

**PENGUJIAN KEAMANAN SECARA STATIS DAN DINAMIS TERHADAP
APLIKASI MOBILE DI PT INDOMOBIL FINANCE INDONESIA**

SKRIPSI INTERNSHIP

Oleh

Dhia Hauzan Muafa 2301926245

Kelas/Kelompok : LA07



Cyber Security

School of Computer Science

Universitas Bina Nusantara

Jakarta

2023

**PENGUJIAN KEAMANAN SECARA STATIS DAN DINAMIS TERHADAP
APLIKASI MOBILE DI PT INDOMOBIL FINANCE INDONESIA**

SKRIPSI INTERNSHIP

diajukan sebagai salah satu syarat
untuk gelar kesarjanaan pada
Program Studi Cyber Security
Jenjang Pendidikan Strata-1

Oleh

Dhia Hauzan Muafa 2301926245
Kelas/Kelompok : LA07



Cyber Security
School of Computer Science
Universitas Bina Nusantara
Jakarta
2023

Pernyataan Kesiapan Skripsi untuk Sidang Skripsi

Pernyataan Penyusunan Skripsi

Saya, Dhia Hauzan Muafa

dengan ini menyatakan bahwa Skripsi yang berjudul:

**PENGUJIAN KEAMANAN SECARA STATIS DAN DINAMIS TERHADAP
APLIKASI MOBILE DI PT INDOMOBIL FINANCE INDONESIA**

**STATIC AND DYNAMIC APPLICATION SECURITY TESTING ON
MOBILE APPLICATION AT PT. INDOMOBIL FINANCE INDONESIA**

**adalah benar hasil karya saya dan belum pernah diajukan sebagai karya
ilmiah, sebagian atau seluruhnya, atas nama saya atau pihak lain**

Dhia Hauzan Muafa

2301926245

Disetujui oleh Pembimbing

Saya setuju Skripsi tersebut layak diajukan untuk Sidang Skripsi

Rini Wongso, S.Kom., M.T.I

D4730

2 Februari 2023

Renandho Imanuel S.Kom

Site Supervisor

2 Februari 2023

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa karena telah melimpahkan rahmat-Nya dan memberikan kesempatan kepada Penulis untuk dapat menyelesaikan skripsi yang berjudul **“PENGUJIAN KEAMANAN SECARA STATIS DAN DINAMIS TERHADAP APLIKASI MOBILE DI PT INDOMOBIL FINANCE INDONESIA”**. Adapun skripsi ini diajukan untuk memenuhi syarat kelulusan jenjang studi Strata-1 pada Universitas Bina Nusantara.

Penulis menyadari bahwa pengerjaan skripsi ini tidak mungkin terjadi tanpa adanya bantuan dan bimbingan dari berbagai pihak. Maka dari itu, pada kesempatan ini Penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung terhadap proses penyusunan skripsi ini. Terima kasih penulis ucapkan kepada:

1. Bapak **Prof. Ir. Harjanto Prabowo, M.M.**, selaku Rektor Universitas Bina Nusantara,
2. Bapak **Fredy Purnomo, S.Kom., M.Kom.**, selaku *Dean of School of Computer Science* Universitas Bina Nusantara,
3. Bapak **Yohan Muliono, S.Kom., M.TI., CND, CEHmaster, eMAPT**, selaku *Head of Cyber Security* Universitas Bina Nusantara,
4. Ibu **Rini Wongso, S.Kom., M.T.I** selaku dosen pembimbing *internship* dan skripsi dari Universitas Bina Nusantara yang telah membimbing penulis dari awal pelaksanaan penyusunan skripsi sekaligus *internship* pada semester ganjil.
5. Bapak **Renandho Imanuel, S.Kom., IT Infrastructure & Development Department Head**, selaku *supervisor* dari perusahaan PT Indomobil Finance Indonesia yang telah membimbing selama kegiatan *internship* berlangsung,
6. serta seluruh staf dan mentor di PT. Indomobil Finance Indonesia dan Universitas Bina Nusantara yang telah memberikan bimbingan, ilmu, dan bantuan selama penyusunan skripsi berlangsung.

Penulis berharap penelitian ini dapat memberikan informasi yang jelas dan dapat dipahami dengan baik oleh para pengembang aplikasi PT. Indomobil Finance Indonesia terkait keamanan aplikasi baik aplikasi yang diuji saat ini maupun pada pengembangan aplikasi kedepannya.

Jakarta, 17 Januari 2022

Dhia Hauzan Mu’afa

DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	v
DAFTAR TABEL	viii
DAFTAR GAMBAR.....	ix
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	3
1.3 Ruang Lingkup.....	3
1.4 Tujuan dan Manfaat	4
1.5 Metodologi.....	5
1.5.1 Metode Pengumpulan Data.....	5
1.5.2 Metode Pengujian Keamanan	5
1.6 Sistematika Penulisan	6
BAB 2 TINJAUAN REFERENSI.....	7
2.1 <i>Internet</i>	7
2.2 <i>Network</i>	7
2.3 <i>Server</i>	8
2.3.1 Database.....	8
2.3.2 <i>Application Programming Interface (API)</i>	8
2.4 <i>Smartphone</i>	8
2.4.1 Android	9
2.4.2 Root.....	9
2.4.3 APK.....	9
2.5 Flutter	10
2.6 <i>System Security</i>	10
2.6.1 <i>Penetration Testing</i>	10
2.6.2 <i>Common Vulnerability Scoring System (CVSS)</i>	14
2.6.3 <i>Vulnerability</i>	14
2.6.4 <i>Threat</i>	14
2.6.5 <i>Risk</i>	15
2.6.6 <i>Exploit</i>	15

2.6.7	<i>Reverse Engineering</i>	15
2.7	OWASP.....	16
2.7.1	<i>OWASP Mobile Application Security Testing Guide (MASTG)</i>	16
2.7.2	<i>OWASP Mobile Application Security Verification Standard (MASVS)</i>	17
2.7.3	<i>OWASP Mobile Penetration Testing Checklist</i>	18
2.8	ADB	26
2.9	Memu	26
2.10	Frida	27
2.11	Reflutter	27
2.12	Uber-Signer.....	27
2.13	MobSF.....	27
2.14	Burp Suite	28
2.15	Apk-Tool.....	28
2.16	Radare2	28
BAB 3 DESKRIPSI UMUM		29
3.1	Latar Belakang Perusahaan.....	29
3.1.1	<i>Informasi Umum Perusahaan</i>	29
3.1.2	<i>Sejarah Perusahaan</i>	29
3.1.3	<i>Struktur Organisasi Perusahaan</i>	31
3.2	Peran Penulis dalam Perusahaan.....	31
3.3	Kondisi Saat Ini.....	32
3.4	Identifikasi Masalah.....	33
3.5	Solusi yang Diusulkan	33
3.6	Ruang Lingkup Pengujian Keamanan.....	34
BAB 4 HASIL DAN PEMBAHASAN.....		35
4.1	Deskripsi Target Pengujian.....	35
4.2	Status Keamanan Aplikasi	35
4.3	Rencana Implementasi Solusi	36
4.4	Hasil Pengujian	37
4.4.1	<i>MASTG-ARCH: Architecture, Design and Threat Modeling Requirements</i>	37
4.4.2	<i>MASTG-STORAGE: Data Storage and Privacy Requirements</i>	47
4.4.3	<i>MASTG-CRYPTO: Cryptography Requirements</i>	61
4.4.4	<i>MASTG-AUTH: Authentication and Session Management Requirements</i>	62
4.4.5	<i>MASTG-NETWORK: Network Communication Requirements</i>	70
4.4.6	<i>MASTG-PLATFORM: Platform Interaction Requirements</i>	74
4.4.7	<i>MASTG-CODE: Code Quality and Build Setting Requirements</i>	87
4.4.8	<i>MASTG-RESILIENCE: Resilience Requirements</i>	98

4.5	Evaluasi Hasil Pengujian	113
4.5.1	Tabel Status Pengujian.....	113
4.5.2	Ringkasan Hasil Pengujian	121
4.5.3	Evaluasi Dengan Wawancara Terhadap Hasil Pengujian	122
BAB 5 KESIMPULAN DAN SARAN.....	125	
5.1	KESIMPULAN.....	125
5.2	SARAN	126
REFERENSI.....	129	
LAMPIRAN.....	131	
RIWAYAT HIDUP.....	132	

DAFTAR TABEL

Table 2.1 <i>Architecture, Design and Threat Modeling Requirements</i>	18
Table 2.2 <i>Data Storage and Privacy Requirements</i>	19
Table 2.3 <i>Architecture, Design and Threat Modeling Requirements</i>	20
Table 2.4 <i>Authentication and Session Management Requirements</i>	21
Table 2.5 <i>Network Communication Requirements</i>	22
Table 2.6 <i>Platform Interaction Requirements</i>	23
Table 2.7 <i>Code Quality and Build Setting Requirements</i>	24
Table 2.8 <i>Resilience Requirements</i>	25
Table 4.1 Evaluasi Kategori <i>Architecture, Design, and Threat Modelling</i>	113
Table 4.2 Evaluasi Kategori <i>Data Storage and Privacy</i>	114
Table 4.3 Evaluasi Kategori <i>Cryptography</i>	115
Table 4.4 Evaluasi Kategori <i>Authentication and Session Management</i>	116
Table 4.5 Evaluasi Kategori <i>Network Communication</i>	117
Table 4.6 Evaluasi Kategori <i>Platform Interaction</i>	118
Table 4.7 <i>Code Quality and Build Setting</i>	119
Table 4.8 Evaluasi Kategori <i>Resilience</i>	120
Table 4.9 Ringkasan Hasil Pengujian	121

