ANALISIS HASIL UJI KERENTANAN APLIKASI MOBILE PEMERINTAH KOTA DENPASAR

N.W.A.P. Astawa[1], I.M. Widiartha[2], dan I.D.M.B.A. Darmawan[3]

ABSTRAK

Denpasar Parama Sewaka merupakan aplikasi mobile milik Pemerintah Kota Denpasar yang menyediakan berbagai layanan penting kepada masyarakat Kota Denpasar. Aplikasi ini baru diluncurkan sehingga memungkinkan adanya banyak kerentanan dalam aplikasi yang dapat dimanfaatkan oleh pihak tidak berwenang untuk melakukan serangan. Untuk itu diperlukan adanya uji penetrasi untuk menguji kerentanan aplikasi ini. Salah satu framework yang dapat digunakan adalah Mobile Security Framework (MobSF) yang melakukan evaluasi terhadap sertifikat keamanan dan standar NIAP v1.3. Metode ini memungkinkan identifikasi kerentanan dengan lebih rinci, seperti analisis sertifikat yang mengungkapkan adanya kerentanan Janus pada Android 5.0-8.0 dan berbagai kerentanan lainnya yang dikelompokkan berdasarkan tingkat keparahannya. Hasilnya menunjukkan bahwa aplikasi tersebut memiliki total 21 kerentanan, yang terdiri dari kerentanan tinggi, peringatan, dan informasi.

Kata kunci: uji penetrasi, aplikasi mobile, kerentanan, analisis keamanan, aplikasi pemerintah

ABSTRACT

Denpasar Parama Sewaka is a mobile application owned by the Denpasar City Government that provides various important services to the people of Denpasar City. This application is newly launched so there may be many vulnerabilities in the application that can be exploited by unauthorized parties to carry out attacks. For this reason, a penetration test is needed to test the vulnerability of this application. One framework that can be used is the Mobile Security Framework (MobSF) which evaluates security certificates and the NIAP v1.3 standard. This method allows for more detailed identification of vulnerabilities, such as the certificate analysis that revealed the Janus vulnerability on Android 5.0-8.0 and various other vulnerabilities grouped by severity. The results show that the app has a total of 21 vulnerabilities, consisting of high, warning, and informational vulnerabilities.

Keywords: penetration testing, mobile application, vulnerabilities, security analysis, government mobile applications

[1] *Program Studi Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana, amandaputri@unud.ac.id*
[2] *Program Studi Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana, madewidiartha@unud.ac.id*
[3] *Program Studi Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana, dewabayu@unud.ac.id*

Submitted: 9 Januari 2025 *Revised: 24 Januari 2025* *Accepted: 25 Januari 2025*

1. PENDAHULUAN

Di era digital saat ini, aplikasi mobile telah menjadi sarana penting dalam memberikan layanan publik yang cepat dan efisien. Pemerintah Kota Denpasar, melalui aplikasi Denpasar Parama Sewaka, berupaya untuk memberikan berbagai layanan penting kepada masyarakat Kota Denpasar. Aplikasi ini diharapkan dapat meningkatkan aksesibilitas layanan pemerintah, mulai dari informasi umum hingga layanan administrasi dan pembayaran. Peluncuran aplikasi ini mencerminkan komitmen pemerintah dalam memanfaatkan teknologi untuk memperbaiki kualitas pelayanan publik. Namun, seiring dengan peluncuran aplikasi mobile, muncul tantangan baru terkait dengan keamanan siber (Alanda, et. al., 2019). Aplikasi yang baru diluncurkan sering kali menjadi target serangan siber karena kerentanannya yang belum teridentifikasi dan diperbaiki. Oleh karena itu, keamanan aplikasi menjadi prioritas utama untuk memastikan bahwa data dan informasi pengguna tetap aman.

Aplikasi Denpasar Parama Sewaka, meskipun memiliki banyak manfaat, juga menghadapi berbagai tantangan keamanan. Kerentanan dalam aplikasi mobile dapat dimanfaatkan oleh pihak tidak berwenang untuk melakukan serangan, yang dapat berakibat pada pencurian data pribadi, pengambilalihan akun, dan bahkan kerusakan sistem (Aldi & Ray, 2024). Standar keamanan aplikasi mobile yang tidak memadai juga dapat menjadi masalah utama. Sertifikat keamanan yang tidak dikelola dengan baik dan tidak memenuhi standar keamanan yang ditetapkan, seperti standar NIAP v1.3, dapat membuka celah bagi serangan siber. Oleh karena itu, diperlukan evaluasi menyeluruh terhadap sertifikat keamanan dan standar yang digunakan oleh aplikasi untuk memastikan bahwa aplikasi tersebut aman digunakan oleh masyarakat.

