dilakukan pengujian kerentanan berdasarkan tahapan yang telah dilakukan sebelumnya, tujuannya untuk membuktikan bahwa kerentanan tersebut telah valid. Tahap ini dilakukan dengan teknik analisis statis dan dinamis karena pengujian dilakukan dengan menjalankan aplikasi pada perangkat yang telah terkondisikan.

Pada tahap ini dilakukan proses pengujian penetrasi dilakukan dengan OWASP *Mobile Security Testing Guide* (MSTG) yang didasari OWASP *Mobile Application Security Verification Standard* (MASVS) pada cakupan *Data Storage on Android* dan *Local Authentication on Android* di level 1. Alasan pengujian dilakukan pada cakupan *Data Storage on Android* dan *Local Authentication on Android* di level 1, dikarenakan cakupan *Data Storage on Android* terkait media penyimpanan data aplikasi ABC, penyimpanan data menjadi sesuatu yang penting bagi aplikasi *mobile*, apabila data tidak dilindungi dengan baik maka dapat mengakibatkan bocornya data yang kemudian dapat disalahgunakan oleh penyerang. Data yang terdapat pada aplikasi ABC ini berupa data XXX masyarakat.

Sedangkan cakupan *Local Authentication on Android* di level 1 ini keterkaitannya terhadap mekanisme autentikasi lokal, autentikasi lokal yaitu autentikasi yang dilakukan oleh aplikasi terhadap kredensial yang disimpan secara lokal pada perangkat. Tujuannya untuk efisiensi ketika kembali ke sesi yang telah digunakan dan memberikan rasa nyaman bagi pengguna. Pengujian level 1 dipilih, karena level tersebut dapat dijadikan sebagai standar minimal keamanan pada suatu aplikasi *mobile*.

a. *Testing Local Storage for Sensitive Data* (MSTG-STORAGE-1 and MSTG-STORAGE-2)

   Developer aplikasi *mobile* diharapkan untuk menyimpan sesedikit mungkin data sensitif pada *local storage* yang bersifat permanen. Tujuannya agar data sensitif yang disimpan tidak bocor akibat serangan aplikasi berbahaya. Pada perangkat Android terdapat beberapa media penyimpanan informasi, seperti *internal storage*, *external storage*, dan *server* atau *cloud storage*. Pada penelitian ini dilakukan identifikasi sumber dengan tujuan untuk menentukan jenis penyimpanan pada Aplikasi ABC dan mengetahui informasi sensitif yang tersimpan.

   a) Melakukan identifikasi *file* Android.Manifest.xml untuk memeriksa perizinan *external storage*.
   b) Mengecek bagian *shared preferences* di *file* XML (didata/data/id.go.xxx/shared_prefs) untuk memeriksa temuan informasi sensitif.
   c) Identifikasi perizinan pada *package file* aplikasi ABC di direktori /data/data/id.go.xxx. Harusnya hanya pengguna dan grup yang telah dibuat aplikasi ketika diinstal yang punya akses *read*, *write*, dan *execute.*

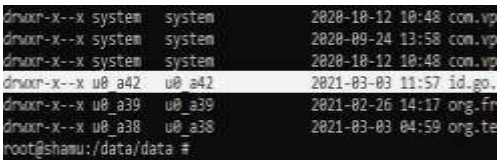
Gambar 1. *File AndroidManifest.xml*

Berdasarkan Gambar 1 terkait *file* AndroidManifest.xml, aplikasi ABC meminta perizinan untuk menulis pada media *external storage*,

artinya aplikasi tersebut memungkinkan untuk menulis data pada media *external storage*, selain itu berdasarkan kondisi tersebut maka pada *external storage* juga memungkinkan untuk menyimpan *cache*.

Gambar 2. *File Shared Preferences*

Berdasarkan Gambar 2, selanjutnya dilakukan identifikasi di *shared preferences* pada *file* XML (data/data/id.go.xxx/shared_prefs) untuk mengetahui informasi yang bersifat sensitif pada aplikasi ABC. Hasilnya didapatkan kerentanan informasi sensitif dari pengguna khususnya di *file shared preferences.*



Gambar 3. Perizinan direktori pada id.go.xxx

Berdasarkan Gambar 3 diketahui bahwa direktori id.go.xxx telah memenuhi standar keamanan aplikasi, yaitu hanya pengguna yang telah menginstal aplikasi ABC yang dapat melakukan *read*, *write*, dan *execute*. Di pengujian ini pengguna aplikasi ABC telah terinstal adalah pengguna u0_a42.

b. *Testing Local Storage for Input Validation* (MSTG PLATFORM-2)

Ketika menggunakan *file shared preferences* untuk membaca atau menulis nilai *int/boolean* maka pengguna tidak dapat memeriksa apakah data tersebut dapat ditimpa atau tidak. Dalam kasus *string* atau *string set* pengguna harus berhati-hati saat menginterpretasikan data. Pada pengujian ini dilakukan analisis validasi *input* terhadap mekanisme *login* berdasarkan *source code* pada *MainActivity* aplikasi ABC hasil dari *reverse engineering*.