DAFTAR GAMBAR

Gambar 2.1 Struktur penggunaan OWASP	16
Gambar 3.1 Logo Perusahaan PT Indomobil Finance Indonesia.....	29
Gambar 3.2 Struktur Organisasi PT Indomobil Finance Indonesia.	31
Gambar 4.1 Isi File AndroidManifest.xml Aplikasi dari Tools MobSF.....	37
Gambar 4.2 Daftar Aktivitas dan Layanan Aplikasi dari Tools MobSF.....	38
Gambar 4.3 Pengujian SQL Injection Pada Field Nama Karyawan.	39
Gambar 4.4 Bukti hasil SQL Injection pada Database.	39
Gambar 4.5 Import Library Untuk Mengecek Version Code Aplikasi.....	44
Gambar 4.6 Pengecekan Version Code pada Source Code.....	44
Gambar 4.7 Tampilan Pop Up Force Update pada Aplikasi.....	45
Gambar 4.8 Permission pada Folder Package Aplikasi.	48
Gambar 4.9 Status Keamanan Enkripsi Data Sensitif pada File SharedPreferences.	48
Gambar 4.10 Storage Permission pada File AndroidManifest.xml	49
Gambar 4.11 Penggunaan Permission External Storage pada Aplikasi.....	49
Gambar 4.12 Log Aplikasi pada Debug Console di VSCode.....	50
Gambar 4.13 Tampilan Keyboard Cache (di Atas Keyboard).....	51
Gambar 4.14 Nilai Permission Export untuk Provider pada File AndroidManifest.xml	52
Gambar 4.15 Input Field pada Halaman Login.....	53
Gambar 4.16 Input Field pada Halaman Lupa Password.....	53
Gambar 4.17 Penggunaan Command “Adb Backup” pada Aplikasi.....	54
Gambar 4.18 Unpack File Backup Aplikasi Menggunakan abe.jar.....	55
Gambar 4.19 Isi dari File Backup SharedPreferences	55
Gambar 4.20 Penggunaan Package yang Memiliki Fitur FLAG_SECURE.	55
Gambar 4.21 Isi Package Window Manager pada Flutter.	56
Gambar 4.22 Tampilan View pada Halaman Peraturan Perusahaan Saat Dipindahkan ke Background.....	56
Gambar 4.23 Dump Memory Menggunakan Objection pada Aplikasi	57
Gambar 4.24 Isi File Memory yang Telah Diambil dalam Bentuk Teks.....	58
Gambar 4.25 Log Percobaan Login (Kiri) dan Isi File SharedPreferences (Kanan)	61
Gambar 4.26 Intercept Jaringan yang Berisi Otentikasi pada Aplikasi.	62
Gambar 4.27 Intercept Jaringan pada Halaman Ubah Password Untuk Policy Minimal 6 Karakter.	64
Gambar 4.28 Intercept Jaringan pada Halaman Ubah Password untuk Policy Mengandung Karakter Special, Huruf Besar dan Alphanumeric.....	64
Gambar 4.29 Uji Bruteforce pada Halaman Login Menggunakan Burpsuite.....	65
Gambar 4.30 Intercept Request pada Halaman Approver.	67
Gambar 4.31 Respon yang Berisi Daftar Authority yang Dimiliki Setiap User.....	69
Gambar 4.32 Daftar History Hasil Intercept yang Berisi Host dari API.....	70
Gambar 4.33 Intercept Request pada Halaman Lupa Password.	73
Gambar 4.34 Daftar Permission pada File AndroidManifest.xml	74
Gambar 4.35 Daftar Activity, Service, Receiver dan Provider yang Terexport.	76
Gambar 4.36 Pengecekan IPC yang memiliki nilai android:exported=true.....	76

Gambar 4.37 Nilai Webview Setting untuk Penggunaan Javascript dari File Java Hasil Reverse Engineering.....	77
Gambar 4.38 Nilai String dari Variable f61991 yang Digunakan pada Gambar 4.37	77
Gambar 4.39 Nilai Javascript Mode pada File .dart untuk Class Webview yang Dipanggil.	77
Gambar 4.40 Daftar Host dari History Intercept pada Halaman Insentif yang Menggunakan Webview.....	78
Gambar 4.41 Respon dari Intercept Request pada Halaman Webview.....	80
Gambar 4.42 Percobaan Tamper pada Script untuk Webview.....	80
Gambar 4.43 Hasil Percobaan Tamper pada Halaman Insentif.....	81
Gambar 4.44 Keadaan Awal Saat Nilai Key yang Dikirim Benar pada Halaman iBulletin..	82
Gambar 4.45 Keadaan Setelah Dilakukan Pengubahan Pada Nilai Key yang Dikirim ke Halaman iBulletin.....	82
Gambar 4.46 Dump Memory untuk Mengecek String dari Script yang Tadi Ditambahkan (subbab 4.2.6.7).....	84
Gambar 4.47 Isi File Cache Webview pada Local Storage.....	84
Gambar 4.48 Isi File Cookies Webview pada Local Storage.....	84
Gambar 4.49 Penggunaan Keyboard dari Custom Third-Party.....	86
Gambar 4.50 Status Penandatanganan Aplikasi pada Skema v1, v2 dan v3 dari Tool MobSF.	87
Gambar 4.51 Status Signer dan Certificate pada Setiap File dengan Tool Jarsigner.....	87
Gambar 4.52 Isi File AndroidManifest.xml untuk Mencari Nilai Android:debuggable.....	88
Gambar 4.53 Penggunaan Command run-as Pada Adb Shell untuk Status Debug Aplikasi.	88
Gambar 4.54 Isi Native Debug Symbol pada File Shared Libraries .so.	89
Gambar 4.55 Menampilkan Log dengan Adb Logcat untuk Mencari Error atau Debugging Message.	91
Gambar 4.56 Hasil Penggunaan Plugin OWASP Dependency Check.	92
Gambar 4.57 Tampilan Bentuk Awal Hasil Respon dari Login.	93
Gambar 4.58 Pengujii Mencoba Tamper Respon dengan Respon Sukses.	93
Gambar 4.59 Tampilan Aplikasi Setelah Tamper Respon Saat Login.	94
Gambar 4.60 SQL Injection dengan Mengandalkan Error-Based Respon.	95
Gambar 4.61 Tamper Request dengan Mengirimkan Data atau File Berekstensi .exe.....	95
Gambar 4.62 Memory Page pada VS Code Terhadap Aplikasi yang Sedang Berjalan.....	96
Gambar 4.63 Isi File Config untuk Mengecek Nilai Minimal SDK Device.....	97
Gambar 4.64 Mengecek Nilai Protection Canary dan Pic dengan r2 (radare2).	98
Gambar 4.65 Keadaan Aplikasi Saat Status Device Tidak Root.	99
Gambar 4.66 Keadaan Aplikasi Saat Status Device Root.....	99
Gambar 4.67 Keadaan Aplikasi Sebelum Nilai NoDataKaryawan Pada File SharedPreference Diubah.....	101
Gambar 4.68 Intercept Request untuk Mengecek Nilai NoDataKaryawan yang Digunakan oleh Aplikasi.	101
Gambar 4.69 Tamper Isi File AndroidManifest.xml lalu Rebuild aplikasi.	101
Gambar 4.70 Menjalankan Command Run-as Menggunakan Adb Shell.	102
Gambar 4.71 Nilai Android:debuggable pada AndroidManifest.xml Melalui Reverse Engineering.....	102

Gambar 4.72 Membuka File Mapped Memory pada Process Id IMFI One.	103
Gambar 4.73 Menjalankan Aplikasi dengan Emulator.....	104
Gambar 4.74 Command Untuk Melakukan Hook Method Dengan Frida.	105
Gambar 4.75 Script Untuk Melakukan Hook Method.	105
Gambar 4.76 Hasil Tes Injeksi Script Dengan Hook Method.	106
Gambar 4.77 Source Code Java Hasil Reverse Engineering.	108
Gambar 4.78 Menyalin Data Cache dan SharedPreferences.	109
Gambar 4.79 Memindahkan File Cache dan SharedPreference ke Emulator Device Baru.	110
Gambar 4.80 File Library dari Source Code Hasil Reverse Engineering.	111

DAFTAR LAMPIRAN

Lampiran 1 Bukti Wawancara Evaluasi Pengujian Keamanan.....131

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Seiring berkembangnya jaman, penggunaan teknologi seperti perangkat komputer dan *smartphone* sudah menjadi bagian dari kehidupan sehari-hari manusia. Menurut laporan yang diterbitkan oleh Stock Apps, perangkat *smartphone* telah dilaporkan memiliki jumlah pengguna di dunia mencapai 5,3 miliar pada bulan juni 2021. Jumlah tersebut menunjukan bahwa persentase pengguna *smartphone* memiliki jumlah sebanyak 67% jika dibandingkan dengan perkiraan total populasi penduduk Bumi yang mencapai 7,9 miliar (S, 2022). Perkembangan *smartphone* ini tidak dipungkiri telah memberikan dampak yang sangat besar ke berbagai sektor bisnis yang meliputi komunikasi, hiburan, keuangan, kesehatan, bisnis jual beli, dan lainnya. Hal ini membuat maraknya pembuatan aplikasi berbasis *mobile* pada berbagai macam sektor bisnis tersebut, khususnya pada *platform android*.

Bersamaan dengan berkembangnya jumlah pengguna *smartphone* khususnya pada sistem operasi *android* menyebabkan aplikasi pada *android* menjadi target kejahatan utama oleh *cracker* dan *hacker*. *Hacker* adalah seorang peretas yang masuk ke dalam sistem aplikasi dengan memanfaatkan celah keamanan yang berada pada sistem tersebut. Sedangkan, *Cracker* adalah seorang peretas sistem aplikasi yang tidak hanya masuk ke dalam sistem aplikasi melainkan juga melakukan eksplorasi terhadap kelemahan aplikasi dan memanfaatkannya untuk kepentingan pribadi (Richet, 2013). Hal tersebut menimbulkan pertanyaan pengguna terkait bagaimana penyedia layanan aplikasi tersebut menjaga kerahasiaan, ketersediaan, dan keaslian dari data pengguna yang digunakan di dalam aplikasi. Hal ini didasarkan pada banyaknya serangan siber pada dunia digital yang merugikan semua pihak baik penyedia layanan dan juga pengguna layanan aplikasi tersebut. Sebagai contoh ancaman dan serangan siber yang umum terjadi pada perangkat *mobile* adalah serangan *phising*, *man-in-the-middle* (MITM), *malware* dan penggunaan aplikasi ilegal.

Masalah terkait kerahasiaan data pribadi di Indonesia akhir-akhir menjadi perhatian yang semakin meningkat dikarenakan pemerintah dan perusahaan swasta yang mulai mengumpulkan, menggunakan dan memproses data pribadi

untuk kepentingan tertentu. Hal ini menyebabkan munculnya permasalahan yang terjadi karena masalah tersebut seperti: (1) munculnya pengaduan baik individu maupun kelompok terhadap pelanggaran informasi pribadi terkait terganggunya privasi data pribadi individu baik melalui media cetak maupun elektronik; (2) munculnya keluhan dari pengguna karena identitas dan privasi mereka pada data pribadi tidak disimpan dengan baik, misalnya di industri perbankan, atau lebih khusus di industri perkreditan, di mana privasi dari pelanggannya dapat diakses, disebarluaskan dan dibagikan antar lembaga mereka tanpa sepengetahuan pelanggan.