Penelitian sebelumnya telah menunjukkan pentingnya uji penetrasi dalam mengidentifikasi dan mengatasi kerentanan dalam aplikasi mobile. Menurut Sangeeta dan Kanwalvir (2020), analisis dan pengujian malware pada aplikasi Android menunjukkan bahwa banyak aplikasi yang tidak memenuhi standar keamanan yang diperlukan, yang meningkatkan risiko serangan siber. Selain itu, Chairul, dkk (2023) mengungkapkan bahwa penggunaan Mobile Security Framework (MobSF) dapat memberikan evaluasi keamanan yang komprehensif terhadap aplikasi mobile, termasuk pemeriksaan sertifikat keamanan dan kepatuhan terhadap standar NIAP v1.3. Penelitian Imam, Kevin, dan Irawan (2023) juga mendukung pentingnya menggunakan framework yang andal seperti MobSF untuk melakukan uji penetrasi. Framework ini tidak hanya membantu dalam mengidentifikasi kerentanan yang ada tetapi juga memberikan panduan untuk memperbaikinya. Dengan demikian, penerapan MobSF sebagai alat uji penetrasi dapat meningkatkan keamanan aplikasi mobile secara signifikan.

Untuk mengatasi masalah keamanan yang dihadapi oleh aplikasi Denpasar Parama Sewaka, diperlukan uji penetrasi yang komprehensif menggunakan framework yang andal seperti Mobile Security Framework (MobSF). MobSF adalah alat open-source yang menyediakan evaluasi keamanan lengkap untuk aplikasi mobile (Fauzan, 2021). Framework ini memungkinkan identifikasi kerentanan dengan lebih rinci, termasuk analisis sertifikat keamanan dan kepatuhan terhadap standar NIAP v1.3. Dengan menggunakan MobSF, pengembang dan peneliti keamanan dapat melakukan analisis mendalam terhadap aplikasi, mengidentifikasi kerentanan yang dikelompokkan berdasarkan tingkat keparahannya. Evaluasi ini membantu dalam menentukan prioritas perbaikan dan memastikan bahwa aplikasi lebih tahan terhadap serangan siber (Cholis & Deanna, 2019).

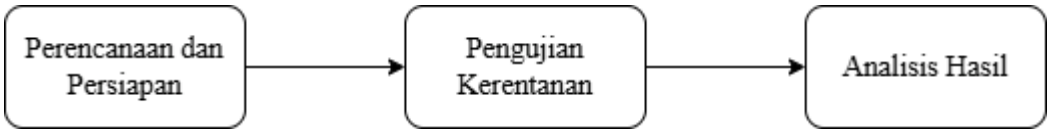
Selain itu, penerapan MobSF juga memungkinkan pengembangan aplikasi yang lebih aman sejak awal. Dengan melakukan uji penetrasi secara rutin, pengembang dapat memantau dan memperbaiki kerentanan yang muncul seiring dengan perkembangan aplikasi. Hal ini tidak hanya meningkatkan keamanan aplikasi, tetapi juga membangun kepercayaan pengguna terhadap layanan yang disediakan. Dalam konteks Denpasar Parama Sewaka, penerapan MobSF sebagai bagian dari uji

penetrasi dapat memberikan jaminan tambahan bahwa aplikasi tersebut aman untuk digunakan oleh masyarakat. Langkah ini akan membantu Pemerintah Kota Denpasar dalam menciptakan lingkungan digital yang lebih aman dan terpercaya, serta memastikan bahwa aplikasi tersebut dapat memberikan layanan yang efisien dan aman kepada masyarakat.