Berdasarkan Gambar 4 dan Gambar 5 telah dilakukan analisis validasi *input* terhadap mekanisme *login* pada *source code* aplikasi ABC. Ternyata mekanisme validasi *input* telah dilakukan di *shared preferences* pada bagian akun pengguna berupa *password*, namun pada bagian akun pengguna berupa NIK belum dilakukan. Hal tersebut dapat memunculkan informasi sensitif yang dapat dimanfaatkan oleh penyerang.

Gambar 4. *Source code* aplikasi ABC (a)

Gambar 4. *Source code* aplikasi ABC (b)

c. *Testing Logs for Sensitive Data* (MSTG STORAGE-3)

Pada umumnya *developer* membuat *file log*, dengan tujuan antara lain untuk melacak kerusakan, kesalahan penggunaan aplikasi, dan statistik penggunaan aplikasi. *File log* tersebut dapat disimpan di media *internal storage* atau bisa juga dikirim ke *endpoint* tertentu. Namun dikhawatirkan ketika aplikasi memungkinkan menulis *log*, terdapat informasi sensitif pengguna yang dapat disalahgunakan untuk melanggar hak privasi pengguna.

Gambar 5. Hasil *log* pada *virtual device*

Pada pengujian ini dilakukan percobaan melihat aktivitas *log* pada *virtual device* ketika menjalankan aplikasi ABC. Pengujian menggunakan bantuan *tool* adb, selanjutnya dilakukan identifikasi *output* dari perintah "$adb logcat" melalui *Command Prompt* (CMD). Pengujian tersebut dilakukan pada salah satu fitur layanan aplikasi ABC, yaitu layanan pengajuan XXX. Proses *capture* aktivitas *log* pada XXX dapat menjadi salah satu acuan untuk

menentukan informasi apa saja yang mungkin tertangkap oleh aktivitas *log* di bagian menu layanan aplikasi ABC. Ketika dilakukan proses pengajuan KIA terdapat salah satu syarat yang harus dipenuhi bagi pengguna, yaitu *upload* foto, setelah itu dilakukan *capture* kamera untuk mendapatkan *file* gambar. Dari hasil analisis *log* ditemukan adanya aktivitas *capture* kamera hingga *upload* foto, seperti yang terlihat pada Gambar 5. Berdasarkan pengecekan yang dilakukan, hasilnya tidak ditemukan data bersifat sensitif yang diekpose melalui *log.*

d. *Determining Whether Sensitive Data is Sent to Third Parties* (MSTG-STORAGE-4)
Ketika terjadi proses komunikasi pengiriman data antara *client* dan *server*, *developer* harus memastikan bahwa hanya data yang dibutuhkan oleh *client* saja yang dikirim. Pengujian pada bagian ini berupa percobaan *intercept* pada *traffic* antara *client* dan *server* ketika menggunakan aplikasi ABC. *Tool* yang dipakai adalah Burp Suite. Tujuan pengujian ini adalah untuk mengetahui informasi apa saja yang didapatkan ketika proses komunikasi dari *server* ke *client.*

Pengujian pertama yaitu melakukan identifikasi paket *traffic network* ketika *client login* ke aplikasi ABC.

Gambar 6. Paket *request* dan *response login* dari *server* (a)

Berdasarkan Gambar 6, diketahui pengguna melakukan *login* ke sistem instansi XYZ melalui aplikasi ABC dengan mengirim akun berupa NIK dan *password* sesuai *credential* dari pengguna. Hasil respon dari *server* menunjukkan kode respon 303 yang berarti permintaan yang dilakukan oleh pengguna aplikasi telah diterima atau sukses, selanjutnya diteruskan ke URL yang terhubung dengan aplikasi ABC dalam hal ini https://xxx.go.id.
Pengujian kedua yaitu memasukkan akun berupa NIK dengan kondisi benar, tetapi menggunakan *password* yang tidak sesuai dengan *credential* dari pengguna aplikasi ABC. Hasil yang didapatkan yaitu *server* merespon bahwa *credential* yang dimasukkan tidak sesuai, kemudian menampilkan keterangan NIK atau kata kunci yang digunakan masih salah, sehingga pengguna tidak dapat mengakses. Hasil respon *server* terlihat pada Gambar 7.