Banyaknya masalah dan ancaman terhadap aplikasi *android* membuat banyak pengembang aplikasi mulai mengembangkan alat pengujian keamanan aplikasi baik secara statis dan dinamis. Pengujian ini bertujuan untuk mengetahui apakah aplikasi sudah sesuai dengan aspek dan standar keamanan ISO/IEC 27001 (ISO, 2022). Hal ini berarti pengujian terhadap aplikasi secara statis dan dinamis itu sangatlah penting untuk mengetahui apakah keamanan aplikasi sudah sesuai dengan standar keamanan yang berlaku. Pengujian secara statis yaitu pengujian dengan cara melakukan analisa pada *source code*, dokumentasi dan lainnya tanpa menjalankan aplikasi. Sedangkan, pengujian secara dinamis adalah pengujian keamanan dilakukan saat aplikasi sedang berjalan secara *real time* (Pan, 2019).

PT. Indomobil Finance Indonesia merupakan perusahaan yang bergerak di bidang bisnis jasa pembiayaan motor, mobil, alat berat, kendaraan niaga, properti dan pembiayaan mikro dengan bentuk pembiayaan konsumen, sewa guna usaha dan anjak piutang. Sebagai salah satu penyedia layanan aplikasi *mobile* yang dimasukan ke dalam aplikasi *playstore*, PT. Indomobil Finance Indonesia juga memiliki tanggung jawab dalam memproses, menggunakan, dan menjaga keamanan data pribadi penggunanya dengan baik sebagai upaya meminimalisir terjadinya eksploitasi dan serangan siber yang memanfaatkan data pribadi pengguna. Serangan ini sering kali terjadi tanpa disadari dan memerlukan waktu lama untuk menanganinya, karena pada dasarnya serangan ini juga dipicu oleh pengguna yang memberikan kesempatan bagi penyerang untuk masuk ke dalam sistem aplikasi.

1.2 Rumusan Masalah

Berdasarkan kondisi di perusahaan saat ini, maka dapat dirumuskan beberapa poin masalah yang akan diangkat yaitu:

1. Seberapa amankah aplikasi IMFI One terhadap terjadinya serangan-serangan dunia digital?
2. Apakah aplikasi IMFI One sudah memenuhi ketentuan keamanan berdasarkan panduan OWASP MASTG dan MASVS?
3. Bagaimana tingkat risiko dan seberapa parah risiko terhadap setiap kelemahan yang ada pada aplikasi IMFI One?
4. Dimana saja bagian dari *code* atau aplikasi yang memiliki kerentanan terhadap kemungkinan terjadinya serangan?
5. Bagaimana tindakan yang harus dilakukan sebagai pencegahan atau penanggulangan kerentanan yang ditemukan?
6. Apa saja akibat yang dapat terjadi jika kerentanan pada aplikasi tidak dapat dicegah atau diatasi?

1.3 Ruang Lingkup

Ruang lingkup pada penggerjaan skripsi sebagai penyelesaian masalah adalah sebagai berikut:

1. Ruang lingkup penulisan dan fokus utama penyelesaian masalah adalah pada pengujian keamanan dengan penetration testing.
2. Penulisan dipusatkan kepada penjelasan dan penjabaran dari status keamanan dan kelemahan yang ditemukan pada aplikasi.
3. Pembuatan dan penyediaan materi pendukung dibuat sebagai bentuk penjelasan terhadap hasil tulisan, metode pengujian, serta maksud dari poin pengujian.
4. Lingkup penggerjaan skripsi dilakukan dengan fokus acuan yang dibuat oleh OWASP baik dari penggunaan *tools*, metode, *requirement*, dan lain-lain.

1.4 Tujuan dan Manfaat

Tujuan:

1. Melakukan pengujian keamanan pada aplikasi testing IMFI One.
2. Menemukan dan membuat daftar kerentanan-kerentanan pada aplikasi IMFI One sebagai acuan developer untuk memperkuat keamanan aplikasi IMFI One.
3. Melakukan analisis untuk menjabarkan kerentanan beserta seberapa parah tingkat risikonya.
4. Memberikan saran dan solusi untuk mencegah dan melakukan perbaikan sebelum terjadinya eksploitasi terhadap kelemahan aplikasi.
5. Berkolaborasi langsung dengan pihak developer aplikasi IMFI One untuk menerapkan hasil pengujian keamanan.

Manfaat:

1. Mengetahui tingkat keamanan pada aplikasi IMFI One.
2. Meningkatkan kualitas aplikasi dari segi keamanan yang sedang atau akan dibuat kedepannya karena acuan ini tidak hanya bisa dipakai untuk aplikasi sekarang namun dapat digunakan kembali untuk kedepannya.
3. Mengurangi dan mencegah kemungkinan terjadinya serangan pada aplikasi *mobile* IMFI One.
4. Meminimalkan kemungkinan terjadinya kerugian yang tidak diinginkan oleh perusahaan baik secara materi maupun non materi.
5. Menjaga kepercayaan pengguna terhadap perusahaan dalam menjaga aset seperti identitas pribadi dari orang yang tidak memiliki wewenang.

1.5 Metodologi

1.5.1 Metode Pengumpulan Data

1.5.1.1 Observasi

yaitu pengumpulan data dengan melakukan pengamatan langsung terkait cara kerja dari aplikasi IMFI One. Data diperoleh dengan menjalankan aplikasi IMFI One dan menggunakan fitur-fitur yang ada pada aplikasi dengan tujuan mendapatkan pengetahuan dan perkiraan apa saja fitur yang kemungkinan memiliki celah pada keamanannya.

1.5.1.2 Studi Pustaka

yaitu pengumpulan data dengan cara membaca, mempelajari informasi yang dapat dimanfaatkan untuk melakukan pengujian keamanan. Informasi yang dikumpulkan dan dipelajari yaitu dokumentasi aplikasi IMFI One dan referensi dari berbagai jenis buku dan jurnal terkait penelitian, metode dan perangkat yang digunakan. Dengan metode ini, informasi yang didapat sebelum pengujian dapat berguna untuk mempelajari struktur aplikasi, contoh kasus yang sejenis, dan penyelesaian masalah-masalah yang terjadi.

1.5.2 Metode Pengujian Keamanan

1.5.2.1 SAST

SAST atau *Static Application Security Testing* adalah sebuah metode pengujian keamanan secara *white box*. *White box penetration testing* dilakukan ketika penguji keamanan memiliki informasi sebanyak mungkin terkait sistem dari target yang akan diuji. Penguji biasanya akan diberikan informasi seperti akses admin, *source code* dan dokumentasi dari aplikasi. Dengan metode ini, penguji tidak menjalankan aplikasi tetapi langsung melakukan analisis terhadap seluruh informasi internal dari aplikasi tersebut. SAST biasanya memerlukan waktu yang lebih lama untuk menganalisis informasi namun dapat memberikan laporan yang lebih komprehensif terkait kelemahan dari sistem.

1.5.2.2 DAST

DAST atau *Dynamic Application Security Testing* adalah sebuah metode pengujian keamanan yang menggunakan pendekatan *black box*. *Black box penetration testing* dilakukan dengan asumsi penguji tidak memiliki pengetahuan apapun

terkait informasi internal dari aplikasi yang akan dites. Dengan menggunakan DAST, penguji melakukan *penetration testing* saat aplikasi dijalankan secara *real time*. Hal ini berarti penguji dapat mendeteksi ancaman serangan yang lebih luas, termasuk kebocoran memori, serangan *cross-site scripting*, *SQL injection*, masalah otentikasi dan enkripsi. Selain itu, DAST juga dapat digunakan untuk mengetes sistem apapun, *endpoint API* atau *web service* yang terhubung ke aplikasi, infrastruktur dan *host system (networking, storage)*.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini adalah sebagai berikut:

1. BAB 1: PENDAHULUAN

Bab ini berisi latar belakang dari pengujian, rumusan masalah, ruang lingkup pengujian, tujuan dan manfaat pengujian, metode pengujian, dan sistematika penulisan.

2. BAB 2: TINJAUAN REFERENSI

Bab ini berisi teori-teori yang berkaitan dengan pembahasan dalam penelitian. Adapun teori yang dijelaskan meliputi *penetration testing*, metodologi yang digunakan pada skripsi, serta sistem operasi dan *tools* berupa emulator dan *software* lainnya yang digunakan pada proses pengujian.

3. BAB 3: DESKRIPSI UMUM

Bab ini berisi penjabaran latar belakang perusahaan, kondisi saat ini, identifikasi masalah, solusi yang diusulkan, ruang lingkup solusi tersebut, serta peran penulis pada proyek.

4. BAB 4: HASIL DAN PEMBAHASAN

Bab ini berisi deskripsi dari target yang diuji, status keamanan aplikasi saat ini, rencana implementasi solusi, hasil pengujian, dan evaluasi pengujian.

5. BAB 5: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari pembahasan penelitian dan saran tentang hasil pengujian.

BAB 2

TINJAUAN REFERENSI

2.1 Internet

Internet adalah sekumpulan jaringan besar yang menghubungkan beberapa jaringan di seluruh dunia melalui jalur telekomunikasi seperti telepon, radio, satelit, *link* dan lainnya. Sejarah tentang internet pertama kali berasal dari Departemen Pertahanan Amerika Serikat di tahun 1969 melalui proyek yang disebut ARPANET (*Advanced Research Project Agency Network*) dimana mereka bisa membuat komunikasi antar komputer berbasis UNIX melalui saluran telepon. (Gani, 2018)

Jaringan internet berkomunikasi satu sama lain berdasarkan *protocol* tertentu seperti *Transmission Control Protocol* (TCP) dan *Internet Protocol* (IP). Saat informasi pertama kali dikirim melalui internet, TCP akan memecah informasi menjadi beberapa paket. Komputer lalu akan mengirim paket tersebut melalui *local network*, *Internet Service Provider* (ISP) atau layanan *online*. Dari sana, paket tersebut akan berjalan ke banyak tingkatan jaringan, komputer dan komunikasi sebelum mencapai tujuan akhir. Data atau informasi yang berjalan tersebut sebelumnya telah ditentukan rutenya oleh *Internet Protocol* (IP). (T.Campbell, 2014)

2.2 Network

Network atau jaringan adalah sekumpulan “interkoneksi” atau hubungan antara 2 perangkat atau lebih yang terhubung dengan media transmisi kabel atau tanpa kabel (*wireless*). Dua perangkat dapat dikatakan terkoneksi satu sama lain apabila keduanya bisa saling berkomunikasi dan bertukar informasi, data maupun *resource* lainnya. Perangkat yang dapat berkoneksi satu sama lain seperti komputer dengan printer, media penyimpanan (*harddisk*, *flash disk*) maupun antar perangkat yang sama seperti komputer ke komputer atau *smartphone* ke *smartphone* lainnya. (Syafrizal, 2020)