2. METODE PELAKSANAAN

Metode pelaksanaan dalam penelitian ini mencakup penggunaan alat bantu dalam pengecekan aplikasi mobile Denpasar Parama Sewaka yaitu Mobile Security Framework (MobSF), yang dijalankan di sistem operasi Kali Linux. MobSF sendiri merupakan adalah kerangka kerja penelitian keamanan, analisis malware, dan penilaian keamanan aplikasi seluler yang otomatis dan komprehensif, yang mampu melakukan analisis statis dan dinamis (Isnaini & Suhartono, 2023). MobSF mendukung berkas aplikasi seluler (APK, XAPK, IPA, dan APPX) serta kode sumber yang di-zip, dan menyediakan antarmuka REST untuk integrasi yang lancar dengan CI/CD atau jalur DevSecOps. Setelah itu hasilnya akan dianalisis sehingga didapatkan saran-saran yang dapat dilakukan untuk meningkatkan keamanan aplikasi.

Pada penelitian ini menggunakan metode uji penetrasi. Tahapan investigasi atau langkah-langkah yang dilakukan dalam uji penetrasi adalah sebagai berikut:



Gambar 2.1. Tahapan Analisis Uji Kerentanan Aplikasi Mobile Denpasar Prama Sewaka

a. Perencanaan dan Persiapan

   Tahapan awal adalah menentukan cakupan serta sasaran pengujian, termasuk sistem yang akan diuji dan metode pengujian yang akan diterapkan. Pada tahap ini, ditentukan jenis pengujian serta analisis apa yang akan dilakukan.

b. Pengujian Kerentanan

   Ini adalah fase menganalisis informasi terperinci yang telah diperoleh tentang risiko dan kerentanan keamanan. Setelah proses upload selesai, maka akan didapatkan hasil kerentanan dari aplikasi tersebut beserta aspek aspek analisisnya seperti analisis sertifikat dan juga NIAP v1.3. Selanjutnya, kerentanan yang ditemukan kemudian dibagi menjadi 3 tingkat, yaitu High, Warning, dan Informatif lalu akan diteliti pula cara utnuk mengatasi kerentanan yang sudah ditemukan.

c. Analisis Hasil

   Pada fase ini, diberikan beberapa saran atau instruksi dari perspektif keamanan sistem sehingga kerentanan yang terdeteksi dapat diperbaiki atau diperbarui oleh pihak terkait. Saran tersebut didasarkan pada NIST Special Publication dan juga jurnal-jurnal terkait.

3. HASIL DAN PEMBAHASAN

3.1 Perencanaan dan Persiapan

Pada penelitian ini digunakan aplikasi mobile milik Dinas Komunikasi, Informatika dan Statistik Kota Denpasar yakni Denpasar Parama Sewaka. Pada tahapan persiapan, penulis mengunduh aplikasi tersebut dari Play Store dan mengubahnya ke dalam bentuk APK dan dilanjutkan dengan proses upload ke software MobSF. Analisis yang akan dilakukan adalah analisis berdasarkan Sertifikat dan NIAP v3.1

3.2 Pengujian Kerentanan

Dari hasil pengujian pada aplikasi mobile Denpasar Parama Sewaka, didapatkan informasi terkait sertifikat untuk aplikasi tersebut. Berikut adalah informasi mengenai sertifikat yang ada pada aplikasi Denpasar Parama Sewaka:

APK is signed v1
signature: True v2
signature: True v3
signature: True
Found 1 unique certificates
Subject: C=ID, ST=Bali, L=Denpasar, O=Djingga Media, OU=Djingga Media, CN=Djinggamedia Team
Signature Algorithm: rsassa_pkcs1v15
Valid From: 2016-02-26 23:42:25+00:00
Valid To: 2041-02-19 23:42:25+00:00
Issuer: C=ID, ST=Bali, L=Denpasar, O=Djingga Media, OU=Djingga Media, CN=Djinggamedia Team
Serial Number: 0x6444cc78 Hash Algorithm: sha256 md5: 3b12f856b312dc8b2b0b36d12bfeb9ea sha1: 1ae5359e7fbf82df6c35731f54fe2bc54b5f2688 sha256: 93e6463f31fb1f5e6f98c1be02c392317216e121160744619f8d985dc6ee4252 sha512: 8ab53dd550de0e6bd62d5b85a7fc2104fbdb867c96476edafb4f6ed0dfd6547ea2d0a25c3f34018e11bff343c53eb07d9698b205dce0663b039de32700f5223d
PublicKey Algorithm: rsa
Bit Size: 2048