Gambar 7. Paket *request* dan *response login* dari server (b)

Gambar 8. Hasil *intercept* komunikasi antara *client* dan *server* menggunakan *tool* Wireshark

Untuk memastikan dan memvalidasi bahwa *traffic network* tersebut sudah benar-benar aman atau belum, maka ketika proses *login* oleh pengguna yang telah dilakukan pada pengujian sebelumnya, dilakukan *intercept* menggunakan *tool* Wireshark seperti yang ditunjukkan pada Gambar 8. Hasilnya proses komunikasi data antara *client* dan *server* ternyata sudah dienkripsi menggunakan pengamanan *Secure Socket Layer* (SSL) v2, sehingga dengan kondisi tersebut dapat dikategorisasikan *traffic network* aplikasi ABC dalam kondisi yang aman.

e. *Determining the Keyboard Cache is Disable for Text Input Fields* (MSTG-STORAGE-5)
Pada saat memasukkan data, aplikasi dapat memberikan keterangan rekomendasi pada *input fields*, berarti aplikasi tersebut memungkinkan menyimpan *keyboard cache*. Pada pengujian ini dilakukan proses memasukkan kata pada *fields* yang berkaitan dengan permintaan data bersifat sensitif, misalnya *username* atau *password.* Jika *fields* memberikan rekomendasi kata yang telah dimasukkan sebelumnya berarti aplikasi tersebut dapat menyimpan *keyboard cache*.

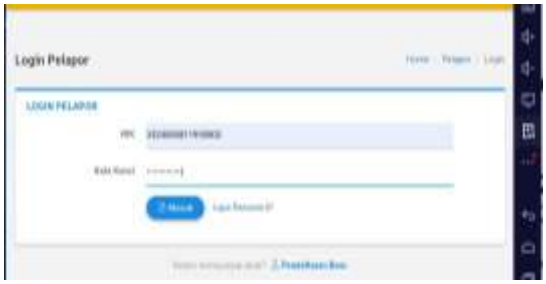
Gambar 9. Pengujian terkait kondisi *keyboard cache fields*

Gambar 9 menunjukkan bahwa aplikasi ABC di bagian *fields* memberikan rekomendasi terkait akun pengguna berupa NIK yang telah dimasukkan sebelumnya. Artinya dengan kondisi tersebut aplikasi ABC dapat menyimpan *keyboard cache.*

f. *Determining Whether Sensitive Stored Data Has Been Exposed via IPC Mechanisms* (MSTG-STORAGE-6)
Content provider sebagai bagian dari mekanisme IPC dapat memberikan akses kepada aplikasi Android untuk melakukan *read* dan *write* data yang disimpan. Hal tersebut dapat menimbulkan kebocoran data yang bersifat sensitif, jika aplikasi Android tidak dikonfigurasi dengan benar, maka data sensitif tersebut dapat dimanfaatkan untuk melakukan eksploitasi. Gambar 10 menunjukkan hasil pengujian data sensitif tidak diekpose melalui IPC dalam aplikasi ABC. Hasil pengujian menggunakan *tool* Drozer menunjukkan bahwa tidak ada penyedia konten yang dapat diakses oleh aplikasi lain.

Gambar 10. Hasil pengujian data sensitif tidak terekpose melalui IPC

g. *Checking for Sensitive Data Disclosure Through the User Interface* (MSTG-STORAGE-7)
Pada umumnya aplikasi Android mengharuskan *user* untuk memasukkan data bersifat sensitif, misalnya ketika *login* akun yang mengharuskan *user* untuk memasukkan *password*. Data tersebut bersifat sensitif, sehingga memungkinkan untuk diekspose yang memungkinkan bisa dilakukan serangan *shoulder surfing* apabila *input* data tidak di-*masking*, seperti menggantinya dengan tanda ".", "*", dan lain sebagainya.
Berdasarkan Gambar 11, pada aplikasi ABC dilakukan pengujian pada *fields password*, tujuannya untuk mengetahui *password* yang telah di-*input*-kan ke aplikasi ABC sudah di-*masking* atau belum. Diketahui *password* yang telah telah di-*input*-kan ke aplikasi ABC sudah diubah menjadi tanda titik atau *dots.* Artinya sudah dilakukan *masking*.



Gambar 11. *Login page* aplikasi ABC

h. *Testing Confirm Credentials* (MSTG-AUTH-1)
Alur konfirmasi kredensial tersedia sejak Android 6.0 dan digunakan untuk memastikan bahwa pengguna tidak perlu memasukkan *password* khusus aplikasi bersama dengan perlindungan *screen lock.* Pada penelitian ini dilakukan penerapan terhadap aplikasi ABC diharuskan menggunakan *screen lock* untuk membuka aplikasi tersebut. Penerapan *screen lock* pada aplikasi berbeda dengan *screen lock* pada perangkat. Hal ini bertujuan untuk mencegah apabila perangkat pengguna tidak menggunakan *screen lock*, ternyata aplikasi ABC tetap memiliki *screen lock*. Selain itu aplikasi ABC juga dapat menggunakan *biometric authentication* agar aplikasi ABC dapat dibuka dengan *biometric authentication* yang digunakan pengguna untuk membuka perangkatnya.