2.3 Server

2.3.1 Database

Database merupakan satu atau lebih komputer yang terintegrasi dan terhubung yang dapat menampung sekumpulan data yang memiliki relasi antara satu dengan lainnya. *Database* umumnya menampung gabungan dari *end-user-data* dan meta data. Sedangkan *Database Management System* (DBMS) adalah sekumpulan program yang mengurus struktur database dan mengontrol akses ke data yang disimpan ke dalam *database*. (Coronel & Morris, 2016)

2.3.2 Application Programming Interface (API)

API atau *Application Programming Interface* adalah suatu teknologi antarmuka yang dibangun oleh pengembang sistem untuk memungkinkan terjadinya fungsi - fungsi sistem seperti pertukaran informasi antara aplikasi dengan server sebagai *endpoint* dapat diakses secara terprogram. *Representational State Transfer* (REST) adalah salah satu gaya arsitektur pengembangan API yang menggunakan *Hypertext Transfer Protocol* (HTTP) untuk data komunikasi. (Pranata, Hijriani, & Junaidi, 2018)

2.4 Smartphone

Smartphone adalah sebuah perangkat komunikasi digital yang dapat mengerjakan tugas atau memiliki fungsi mirip seperti komputer dan telepon genggam pada umumnya. *Smartphone* memiliki *interface* berupa layer sentuh, memiliki akses internet, sistem operasi yang dapat mengunduh dan menjalankan berbagai aplikasi. *Smartphone* dapat menjalankan fitur-fitur umum pada telepon genggam seperti mengirim dan menerima pesan atau email, kalkulator, merekam suara, memainkan music, dan lainnya. (Whyte, 2019)

2.4.1 *Android*

Android adalah sebuah sistem operasi berbasis linux yang didesain dari awal sebagai platform *open source* dan dirancang untuk perangkat *mobile*. Android dalam sejarahnya dikembangkan oleh Android, Inc. Namun sekarang sudah berpindah ke tangan Google yang kemudian dirilis sebagai AOSP (*Android Open Source Project*) pada tahun 2007. Hal tersebut bersamaan dengan didirikannya OHA (*Open Handset Alliance*) sebuah grup dari beberapa perusahaan telecom, software, hardware seperti Google, Intel, NVIDIA, dan lainnya yang didedikasikan untuk mengembangkan dan mendistribusikan Android. (Gilski & Stefanski, 2015)

2.4.2 *Root*

Root adalah proses sah yang dapat dilakukan oleh pemilik perangkat secara sukarela untuk mendapatkan hak atau *privilege* tertinggi dan *control* penuh atas perangkat pengguna. Pada linux, *command* su merujuk kepada “*subtitue user*”, “*super user*”, atau “*switch user*” dari pengguna ke root di Android. *Command* tersebut menyebabkan proses yang hanya dapat dilakukan oleh Administrator atau root dapat dilakukan. Tujuan dari melakukan root kepada Android sendiri adalah untuk memasang dan memungkinkan aplikasi proses aplikasi apapun untuk dilakukan. (Shao, Luo, & Qian, 2014)

2.4.3 *APK*

Sebuah aplikasi pada dasarnya adalah sebuah *application package* (APK) *file*. Android *package* dengan *extension apk* adalah format sebuah *file* yang digunakan pada sistem operasi android dan sistem operasi lain yang berbasis android dalam mendistribusikan dan memasang aplikasi *mobile*, *game* dan *middleware*. Sebuah APK *file* memiliki 3 komponen utama yaitu:

1. Dalvik executable

Disini semua *source code* yang dibuat menggunakan Java *decompile* menjadi Dalvin *executable*. Disinilah juga kode-kode yang menjalankan aplikasi berada.

2. Resources

Resources adalah semua komponen selain kode. Pada aplikasi terkadang terkandung beberapa gambar, audio, video serta beberapa XML *file* yang mendeskripsikan *layout*, paket Bahasa, dan sebagainya.

3. Native Libraries

Native code seperti *library-library* dari C/C++ terkadang juga terkandung dalam APK *file*. (Gargenta, 2011)

2.5 Flutter

Flutter adalah sebuah *framework* multi platform yang menargetkan pengembangan aplikasi seluler. Flutter dirilis secara publik pada tahun 2016 oleh Google. Selain Android dan iOS, aplikasi yang terbuat dari flutter juga dapat berjalan di sistem operasi buatan Google yaitu Fuchsia. Flutter menggunakan mesin *rendering* berperforma tinggi untuk melakukan *render* setiap komponen tampilan menggunakan miliknya sendiri. Hal ini memberikan kesempatan untuk pembuatan aplikasi yang berkinerja tinggi seperti yang bisa dilakukan aplikasi *native*. (Tashildar, Shah, Gala, Giri, & Chavhan, 2020)

2.6 System Security

2.6.1 Penetration Testing

Penetration testing adalah salah satu metode yang dilakukan untuk mengevaluasi keamanan infrastruktur pada sebuah teknologi dengan mencari kerentanan yang berada di infrastruktur tersebut. *Penetration* testing dapat dilakukan secara berkala untuk mengidentifikasi dan *manage* risiko sebagai upaya untuk memperkuat pertahanan dari sebuah infrastruktur. (Shebli & Beheshti, 2018)

2.6.1.1 Tipe *Penetration Testing*

1. *Black Box*

Dalam pengujian menggunakan Teknik *black box*, penguji tidak memiliki pengetahuan tentang arsitektur jaringan dan sistem targetnya. Biasanya pengujian ini dilakukan dari jaringan eksternal ke jaringan internal. Dalam pengujian ini penguji harus memiliki keahlian dan keterampilan untuk melakukan pengujian terhadap aplikasi. (Goel & Mehtre, 2015)

2. *White Box*

Dalam pengujian menggunakan Teknik *white box*, penguji memiliki pengetahuan lengkap tentang konfigurasi dan struktural dari jaringan dan sistem seperti dapat memiliki akses ke dalam *source code*, dokumentasi, dan lainnya. Biasanya pengujian ini dilakukan melalui jaringan internal. Pengujian ini membutuhkan pemahaman mendalam terkait jaringan atau sistem yang diuji agar dapat memberikan hasil yang lebih baik (Goel & Mehtre, 2015)

3. *Grey Box*

Dalam pengujian menggunakan Teknik *grey box*, penguji memiliki Sebagian pengetahuan tentang sistem atau jaringan yang diuji. Penguji tidak memiliki pengetahuan tentang arsitektur secara lengkap namun penguji mengetahui beberapa informasi dasar pengujian jaringan dan konfigurasi sistem. Pengujian ini adalah kombinasi dari kedua Teknik lainnya dan dapat dilakukan dari jaringan internal maupun eksternal. (Goel & Mehtre, 2015)

2.6.1.2 Standar Tahapan *Penetration Testing*

Standar yang digunakan dalam melaksanakan *penetration testing* adalah *Penetration Testing Execution Standard* (PTES) yang dirilis pada tahun 2009. Standar PTES ini terdiri dari 7 tahapan (Chebbi, 2018), yaitu:

a. *Pre-engagement Interactions*

Tahapan ini memiliki tujuan untuk menyediakan dan menjelaskan alat dan teknik yang tersedia untuk membantu kesuksesan pelaksanaan *penetration testing*. Pada tahapan ini akan ditentukan perkiraan berapa lama penguji akan melakukan *penetration testing* serta cakupan atau batasan dari aplikasi yang akan dilakukan testing.

b. *Intelligence Gathering*

Tahapan ini memiliki tujuan untuk mengumpulkan informasi terkait aplikasi yang akan dilakukan testing sebanyak mungkin. Data yang dikumpulkan nantinya akan digunakan sebagai bantuan dalam melakukan uji eksploitasi pada aplikasi testing. Pengumpulan data biasanya dilakukan menggunakan *open-source intelligence* (OSINT).

c. *Threat Modeling*

Tahapan ini dilakukan dengan membuat sebuah model berisikan potensi ancaman terhadap aplikasi dan aset apa saja yang ingin dilindungi dari potensi ancaman tersebut. Modeling dilakukan dengan membagi 2 aspek pada aset dan potensi ancaman yaitu aset memiliki aspek pada aset dan proses pada bisnis tersebut dan potensi ancaman memiliki aspek agen dari pelaku ancaman dan kemampuannya. Model ini nantinya akan digunakan sebagai panduan untuk memahami lebih lanjut pola pikir dari penyerang, metode yang mungkin digunakan, dan aset yang kemungkinan besar dijadikan target penyerangan.

d. *Vulnerability Analysis*

Menganalisa aplikasi yang dilakukan testing untuk mencari kemungkinan kerentanan yang berada pada aplikasi. Kerentanan yang ditemukan nantinya akan dianalisa untuk memastikan apakah terdapat kemungkinan eksloitasi lebih lanjut atau tidak. Kerentanan pada aplikasi dapat berasal dari *host*, *service misconfiguration* atau *insecure application design*.

e. *Exploitation*

Setelah kerentanan pada aplikasi ditemukan dan dinilai memiliki kemungkinan untuk dieksloitasi lebih lanjut, uji eksloitasi akan dilakukan pada kerentanan tersebut. Kerenantan yang ditemukan akan digunakan untuk mencoba mengakses sistem atau aset tanpa adanya izin dan tanpa terdeteksi dari sistem.

f. *Post Exploitation*

Menentukan nilai dari hasil eksloitasi dan mempertahankan kontrol yang didapatkan dari eksloitasi terhadap aplikasi untuk uji eksloitasi lebih lanjut. Metode pada fase ini digunakan untuk membantu penguji mengidentifikasi dan mencatat apa saja data sensitif, konfigurasi, saluran komunikasi, dan hubungan dengan jaringan perangkat lain untuk mendapatkan akses yang lebih lanjut ke jaringan yang dieksloitasi.

g. *Reporting*

Tahapan ini bertujuan untuk memberikan laporan dalam bentuk tertulis yang berisi tujuan dari *penetration testing* dan temuan yang didapat dari *penetration testing*. Pembaca dari laporan ini ditargetkan untuk klien yang memiliki wewenang dalam pembangunan aplikasi ataupun keamanan pada aplikasi dan anggota organisasi yang berkemungkinan untuk terkena dampak dari ancaman serangan.