APK aplikasi mobile Denpasar Prama Sewaka telah ditandatangani secara digital dengan menggunakan tiga versi tanda tangan yang valid, yaitu v1, v2, dan v3. Penandatanganan ini diverifikasi dengan satu sertifikat unik yang diidentifikasi sebagai berasal dari Djingga Media, dengan rincian subjek C=ID, ST=Bali, L=Denpasar, O=Djingga Media, OU=Djingga Media Team, CN=Djinggamedia Team. Sertifikat ini menggunakan algoritma tanda tangan rsa, sha256WithRSAEncryption dan memiliki periode validitas mulai dari 26 Februari 2016 hingga 19 Februari 2041. Nomor seri sertifikat adalah 0x6444c78.

Sertifikat ini juga menggunakan algoritma hash SHA-256 untuk memverifikasi integritasnya. Beberapa nilai hash yang dihasilkan termasuk md5 (3e1238563b12cdb8b203d6d12fbe9bea), sha1 (125ea5967fbf82dfc3573f15af2bc5435f2688), sha256 (396e34d53f1d1f5e6f3bef1c0eb9237121f6e121160746419f8d95de6ee4252), dan sha512 (ba36d5d50d0e6cb62d5b85a7c2f2104dbb867c96476edafb4fed0dfd547ea2d0a25cf343018e11bff343c53eb07d6968b205d0e663b039de32700f5223d). Algoritma kunci publik yang digunakan adalah RSA dengan ukuran kunci sebesar 2048 bit, menunjukkan tingkat keamanan yang tinggi. Informasi

ini menunjukkan bahwa APK tersebut telah melalui proses verifikasi yang ketat dan menggunakan metode keamanan yang kuat untuk menjamin integritas dan keasliannya.

Pada analisis sertifikat, ditemukan sebanyak 2 kerentanan yang dibagi menjadi 1 warning, dan 1 info. Penjelasan lebih lanjut mengenai kerentanan yang ditemukan dapat dilihat pada **Tabel 3.1**.

Tabel 3.1. Hasil Analisis Sertifikat Aplikasi Mobile Denpasar Prama Sewaka

Tingkat	Kerentanan	Deskripsi
Warning	Vulnerable to Janus Vulnerability	Terdapat kerentanan Janus pada Android 5.0-8.0, jika aplikasi hanya ditandatangani dengan skema tanda tangan V1.
Info	Signed Application	Aplikasi ini ditandatangani dengan sertifikat penandatanganan kode.

Adapun pada analisis NIAP v1.3, ditemukan sebanyak 19 kerentanan yang dibagi menjadi 2 high, 1 warning, dan 16 info seperti yang dijabarkan pada **Tabel 3.2**.

Identifier	Syarat	Fitur	Deskripsi
FCS_RBG_EXT.1.1	Persyaratan fungsional keamanan	Layanan generasi bit acak	Aplikasi ini menggunakan fungsionalitas DRBG yang disediakan platform untuk operasi kriptografinya
FCS_STO_EXT.1.1		Penyimpanan kredensial	Aplikasi tidak menyimpan kredensial apa pun ke memori non-volatil
FCS_CKM_EXT.1.1		Layanan Generasi Kunci Kriptografis	Aplikasi menerapkan pembuatan kunci asimetris
FDP_DEC_EXT.1.1		Akses ke Sumber Daya Platform	Aplikasi ini memiliki akses ke ['konektivitas jaringan', 'lokasi']
FDP_DEC_EXT.1.2		Akses ke Repositori Informasi Sensitif	Aplikasi ini tidak memiliki akses ke tempat penyimpanan informasi sensitif.
FDP_NET_EXT.1.1		Komunikasi Jaringan	Aplikasi memiliki komunikasi jaringan yang diinisiasi oleh pengguna/aplikasi
FDP_DAR_EXT.1.1		Enkripsi Data Aplikasi Sensitif	Aplikasi mengimplementasikan fungsionalitas untuk mengenkripsi data sensitif dalam memori non-volatile
FMT_MEC_EXT.1.1		Mekanisme Konfigurasi yang Didukung	Aplikasi ini menggunakan mekanisme yang direkomendasikan oleh vendor platform untuk menyimpan dan mengatur opsi konfigurasi
FTP_DIT_EXT.1.1	Persyaratan Fungsional Keamanan Berdasarkan Pemilihan	Proteksi Data pada Transit	Aplikasi ini mengenkripsi beberapa data yang ditransmisikan dengan HTTPS/TLS/SSH antara aplikasi itu sendiri dan produk TI tepercaya lainnya