4.5. *Reporting*

Berdasarkan tahap pengujian yang telah dilakukan, ada tiga bagian pengujian terdapat kerentanan yang tervalidasi berdasarkan hasil uji penetrasi, yaitu pada bagian *Testing Local Storage for Sensitive Data, Testing Local Storage for Input Validation,* dan *Determining Whether the Keyboard Cache Is Disabled for Text Input Fields.* Hasil temuan kerentanan tersebut ditujukkan pada Tabel 3.

Tabel 3. Hasil kerentanan tervalidasi

No	Pengujian *Data Storage on Android* dan *Local Authentication and Session Management on Android* di level 1	Ada Kerentanan
1	*Testing Local Storage for Sensitive Data*	Ya
2	*Testing Local Storage for Input Validation*	Ya
3	*Testing Logs for Sensitive Data*	Tidak
4	*Determining Whether Sensitive Data is Sent to Third Parties*	Tidak
5	*Determining Whether the Keyboard Cache Is Disabled for Text Input Fields*	Ya
6	*Determining Whether Sensitive Stored Data Has Been Exposed via IPC*	Ya
7	*Checking for Sensitive Data Disclosure Through the User*	Tidak
8	*Testing Confirm Credentials*	Tidak

Setelah itu dilakukan rekomendasi perbaikan berdasarkan kerentanan yang dilakukan untuk meningkatkan keamanan aplikasi *mobile.* Rekomendasi pengamanan yang diberikan adalah:

a. *Testing Local Storage for Sensitive Data*
Rekomendasi keamanan yang diberikan yaitu dengan memberikan pengamanan enkripsi di *shared preferences.*

b. *Testing Local Storage for Input Validation*
Rekomendasi keamanan yang diberikan yaitu memberikan pengamanan validasi *input* pada akun pengguna aplikasi.

c. *Determining Whether the Keyboard Cache Is Disabled for Text Input Fields*
Rekomendasi keamanan yang diberikan yaitu aplikasi tidak menyimpan *cache*.

5. KESIMPULAN

Berdasarkan pengujian pada aplikasi *mobile* ABC berbasis Android menggunakan OWASP *Mobile Security Testing Guide*, dengan cakupan *Data Storage on Android* dan *Local Authentication on Android* di level 1, pada delapan bagian pengujian yang telah dilakukan, dapat disimpulkan bahwa jenis kerentanan yang berhasil ditemukan pada aplikasi *mobile* ABC yaitu:

a. *Testing Local Storage for Sensitive Data*
Pada bagian ini memiliki kerentanan data sensitif dapat bocor.

b. *Testing Local Storage for Input Validation*
Data sensitif pada aplikasi dapat terekpose.

c. *Determining Whether the Keyboard Cache Is Disabled for Text Input Fields*
Ketika aplkasi dapat menampilkan *keyboard cache*, dikhawatirkan infromasi sensitif tersebut disalahgunakan

REFERENSI

[1] C. Sam and L. Sydow, "appannie, Retrospective: A Monumental Year for the App Economy," 2017. [Online]. Available: https://www.appannie.com/en/insights/market-data/app-annie-2017 retrospective/. [Accessed 14/12/2020].

[2] Y. Kouraogo, K. Zkik, E. J. El Idrissi Noreddine, and G. Orhanou, "Attacks on Android banking applications," *Proc. - 2016 Int. Conf. Eng. MIS, ICEMIS 2016*, 2016, doi: 10.1109/ICEMIS.2016.7745337.

[3] D. Kaitlyn, "securitytoday, Insecure Data Storage in Mobile Application Poses Security Issues," 2019. [Online]. Available: https://securitytoday.com/articles/2019/06/24/insecure-data-storage-in-mobile-applications-poses-security-issues.aspx. [Accessed 14/12/2020].

[4] K. Qian, R. M. Parizi, and D. Lo, "OWASP Risk Analysis Driven Security Requirements Specification for Secure Android Mobile Software Development," *DSC 2018 - 2018 IEEE Conf. Dependable Secur. Comput.*, pp. 1–2, 2019, doi: 10.1109/DESEC.2018.8625114.

[5] B. M. J. Williemsen, S. Schleier, *MSTG Mobile Security Testing Guide*. OWASP, 2018.

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

[7] Sugiyono, *Metode Penelitian Kuantitatif Kualitatif dan R&D*. Bandung: Alfabeta, 2013.