2.6.2 Common Vulnerability Scoring System (CVSS)

CVSS merupakan sebuah *framework* terbuka yang dipakai untuk mengukur, mengklasifikasikan karakteristik dan dampak yang dapat ditimbulkan oleh kerentanan yang ada di sistem atau aplikasi. CVSS melakukan pengukuran parah risikonya berdasarkan beberapa parameter perhitungan seperti *Attack Vector* (AV), *Attack Complexity* (AC), *Privileges Required* (PR), *User Interaction* (UI), *Scope* (S), *Confidentiality* (C), *Integrity* (I) dan *Availability* (A). Sedangkan berdasarkan metriknya, CVSS terdiri dari tiga pengelompokan dalam pengukurannya yaitu *Base*, *Temporal* dan *Environmental*. (First, 2015)

1. *Base* sendiri mewakili karakteristik intrinsik dan fundamental dari kerentanan yang konstan sepanjang waktu dan dalam lingkungan *user*.
2. *Temporal*, mewakili karakteristik kerentanan yang berubah sepanjang waktu tetapi tidak dalam lingkungan *user*.
3. *Environmental*, mewakili karakteristik dari kerentanan yang relevan dan unik kepada lingkungan dari *user* tertentu.

2.6.3 Vulnerability

Vulnerability adalah adanya celah atau kelemahan pada keamanan sebuah perangkat lunak dan sistem jaringan yang dapat dieksloitasi oleh peretas secara ilegal untuk mendapatkan keuntungan. Setiap pemilik sistem atau jaringan harus secara rutin melakukan penambalan kerentanan pada sistem mereka untuk meminimalkan dampak dari serangan eksternal. (Shah, Farris, & Jajodia, 2019)

2.6.4 Threat

Threat adalah kemungkinan tindakan jahat yang bertujuan untuk merusak, mencuri data atau mengganggu sistem. *Threat* pada dasarnya berasal dari dua sumber utama yaitu manusia dan alam. *Threat* yang berasal dari manusia disebabkan oleh orang-orang seperti *hacker* atau ancaman internal beserta eksternal yang ingin merusak sistem atau mencuri data. Sedangkan ancaman alam seperti bencana alam yaitu gempa bumi, angin topan, kebakaran yang dapat merusak sistem seperti komputer, server dan tidak ada pencegahan khusus mengenai ini kecuali melakukan hal seperti *back up* data. (Obotivere & Nwaezeigwe, 2020)

2.6.5 *Risk*

Risk pada keamanan siber adalah kemungkinan risiko dari adanya gangguan, kerugian materi dan non materi, kerusakan reputasi perusahaan atau organisasi dalam kegagalan atau kerusakan sistem teknologi dari adanya serangan eksternal dan internal. Contoh risiko *cyber security* termasuk risiko kehilangan data sensitif, gangguan dalam jaringan perusahaan, gangguan pada sistem, gangguan pada layanan, serta kerusakan fisik pada barang elektronik. (Florackis, Louca, Michaely, & Weber, 2020)

2.6.6 *Exploit*

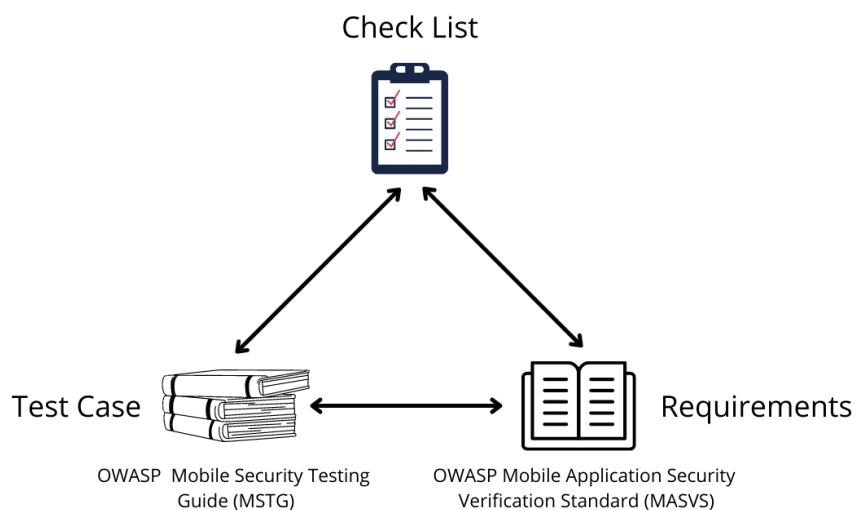
Exploit adalah serangan yang memanfaatkan celah dari sebuah kerentanan pada sistem. Ketika *vulnerability* berarti ada suatu kelemahan pada sistem yang bisa dimanfaatkan untuk kepentingan pelaku, maka *exploit* berarti ada cara pasti yang dapat digunakan untuk mendapatkan keuntungan dengan adanya celah tersebut. (Ginta, Kusum, & Negara, 2013)

2.6.7 *Reverse Engineering*

Pada dasarnya *Reverse Engineering* adalah proses dari pengambilan pengetahuan atau informasi penting dari sebuah produk. Informasi biasanya didapatkan setelah pembongkaran produk asli menjadi beberapa bagian kecil lalu memasangnya kembali. Sebagian besar metode *reverse engineering* berfokus pada penggalian informasi secara otomatis kepada *source code* tanpa memperhitungkan pengetahuan eksternal yaitu dari manusia. Biasanya informasi eksternal yang tersedia adalah sepengetahuan developer padahal kebanyakan sistem perangkat lunak bersifat implisit dan sulit untuk dipulihkan dengan metode *reverse engineering* yang dilakukan secara otomatis. Beberapa informasi penting bahkan terkadang tidak tercantum di *source code* saja dan memerlukan pengetahuan tambahan dari manusia. (Zhigalov & Ivanov, 2019)

2.7 OWASP

Open Web Application Security Project (OWASP) merupakan sebuah Yayasan non-profit yang didirikan dengan tujuan untuk meningkatkan keamanan *software*. Melalui projek *open-sourceny*, OWASP yang merupakan sebuah komunitas terbuka adalah sumber bagi pengembang aplikasi untuk mengamankan dan mengatasi serangan yang terjadi kepada aplikasinya. OWASP selama ini telah ikut serta dalam peranan untuk meningkatkan keamanan aplikasi para pengembang, pelaku bisnis dengan melakukan kegiatan seperti penerbitan artikel, dokumentasi, metodologi, alat dan teknologi dalam bidang keamanan siber. Pada pengujian keamanan kali ini penulis menggunakan *checklist* dan standar *requirement* utama dari OWASP seperti pada gambar 2.1. (OWASP, 2001)



Gambar 2.1 Struktur penggunaan OWASP

2.7.1 OWASP Mobile Application Security Testing Guide (MASTG)

Standar yang sebelumnya dikenal dengan nama MSTG ini digunakan sebagai *manual* dari *testing* keamanan sebuah aplikasi *mobile* untuk panduan pengembangan, pengujian, dan *reverse engineering* keamanan android. Panduan dalam standar MASTG mengandung berbagai prosedur dalam melakukan *penetration testing* dan hal lainnya untuk menilai potensi ditemukannya ancaman pada aplikasi.

2.7.2 *OWASP Mobile Application Security Verification Standard (MASVS)*

Standar MASVS memberikan garis batasan dalam *mobile application security* dengan menggunakan pengukuran ketahanan aplikasi dari serangan (MASVS-L2) dan pelindungan dari ancaman pada sisi klien (MASVS-R). Standar ini digunakan dengan tujuan untuk menyediakan ketentuan pada arsitek *software* dan *developer* yang ingin mengembangkan keamanan aplikasi *mobile*-nya. Kesimpulannya, MASVS digunakan untuk memberikan standar keamanan yang dapat digunakan untuk aplikasi, sebagai panduan selama fase-fase pengembangan dan pengujian aplikasi, dan sebagai dasar untuk verifikasi seberapa amannya keamanan aplikasi.

2.7.3 OWASP Mobile Penetration Testing Checklist

Table 2.1 Architecture, Design and Threat Modeling Requirements.

ID	MASVS-ID	Detailed Verification Requirement	L1	L2
1.1	MASTG-ARCH-1	<i>All app components are identified and known to be needed.</i>		
1.2	MASTG-ARCH-2	<i>Security controls are never enforced only on the client side, but on the respective remote endpoints.</i>		
1.3	MASTG-ARCH-3	<i>A high-level architecture for the mobile app and all connected remote services has been defined and security has been addressed in that architecture.</i>		
1.4	MASTG-ARCH-4	<i>Data considered sensitive in the context of the mobile app is clearly identified.</i>		
1.5	MASTG-ARCH-5	<i>All app components are defined in terms of the business functions and/or security functions they provide.</i>		
1.6	MASTG-ARCH-6	<i>A threat model for the mobile app and the associated remote services has been produced that identifies potential threats and countermeasures.</i>		
1.7	MASTG-ARCH-7	<i>All security controls have a centralized implementation.</i>		
1.8	MASTG-ARCH-8	<i>There is an explicit policy for how cryptographic keys (if any) are managed, and the lifecycle of cryptographic keys is enforced. Ideally, follow a key management standard such as NIST SP 800-57.</i>		
1.9	MASTG-ARCH-9	<i>A mechanism for enforcing updates of the mobile app exists.</i>		
1.10	MASTG-ARCH-10	<i>Security is addressed within all parts of the software development lifecycle.</i>		
1.11	MASTG-ARCH-11	<i>A responsible disclosure policy is in place and effectively applied.</i>		
1.12	MASTG-ARCH-12	<i>The app should comply with privacy laws and regulations.</i>		