Identifier	Syarat	Fitur	Deskripsi
FCS_RBG_EXT.2.1,FCS_RBG_EXT2.2	Persyaratan Fungsional Keamanan Berdasarkan Pemilihan	Generasi Bit Acak dari Aplikasi	Aplikasi ini menjalankan semua layanan pembangkitan bit acak deterministik (DRBG) sesuai dengan NIST Special Publication 800- 90A menggunakan Hash_DRBG. RBG deterministik diunggulkan oleh sumber entropi yang mengakumulasi entropi dari DRBG berbasis platform dan sumber noise berbasis perangkat lunak, dengan minimal 256 bit entropi yang setidaknya sama dengan kekuatan keamanan terbesar (menurut NIST SP 800-57) dari kunci dan hash yang akan dihasilkannya
FCS_CKM.1.1(1)		Generasi Kunci Kriptografis Asimetris	Aplikasi generasi kunci kriptografis asimetris tidak sesuai dengan FCS_CKM.1.1(1) menggunakan skema algoritma generasi kunci RSA dan ukuran kunci kriptografis sebesar 1024 bit atau lebih rendah
FCS_COP.1.1(1)		Operasi Kriptografis - Enkripsi/Dekripsi	Aplikasi melakukan enkripsi/dekripsi tidak sesuai dengan FCS_COP.1.1(1), mode AES-ECB digunakan
FCS_COP.1.1(2)		Operasi Kriptografis - Hashing	Aplikasi melakukan layanan penghashan kriptografis sesuai dengan algoritma kriptografis yang ditentukan SHA-1/SHA-256/SHA-384/SHA-512 dan ukuran panjang pesan 160/256/384/512 bit
FCS_HTTPS_EXT.1.1		Protokol HTTPS	Aplikasi ini mengimplementasikan protokol HTTPS yang sesuai dengan RFC 2818
FCS_HTTPS_EXT.1.2			Aplikasi mengimplementasikan HTTPS menggunakan TLS
FCS_HTTPS_EXT.1.3			Aplikasi akan memberi tahu pengguna dan tidak membuat sambungan atau meminta otorisasi aplikasi untuk membuat sambungan jika sertifikat peer dianggap tidak valid
FIA_X509_EXT.2.1		Otentikasi Sertifikat X.509	Aplikasi ini menggunakan sertifikat X.509v3 seperti yang didefinisikan oleh RFC 5280 untuk mendukung otentikasi untuk HTTPS, TLS
FPT_TUD_EXT.2.1		Integritas untuk Instalasi dan Pembaruan	Aplikasi harus didistribus menggunakan format pengelola paket yang didukung platform

FCS_CKM.1.1(2)	Persyaratan Fungsional Keamanan Opsional	Generasi Kunci Simetris Kriptografis	Aplikasi harus menghasilkan kunci kriptografi simetris menggunakan Random Bit Generator seperti yang ditentukan dalam FCS_RBG_EXT.1 dan ukuran kunci kriptografi yang ditentukan 128 bit atau 256 bit