Table 2.2 Data Storage and Privacy Requirements

ID	MASVS-ID	Detailed Verification Requirement	L1	L2
2.1	MASTG-STORAGE-1	<i>System credential storage facilities need to be used to store sensitive data, such as PII, user credentials or cryptographic keys.</i>		
2.2	MASTG-STORAGE-2	<i>No sensitive data should be stored outside of the app container or system credential storage facilities.</i>		
2.3	MASTG-STORAGE-3	<i>No sensitive data is written to application logs.</i>		
2.4	MASTG-STORAGE-4	<i>No sensitive data is shared with third parties unless it is a necessary part of the architecture.</i>		
2.5	MASTG-STORAGE-5	<i>The keyboard cache is disabled on text inputs that process sensitive data.</i>		
2.6	MASTG-STORAGE-6	<i>No sensitive data is exposed via IPC mechanisms.</i>		
2.7	MASTG-STORAGE-7	<i>No sensitive data, such as passwords or pins, is exposed through the user interface.</i>		
2.8	MASTG-STORAGE-8	<i>No sensitive data is included in backups generated by the mobile operating system.</i>		
2.9	MASTG-STORAGE-9	<i>The app removes sensitive data from views when moved to the background.</i>		
2.10	MASTG-STORAGE-10	<i>The app does not hold sensitive data in memory longer than necessary, and memory is cleared explicitly after use.</i>		
2.11	MASTG-STORAGE-11	<i>The app enforces a minimum device-access-security policy, such as requiring the user to set a device passcode.</i>		
2.12	MASTG-STORAGE-12	<i>The app educates the user about the types of personally identifiable information processed, as well as security best practices the user should follow in using the app.</i>		
2.13	MASTG-STORAGE-13	<i>No sensitive data should be stored locally on the mobile device. Instead, data should be retrieved from a remote endpoint when needed and only be kept in memory.</i>		
2.14	MASTG-STORAGE-14	<i>If sensitive data is still required to be stored locally, it should be encrypted using a key derived from hardware backed storage which requires authentication.</i>		
2.15	MASTG-STORAGE-15	<i>The app's local storage should be wiped after an excessive number of failed authentication attempts.</i>		

Table 2.3 Architecture, Design and Threat Modeling Requirements

ID	MASVS-ID	Detailed Verification Requirement	L1	L2
3.1	MASTG-CRYPTO-1	<i>The app does not rely on symmetric cryptography with hardcoded keys as a sole method of encryption.</i>		
3.2	MASTG-CRYPTO-2	<i>The app uses proven implementations of cryptographic primitives.</i>		
3.3	MASTG-CRYPTO-3	<i>The app uses cryptographic primitives that are appropriate for the particular use-case, configured with parameters that adhere to industry best practices.</i>		
3.4	MASTG-CRYPTO-4	<i>The app does not use cryptographic protocols or algorithms that are widely considered deprecated for security purposes.</i>		
3.5	MASTG-CRYPTO-5	<i>The app doesn't re-use the same cryptographic key for multiple purposes.</i>		
3.6	MASTG-CRYPTO-6	<i>All random values are generated using a sufficiently secure random number generator.</i>		

Table 2.4 Authentication and Session Management Requirements

ID	MASVS-ID	Detailed Verification Requirement	L1	L2
4.1	MASTG-AUTH-1	<i>If the app provides users access to a remote service, some form of authentication, such as username/password authentication, is performed at the remote endpoint.</i>		
4.2	MASTG-AUTH-2	<i>If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user's credentials.</i>		
4.3	MASTG-AUTH-3	<i>If stateless token-based authentication is used, the server provides a token that has been signed using a secure algorithm.</i>		
4.4	MASTG-AUTH-4	<i>The remote endpoint terminates the existing session when the user logs out.</i>		
4.5	MASTG-AUTH-5	<i>A password policy exists and is enforced at the remote endpoint.</i>		
4.6	MASTG-AUTH-6	<i>The remote endpoint implements a mechanism to protect against the submission of credentials an excessive number of times.</i>		
4.7	MASTG-AUTH-7	<i>Sessions are invalidated at the remote endpoint after a predefined period of inactivity and access tokens expire.</i>		
4.8	MASTG-AUTH-8	<i>Biometric authentication, if any, is not event-bound (i.e. using an API that simply returns "true" or "false"). Instead, it is based on unlocking the keychain/keystore.</i>		
4.9	MASTG-AUTH-9	<i>A second factor of authentication exists at the remote endpoint and the 2FA requirement is consistently enforced.</i>		
4.10	MASTG-AUTH-10	<i>Sensitive transactions require step-up authentication.</i>		
4.11	MASTG-AUTH-11	<i>The app informs the user of all sensitive activities with their account. Users are able to view a list of devices, view contextual information (IP address, location, etc.), and to block specific devices.</i>		
4.12	MASTG-AUTH-12	<i>Authorization models should be defined and enforced at the remote endpoint.</i>		

Table 2.5 Network Communication Requirements

ID	MASVS-ID	Detailed Verification Requirement	L1	L2
5.1	MASTG-NETWORK-1	<i>Data is encrypted on the network using TLS. The secure channel is used consistently throughout the app.</i>		
5.2	MASTG-NETWORK-2	<i>The TLS settings are in line with current best practices, or as close as possible if the mobile operating system does not support the recommended standards.</i>		
5.3	MASTG-NETWORK-3	<i>The app verifies the X.509 certificate of the remote endpoint when the secure channel is established. Only certificates signed by a trusted CA are accepted.</i>		
5.4	MASTG-NETWORK-4	<i>The app either uses its own certificate store, or pins the endpoint certificate or public key, and subsequently does not establish connections with endpoints that offer a different certificate or key, even if signed by a trusted CA.</i>		
5.5	MASTG-NETWORK-5	<i>The app doesn't rely on a single insecure communication channel (email or SMS) for critical operations, such as enrollments and account recovery.</i>		
5.6	MASTG-NETWORK-6	<i>The app only depends on up-to-date connectivity and security libraries.</i>		

Table 2.6 Platform Interaction Requirements

ID	MASVS-ID	Detailed Verification Requirement	L1	L2
6.1	MASTG-PLATFORM-1	<i>The app only requests the minimum set of permissions necessary.</i>		
6.2	MASTG-PLATFORM-2	<i>All inputs from external sources and the user are validated and if necessary sanitized. This includes data received via the UI, IPC mechanisms such as intents, custom URLs, and network sources.</i>		
6.3	MASTG-PLATFORM-3	<i>The app does not export sensitive functionality via custom URL schemes, unless these mechanisms are properly protected.</i>		
6.4	MASTG-PLATFORM-4	<i>The app does not export sensitive functionality through IPC facilities, unless these mechanisms are properly protected.</i>		
6.5	MASTG-PLATFORM-5	<i>JavaScript is disabled in WebViews unless explicitly required.</i>		
6.6	MASTG-PLATFORM-6	<i>WebViews are configured to allow only the minimum set of protocol handlers required (ideally, only https is supported). Potentially dangerous handlers, such as file, tel and app-id, are disabled.</i>		
6.7	MASTG-PLATFORM-7	<i>If native methods of the app are exposed to a WebView, verify that the WebView only renders JavaScript contained within the app package.</i>		
6.8	MASTG-PLATFORM-8	<i>Object deserialization, if any, is implemented using safe serialization APIs.</i>		
6.9	MASTG-PLATFORM-9	<i>The app protects itself against screen overlay attacks. (Android only)</i>		
6.10	MASTG-PLATFORM-10	<i>A WebView's cache, storage, and loaded resources (JavaScript, etc.) should be cleared before the WebView is destroyed.</i>		
6.11	MASTG-PLATFORM-11	<i>Verify that the app prevents usage of custom third-party keyboards whenever sensitive data is entered (iOS only).</i>		

Table 2.7 Code Quality and Build Setting Requirements

ID	MASVS-ID	Detailed Verification Requirement	L1	L2
7.1	MASTG-CODE-1	<i>The app is signed and provisioned with a valid certificate, of which the private key is properly protected.</i>		
7.2	MASTG-CODE-2	<i>The app has been built in release mode, with settings appropriate for a release build (e.g. non-debuggable).</i>		
7.3	MASTG-CODE-3	<i>Debugging symbols have been removed from native binaries.</i>		
7.4	MASTG-CODE-4	<i>Debugging code and developer assistance code (e.g. test code, backdoors, hidden settings) have been removed. The app does not log verbose errors or debugging messages.</i>		
7.5	MASTG-CODE-5	<i>All third party components used by the mobile app, such as libraries and frameworks, are identified, and checked for known vulnerabilities.</i>		
7.6	MASTG-CODE-6	<i>The app catches and handles possible exceptions.</i>		
7.7	MASTG-CODE-7	<i>Error handling logic in security controls denies access by default.</i>		
7.8	MASTG-CODE-8	<i>In unmanaged code, memory is allocated, freed and used securely.</i>		
7.9	MASTG-CODE-9	<i>Free security features offered by the toolchain, such as byte-code minification, stack protection, PIE support and automatic reference counting, are activated.</i>		

Table 2.8 Resilience Requirements

ID	MASVS-ID	Detailed Verification Requirement	R
8.1	MASTG-RESILIENCE-1	<i>The app detects, and responds to, the presence of a rooted or jailbroken device either by alerting the user or terminating the app.</i>	
8.2	MASTG-RESILIENCE-2	<i>The app prevents debugging and/or detects, and responds to, a debugger being attached. All available debugging protocols must be covered.</i>	
8.3	MASTG-RESILIENCE-3	<i>The app detects, and responds to, tampering with executable files and critical data within its own sandbox.</i>	
8.4	MASTG-RESILIENCE-4	<i>The app detects, and responds to, the presence of widely used reverse engineering tools and frameworks on the device.</i>	
8.5	MASTG-RESILIENCE-5	<i>The app detects, and responds to, being run in an emulator.</i>	
8.6	MASTG-RESILIENCE-6	<i>The app detects, and responds to, tampering the code and data in its own memory space.</i>	
8.7	MASTG-RESILIENCE-7	<i>The app implements multiple mechanisms in each defense category (8.1 to 8.6). Note that resiliency scales with the amount, diversity of the originality of the mechanisms used.</i>	
8.8	MASTG-RESILIENCE-8	<i>The detection mechanisms trigger responses of different types, including delayed and stealthy responses.</i>	
8.9	MASTG-RESILIENCE-9	<i>Obfuscation is applied to programmatic defenses, which in turn impede de-obfuscation via dynamic analysis.</i>	
8.10	MASTG-RESILIENCE-10	<i>The app implements a 'device binding' functionality using a device fingerprint derived from multiple properties unique to the device.</i>	
8.11	MASTG-RESILIENCE-11	<i>All executable files and libraries belonging to the app are either encrypted on the file level and/or important code and data segments inside the executables are encrypted or packed. Trivial static analysis does not reveal important code or data.</i>	
8.12	MASTG-RESILIENCE-12	<i>If the goal of obfuscation is to protect sensitive computations, an obfuscation scheme is used that is both appropriate for the particular task and robust against manual and automated de-obfuscation methods, considering currently published research. The effectiveness of the obfuscation scheme must be verified through manual testing. Note that hardware-based isolation features are preferred over obfuscation whenever possible.</i>	
8.13	MASTG-RESILIENCE-13	<i>As a defense in depth, next to having solid hardening of the communicating parties, application level payload encryption can be applied to further impede eavesdropping.</i>	

2.8 ADB

Android Debug Bridge (ADB) adalah alat yang disediakan oleh Google bersamaan dengan Android untuk memfasilitasi pengelolaan sistem dan *debugging* pada Android. ADB menggunakan USB atau TCP sebagai *transport layer* untuk berkomunikasi dengan perangkat yang menggunakan Android. ADB menggunakan arsitektur yang sederhana seperti *client to server* arsitektur dan terbuat dari 3 komponen utama (Regupathy, 2014), yaitu:

1. Server, berjalan di belakang layer dari *host* sistem dan berkomunikasi dengan klien dan ADB *daemon* yang berjalan pada emulator atau perangkat. Server juga menyediakan detail tentang klien yang terhubung beserta statusnya
2. *Client*, sebagai seorang klien pada sistem *host* yang dapat terhubung ke dalam server. *Client* pada ADB dapat berupa perintah pada adb shell atau adb logcat yang berjalan pada terminal ke Dalvik Debug Monitor Server (DDMS).
3. ADB Daemon, berjalan pada perangkat Android atau emulator sebagai bagian dari USB Android *Framework* dan bertugas untuk berinteraksi dengan server untuk mengelola perangkat Android.

2.9 Memu

Memu adalah salah satu Android Emulator yang paling banyak digunakan orang-orang karena memiliki performa yang tinggi serta pelayanan *multiple-android-kernel*. Memu menyediakan fitur bagi pengguna khususnya *programmer* untuk dapat menjalankan aplikasi pada komputer untuk testing dan hal lainnya tanpa perlu menggunakan perangkat ponsel Android dengan membuat sebuah perangkat virtual. Dengan memu, pengguna dapat memilih jenis dan tipe perangkat yang digunakan seperti tablet, tipe *smartphone*, dan lainnya. (Alabduljabbar, 2021)

2.10 Frida

Frida adalah *toolkit* instrumentasi yang memiliki banyak keunggulan diantara Frida sangat cocok untuk alat pembantu pengujian dan pengevaluasian aplikasi android *native*. Frida memungkinkan penggunanya untuk melakukan *intercept* data yang diterima dan dikirim oleh aplikasi. Selain itu pengguna juga bisa menyuntikkan kode atau *script* ke dalam proses tersebut. (Ravnas, 2012)

2.11 Reflutter

Reflutter adalah *framework* yang membantu penggunanya untuk melakukan *reverse engineering* pada aplikasi menggunakan versi *framework* flutter yang telah dipatch dan siap untuk aplikasi *repacking*. Framework ini memiliki proses *deserialisation snapshot* yang telah dimodifikasi untuk memungkinkan penggunanya melakukan analisis dinamis dengan cara yang lebih nyaman. (Reflutter, 2021)

2.12 Uber-Signer

Uber-Signer adalah *tools* yang membantu dalam menandatangani aplikasi, *zip aligning* dan memverifikasi beberapa paket aplikasi Android (APK) dengan debug atau menyediakan sertifikat kepada aplikasi. Uber-signer mendukung skema penandatanganan kepada Android v1, v2 dan v3. Penandatanganan yang mudah dan nyaman dengan *embedded debug keystore*. *Tools* ini dapat secara otomatis memverifikasi tanda tangan dan *zipalign* setelah penandatanganan. (patrickfav, 2016)

2.13 MobSF

Mobile Security Framework atau MobSF adalah alat all-in-one yang dilakukan secara otomatis untuk melakukan pengujian keamanan (*penetration testing*), analisis *malware* dan *security assessment framework* yang mendukung analisis aplikasi secara statis dan dinamis. Penganalisaan yang dilakukan secara dinamis membantu pengguna untuk melakukan penilaian keamanan secara *runtime* dan *instrument* pengujian yang interaktif. (MobSF, 2015)

2.14 Burp Suite

Burp Suite adalah *tools* yang digunakan untuk *penetration testing* aplikasi khususnya web. Burp suite dikembangkan oleh perusahaan bernama *Portswigger*. *Burp Suite* memiliki beberapa fitur yang sangat berguna seperti melakukan *Intercept* dan melakukan *tampering* pada *request* (HTTP/HTTPS), pengujian manual pada *out-of-band vulnerability*, pengujian web socket, pengujian kekuatan *token*, serta pengujian kerentanan dari *clickjacking* dan *Cross-Site Request Forgery* (CSRF). (Rahalkar, 2021)

2.15 Apk-Tool

Apktool adalah *tool open source* berupa *command-line* dari Bahasa pemrograman java yang digunakan untuk melakukan reverse engineering dari paket aplikasi Android (APK). Apktool dapat digunakan untuk mengekstrak isi dari file APK yang berisi *resource* aplikasi, *bytecode*, serta file manifest dari aplikasi. Apktool biasanya dimanfaatkan untuk melakukan *decode*, menganalisis dan memodifikasi *resource*, *bytecode* untuk mengetahui fungsionalitas dari aplikasi dan masih banyak lagi. (ibotpeaches, 2010)

2.16 Radare2

Radare adalah sebuah *framework open source* yang digunakan untuk *reverse engineering* dan analisa *binary*. Radare2 dapat digunakan untuk *disassembling*, Analisa data, *patching binary*, mencari, mengganti, membandingkan dan memvisualisasikan data, dan banyak lagi. Radare2 terdiri dari berbagai *library* dan program yang diautomasi dengan hampir seluruh bahasa pemrograman. (Alvarez, 2006)

BAB 3

DESKRIPSI UMUM

3.1 Latar Belakang Perusahaan

*3.1.1 **Informasi Umum Perusahaan***



Gambar 3.1 Logo Perusahaan PT Indomobil Finance Indonesia

PT Indomobil Finance Indonesia (“Perseroan”) merupakan perusahaan yang bergerak dalam bidang bisnis jasa pembiayaan baik kendaraan bermotor, alat berat, properti, maupun pembiayaan mikro dengan bentuk pembiayaan konsumen, sewa guna usaha, dan anjak piutang. PT Indomobil Finance Indonesia didirikan pada tahun 1993 yang sebelumnya bernama PT Indomaru Multi Finance dan berganti nama setelah pengambilalihan saham Marubeni Corporation oleh Indomobil Group dan menghasilkan nama dan logo perusahaan terbarunya yang dapat dilihat pada gambar 1.1.

Melalui strategi bisnis yang terus diselaraskan dengan tuntutan perkembangan iklim usaha, Perseroan tetap menjaga komitmen untuk menjadi perusahaan pembiayaan yang handal dan terpercaya di Indonesia. Sesuai dengan Peraturan Otoritas Jasa Keuangan (POJK) Nomor 29/POJK.05/2014 Tahun 2014 tentang Penyelenggaraan Usaha Perusahaan Pembiayaan dimana kegiatan usaha Perseroan adalah:

- Pembiayaan Investasi
- Pembiayaan Modal Kerja
- Pembiayaan Multiguna
- Sewa Operasi
- Pembiayaan Syariah

*3.1.2 **Sejarah Perusahaan***

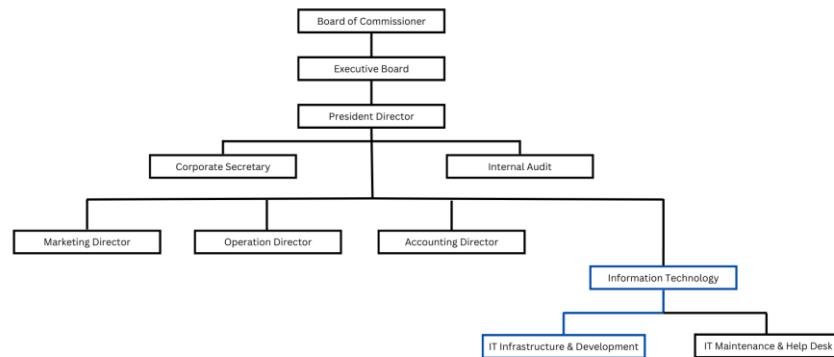
- (1993) PT Indomobil Finance Indonesia didirikan pada November 1993 dengan nama PT Indomaru Multi Finance.
- (2000) Perubahan kepemilikan saham Perseroan dengan komposisi 99,25% dimiliki oleh PT Indomobil Sukses Internasional Tbk dan

0,75% dimiliki oleh PT IMG Sejahtera Langgeng dan mengalami peresmian kantor cabang Perseroan yang pertama

- (2003) Perubahan nama Perseoran dari PT Indomaru Multi Finance menjadi PT Indomobil Finance Indonesia
- (2004) Perseroan melakukan penerbitan Obligasi senilai Rp300 miliar
- (2005) Perseroan melakukan penerbitan Obligasi senilai Rp350 miliar
- (2006) Perseroan memperoleh pinjaman senilai USD60 juta
- (2009) Perseroan membuka sektor pembiayaan kendaraan niaga dan menerbitkan Obligasi sebesar Rp500 miliar
- (2010) Perseroan melakukan diversifikasi produk untuk sektor pembiayaan alat-alat berat
- (2011) Perseroan mendapatkan *rating* “idA, menerbitkan Obligasi sebesar Rp1 triliun, memperoleh pinjaman sebesar USD75 juta, dan memperoleh tambahan modal sebesar Rp500 juta
- (2012) Perseroan menerbitkan Obligasi sebesar Rp1.3 triliun, dan memperoleh pinjaman sebesar USD75 juta
- (2013) Perseroan menerbitkan Obligasi sebesar Rp822 miliar, memperoleh pinjaman sebesar USD126 juta dan mengalihkan saham kepada PT Indomobil Multi Jasa Tbk
- (2014) Perseroan menerbitkan Obligasi sebesar Rp440 miliar dan memperoleh pinjaman sebesar USD172.5 juta
- (2015) Perseroan menerbitkan Obligasi sebesar Rp1.09 triliun, memperoleh pinjaman sebesar USD300 juta, dan memperoleh tambahan modal sebesar Rp50 miliar
- (2016) Perseroan menerbitkan Obligasi sebesar Rp1.5 triliun
- (2017) Perseroan menerbitkan Obligasi sebesar Rp910 miliar dan memperoleh pinjaman sebesar USD250 juta
- (2018) Perseroan menerbitkan Obligasi sebesar Rp2.08 triliun dan memperoleh pinjaman sebesar USD275 juta
- (2019) Perseroan memperoleh tambahan modal sebesar Rp242 miliar dan memperoleh pinjaman sebesar USD290 juta

- (2020) Perseroan menerbitkan Obligasi sebesar Rp336 miliar, memperoleh tambahan modal sebesar Rp150 miliar, dan memperoleh pinjaman sebesar USD255 juta

3.1.3 Struktur Organisasi Perusahaan



Gambar 3.2 Struktur Organisasi PT Indomobil Finance Indonesia.