Tabel 3.2. Hasil Analisis NIAP v1.3 Aplikasi Mobile Denpasar Prama Sewaka

3.3 Analisis Hasil

Berdasarkan hasil analisis tersebut, terdapat 1 kerentanan bersifat warning dan 2 kerentanan bersifat high. Untuk kerentanan bersifat warning terdapat pada indentifier FCS_STO_EXT.1.1. Fitur yang diuji adalah Penyimpanan Kredensial, dengan deskripsi bahwa aplikasi tidak menyimpan kredensial apapun di memori non-volatil. Seperti yang dijelaskan dalam NIST Special Publication 800-88 (Guidelines for Media Sanitization), hal ini menunjukkan bahwa aplikasi dirancang untuk menghindari penyimpanan data sensitif seperti kata sandi, kunci kriptografi, atau token autentikasi di media penyimpanan yang dapat mempertahankan data meskipun perangkat seperti hard disk atau SSD dimatikan. Tidak menyimpan kredensial di memori non-volatil merupakan hal yang cukup penting, mengingat penyerang dapat menggunakan memori non-volatil untuk mengambil perangkat dan memulihkan data sensitif dengan teknik forensik. Pihak yang tidak berwenang dapat dengan mudah mengakses data media bahkan setelah perangkat dimatikan. Untuk melindungi kredensial dengan lebih baik, disarankan agar data sensitif hanya disimpan di memori volatil, seperti RAM, yang otomatis terhapus ketika perangkat dimatikan atau di-restart. Jika penyimpanan sementara di memori non-volatil tidak dapat dihindari, maka harus dipastikan bahwa kredensial tersebut dienkripsi dengan baik dan dihapus secara aman setelah tidak diperlukan lagi.

Selanjutnya untuk kerentanan yang bersifat high, pertama terdapat pada identifier FCS_CKM.1.1(1). Fitur yang diuji adalah Generasi Kunci Kriptografis Asimetris, yang memungkinkan aplikasi untuk menghasilkan kunci kriptografi asimetris menggunakan algoritma RSA dengan ukuran kunci minimal 1024-bit. Kunci asimetris terdiri dari pasangan kunci publik dan kunci privat, yang tentunya berperan penting dalam enkripsi dan dekripsi data. Sedangkan algoritma RSA sendiri merupakan salah satu algoritma kriptografi yang banyak digunakan karena didasarkan pada kompleksitas faktorisasi bilangan besar, sehingga banyak dipilih untuk enkripsi data. Namun, menurut NIST Special Publication 800-57 Part 1 (Recommendation for Key Management: Part 1 – General) ukuran kunci RSA 1024-bit saat ini tidak lagi dianggap aman karena perkembangan kemampuan komputasi yang semakin meningkat memungkinkan serangan brute-force menjadi lebih efektif. Sehingga, untuk meningkatkan keamanan aplikasi, direkomendasikan untuk menggunakan ukuran kunci minimal 2048-bit atau lebih besar. Penggunaan ukuran kunci yang lebih besar tidak hanya memperkuat keamanan tetapi juga memperpanjang waktu yang dibutuhkan oleh penyerang untuk berhasil mendekripsi data melalui serangan brute-force.

Terakhir, untuk kerentanan yang bersifat high terdapat pada indentifier FCS_COP.1.1(1). Fitur yang diuji adalah Operasi Kriptografis - Enkripsi/Dekripsi, yang di mana aplikasi menggunakan mode AES-ECB (Electronic Codebook) untuk proses enkripsi dan dekripsi data. AES (Advanced Encryption Standard) merupakan standar enkripsi yang efisien dan kuat, dan digunakan secara luas untuk enkripsi data simetris dengan ukuran kunci 128-bit, 192-bit, atau 256-bit. Namun, mode ECB memiliki kelemahan signifikan dalam hal keamanan. ECB memproses setiap blok data secara independen, yang berarti pola dalam plaintext dapat terlihat dalam ciphertext jika ada blok yang identik. Hal inilah yang membuat mode ini tidak cocok untuk data yang memiliki pola atau data yang panjang karena pola asli dapat terlihat dalam ciphertext, yang mempermudah penyerang untuk mengidentifikasi dan mengeksploitasi data yang terenkripsi. Untuk meningkatkan keamanan aplikasi, sesuai dengan panduan NIST Special Publication 800-38A (Recommendation for Block

Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode) sangat dianjurkan untuk mengganti mode enkripsi dari AES-ECB ke mode yang lebih aman seperti AES-CBC (Cipher Block Chaining) atau AES-GCM (Galois/Counter Mode). Hal ini dikarenakan mode-mode ini menambahkan keacakan melalui vektor inisialisasi (IV) dan memberikan kerahasiaan serta integritas data yang lebih baik. Selain itu, penting untuk memastikan bahwa kunci enkripsi dikelola dengan aman dan praktik terbaik untuk penyimpanan dan distribusi kunci diikuti dengan ketat

.

4. KESIMPULAN

Berdasarkan hasil testing tersebut, didapati bahwa aplikasi ini memiliki tingkat kerentanan yang cukup tinggi, yakni 33/100 dengan total sebanyak 21 kerentanan yang ditemukan baik dalam analisis sertifikat, maupun NIAP v1.3. Hal ini menandakan masih banyak celah dari segi keamanan yang dapat digunakan untuk meretas aplikasi. Sehingga, saran yang dapat diberikan adalah dengan menggunakan ukuran kunci minimal 2048-bit atau lebih besar untuk meningkatkan keamanan, menggunakan mode enkripsi yang lebih aman seperti AES-CBC atau AES-GCM, dan menyimpan data sensitif di memori volatil seperti RAM. Dengan mempertimbangkan saran-saran yang telah diberikan, diharapkan dapat memperkecil kemungkinan aplikasi diretas oleh pihak yang tidak berwenang.

UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada Dinas Komunikasi, Informatika dan Statistik Kota Denpasar atas kesempatan yang telah diberikan untuk melakukan kegiatan pengabdian kepada masyarakat dan mengembangkan kemampuan dalam bidang keamanan siber. Serta penulis turut mengucapkan terima kasih kepada Program Studi Informatika Universitas Udayana atas bimbingan dalam melaksanakan kegiatan pengabdian masyarakat ini.

DAFTAR PUSTAKA

Alanda, A. et. al. (2019). Mobile Application Security Penetration Testing Based on OWASP. *IOP Conf.* **Vol. 846**, pp. 1-11.

Almuhammad S. and Al-Hejri I. (2017). A Comparative Analysis of AES Common Modes of Operation. *Canadian Conference on Electrical and Computer Engineering (CCECE)*.

Anwar, C. et. al. (2023). The Application of Mobile Security Framework (MOBSF) and Mobile A pplication Security Testing Guide to Ensure the Security in Mobile Commerce Applications. *Jurnal Sistim Informasi dan Teknologi*. **Vol. 5**, pp. 97-100.

Bader A. S. and Sagheer A. M. (2018). Modification on AES-GCM to Increment Ciphertext Randomness. *International Journal of Mathematical Sciences and Computing*. **Vol. 4**.

Bahrul M. A. et. al. (2021). Implementasi Kriptografi Menggunakan Algoritma Advanced Encryption Standard (AES) Dengan Metode CBC (Chiper Block Chaining).

Barker E. (2020). Recommendation for Key Management: Part 1 – General. NIST Special Publication 800-57 Part 1 Revision 5. pp. 56-58.

Fachri F. et. al. (2021). Analisis Keamanan Webserver menggunakan Penetration Test. *Jurnal Informatika*. **Vol. 8**, pp. 183-184.

Fauzan A. A. (2021). Android Penetration Testing using Dynamic Analyzer MobSF. Medium. *MII Cyber Security Consulting Services*.

Hanifurohman, C. and Hutagalung, D.D. (2019). Analisa Keamanan Aplikasi Mobile E-Commerce Berbasis Android Menggunakan Mobile Security Framework. *Prosiding Seminar Nasional*. **Vol. 1**.

Himawan I. et. al. (2023). Analisa Resiko Malware Dengan Static MobSF Terhadap Aplikasi Android APK. *Technologia Jurnal Ilmiah*. **Vol. 14**.

Kissel R. et. al. (2014). Guidelines for Media Sanitization. NIST Special Publication 800-88 Revision 1.

Nugraha A. C. F. and Yasa R. N. (2024). Perbandingan Keamanan Aplikasi Pesan Instan Android Menggunakan MobSF (Mobile Security Framework) Berdasarkan Beberapa Standar. *Jurnal Info Kripto*.. **Vol. 18**.

Rani S. and Dhindsa K. S. (2020). Android application security: detecting Android malware and evaluating anti-malware software. *International Journal of Internet Technology and Secured Transactions*. **Vol. 4**.

Santoso N. A. et. al. (2021). Penerapan Metode Penetrasion Testing Pada Keamanan Jaringan Nirkabel. *Jurnal Responsif Riset Sains dan Informatika*. **Vol. 4**, pp. 163-165.

Zen B. P. et. al. (2020). Analisis Security Assessment Menggunakan Metode Penetration Testing Dalam Menjaga Kapabilitas Keamanan Teknologi Informasi Pertahanan Negara. *Jurnal Teknologi Penginderaan*. **Vol. 2**, pp. 106-109.

Halaman ini sengaja dikosongkan