Proses pengujian dan penggeraan skripsi dengan proyek yang diberikan oleh PT Indomobil Finance Indonesia dilakukan bersamaan dengan saat program magang berlangsung. Selama proses penggeraan, penulis berada di bawah bimbingan dan bagian dari divisi *Information Technology*. Divisi ini terbagi menjadi dua departemen berbeda, yaitu *IT Infrastructure & Development* dan *IT Maintenance & Help Desk*. Dalam departemen tersebut, penulis merupakan bagian dari departemen *IT Infrastructure & Development*.

3.2 Peran Penulis dalam Perusahaan

Selama program magang dan penggeraan skripsi, peran penulis dalam organisasi adalah sebagai *full stack developer*. Tugas dan tanggung jawab penulis selama berada di PT Indomobil Finance Indonesia (IMFI) yaitu melakukan pengembangan *software* yang dimiliki oleh perusahaan, melakukan *maintenance* dan *bug fixing* terhadap aplikasi, membuat API, dan lain-lain. Selain itu, penulis juga diberikan kesempatan untuk melakukan pengujian keamanan pada aplikasi yang dimiliki perusahaan sebagai bentuk proyek penggeraan skripsi. Proyek ini dilakukan dengan tujuan untuk menguji seberapa aman aplikasi dan mencari kerentanan atau bug yang ada pada aplikasi tersebut. Dalam penggeraan proyek ini, penulis berkolaborasi dan dibantu oleh para mentor serta rekan sesama staf lainnya dalam bentuk diskusi dan masukan untuk pengujian dan penulisan skripsi.

3.3 Kondisi Saat Ini

PT. Indomobil Finance sebagai perusahaan yang memiliki banyak pekerja dan cabang perusahaan di Indonesia tentunya memiliki sarana dalam membantu keberlangsungan pekerja dalam berkegiatan. Karena pada dasarnya, perusahaan yang tidak memiliki sarana dalam mengatur proses kegiatan para pekerjanya dengan baik dapat menyebabkan masalah-masalah yang dapat mengurangi performa kerja dari karyawannya. Salah satu kendala yang sering dialami oleh pekerja perusahaan adalah munculnya kesulitan dalam menggunakan layanan perusahaan yang dilakukan secara manual. Kegiatan seperti *booking ruang meeting*, absensi, melakukan percakapan antar divisi atau cabang, pengajuan biasanya dilakukan dengan melakukan *request service* ke staf yang biasanya membutuhkan waktu yang lama dan tidak optimal.

Service aplikasi yang dibuat secara internal atau yang biasa disebut e-Service Internal adalah aplikasi yang memiliki fasilitas yang hanya mencakup perusahaan tersebut. Aplikasi e-Service Internal hanya dapat digunakan oleh pekerja atau karyawan perusahaan tersebut dengan menggunakan akun yang diberikan oleh perusahaan. Dengan adanya e-Service pada perusahaan, karyawan dapat melakukan pengajuan kegiatan atau keperluan tanpa harus mencari staf yang bertanggung jawab terkait hal itu. PT. Indomobil Finance sendiri telah menyediakan aplikasi *mobile* bernama IMFI One yang digunakan untuk melakukan pengelolaan dan mengatur segala kegiatan karyawan di setiap cabang yang berlangsung dalam perusahaan.

Pada aplikasi IMFI One, karyawan dapat melakukan absensi masuk dan keluar, *chatting* secara *online* antar karyawan atau staf, informasi daftar cabang dan *extension*, pengajuan izin, cuti dan lembur, pengajuan serta pengecekan agenda, *booking ruangan meeting*, dan daftar *event* perusahaan. Selain itu, aplikasi IMFI One juga menyediakan fitur bagi staf yang berwenang untuk menerima dan mengecek pengajuan-pengajuan karyawan.

3.4 Identifikasi Masalah

Terdapat beberapa permasalahan utama yang dapat terlihat dari kondisi penerapan aplikasi saat ini, diantaranya:

1. Kurangnya kesadaran dan edukasi mengenai keamanan aplikasi.
Berdasarkan pengamatan penulis dan pengalaman pengembang aplikasi-aplikasi pada PT Indomobil Finance Indonesia, terlihat bahwa terjadi kurangnya kesadaran baik dari pihak perusahaan yang tidak menerapkan *requirement* untuk keamanannya maupun dari sisi *engineer* yang belum atau masih sedikit menerapkan panduan-panduan keamanan pada pengembangan program dan aplikasinya.
2. Penambahan dan perubahan *requirement* serta fitur dari aplikasi.
Proses *update* dan perubahan pada aplikasi seringkali terjadi tanpa mempertimbangkan sisi keamanan pada fitur yang ditambahkan, hal ini bisa terjadi karena kebutuhan pengguna atau perusahaan yang mementingkan kinerja fungsi dan kecepatan hasilnya.
3. Tidak adanya pekerja yang bertugas sebagai penjaga, pengecek serta testing terhadap keamanannya.

3.5 Solusi yang Diusulkan

Berdasarkan hasil dari identifikasi masalah yang terjadi saat ini, penulis memiliki beberapa saran yang penulis ajukan sebagai solusi untuk penyelesaian masalah tersebut, diantaranya:

1. Melakukan penerapan *code review* dalam proses pengembangan aplikasi.
Dengan melakukan penerapan *code review* dalam bagian pengembangan aplikasi, para pengembang dapat melakukan *recheck* terhadap sisi penulisan *code* untuk mencegah adanya *bug*, *error*, dan kesalahan lainnya.
2. Menerapkan kebijakan dan perubahan regulasi pada sistem.
Diperlukan adanya kebijakan dan regulasi yang digunakan sebagai acuan bagi para *engineer* dalam mengembangkan dan menjaga penggunaan aplikasi.
3. Melakukan pengujian baik secara fungsionalitas maupun pengujian keamanan. Dalam pengujian keamanan, perusahaan dapat melakukan pengujian dengan melakukan uji *penetration testing* pada aplikasi secara rutin.

3.6 Ruang Lingkup Pengujian Keamanan

Ruang lingkup dari pelaksanaan pengujian keamanan terhadap aplikasi IMFI One yaitu:

1. Lingkup dilakukannya pengujian keamanan adalah seluruh fitur yang terdapat di aplikasi IMFI One.
2. Pengujian keamanan juga dilakukan dengan melakukan analisis pada keseluruhan *source code*, dokumentasi arsitektur dari aplikasi IMFI One.
3. Aplikasi yang digunakan sebagai emulator android untuk menjalankan aplikasi yang akan dilakukan uji keamanan dalam bentuk *file apk* adalah Memu.
4. Pengujian dilakukan menggunakan dua perspektif yaitu penguji dari perspektif sebagai bagian dari dalam perusahaan dan sebagai orang luar perusahaan (publik).
5. Aplikasi yang akan digunakan sebagai *virtual machine* dalam menggunakan sistem operasi virtual adalah VMWare yang akan dijadikan lingkungan dilakukannya pengujian.
6. Data yang diperoleh penguji dalam melakukan pengujian adalah beberapa *user testing* aplikasi dan *file apk* dari aplikasi *development*.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Deskripsi Target Pengujian

IMFI One merupakan aplikasi *internal* yang ditujukan kepada karyawan PT Indomobil Finance Indonesia (IMFI) untuk memudahkan segala keperluan pekerjaan karyawan. Saat ini, aplikasi ini dapat diunduh secara bebas untuk *smartphone* berbasis Android melalui Google Play dan *smartphone* berbasis iOS melalui App Store. Aplikasi IMFI One merupakan aplikasi *superapps* berbasis *mobile* karena memiliki banyak kegunaan seperti:

- 1) Fitur chat antar karyawan.
- 2) Pengajuan izin, cuti, *medical*, lembur, *reimburse*, operasional dan lainnya.
- 3) Pengurusan dan persetujuan *approval*-*approval* karyawan oleh yang berwenang seperti kepala divisi.
- 4) Pemesanan ruangan, agenda *event*, *meeting*.
- 5) *Report* keuangan perusahaan secara umum.
- 6) Informasi *extension* nomor telepon, daftar cabang, *virtual card* karyawan.
- 7) Penyebaran berita, pengumuman, serta informasi dari *executive board*.
- 8) Fitur *workspace* seperti detil *stock opname*, pengembalian, peminjaman dan penyerahan bpkb.
- 9) Fitur insentif yang berupa tampilan *website* menggunakan *webview*.

Aplikasi ini menggunakan mekanisme pendaftaran *user* secara manual melalui *database*, yang berarti pengunduh aplikasi secara umum selain karyawan tidak memiliki akses masuk ke dalam aplikasi. Saat ini, aplikasi IMFI One memiliki banyak fitur yang berkaitan dengan aset dan fitur aplikasi lainnya seperti adanya penggunaan *webview* untuk gambar laporan keuangan, serta pengurusan insentif yang tersambung dengan *website* lainnya.

4.2 Status Keamanan Aplikasi

Sebagai aplikasi yang memiliki banyak kegunaan, IMFI One saat ini belum melalui proses pengujian keamanan, baik secara mandiri maupun melalui standarisasi serta ketentuan khusus dari keamanan aplikasi *mobile* seperti OWASP *Mobile Application Security Verification Standard* dan standar keamanan lainnya. Selain itu, perusahaan Indomobil Finance Indonesia (IMFI) saat ini tidak memiliki pekerja dengan posisi seperti *security engineer* dan *penetration tester* yang berarti hampir seluruh aplikasi baik berbasis *mobile* maupun *website*, belum melalui proses pengujian keamanan sama sekali. Hal ini yang menyebabkan penguji memiliki dorongan untuk menjadikan projek

pengujian aplikasi sebagai dasar, panduan serta referensi bagi *developer* untuk penerapan aplikasi-aplikasi yang dibuat dan dikembangkan kedepannya.

4.3 Rencana Implementasi Solusi

Implementasi pengujian sebagai solusi dibagi berdasarkan kategori dari ketentuan dan panduan keamanan standar yaitu OWASP *Mobile Application Security Verification Standard* (MASVS). Pembagian kategori pengujian dibagi menjadi delapan kategori yang masing-masing memiliki komponen dan tujuan terkait dengan kategorinya. Setiap poin pada masing-masing kategori kemudian dibagi lebih lanjut menjadi beberapa isi atau bentuk paragraf yaitu:

- 1) Paragraf pertama berisi latar belakang atau penjelasan dari istilah atau detil dari poin yang akan diuji, paragraf pertama ini bersifat opsional kepada poin yang berkemungkinan memiliki istilah yang jarang dikenal, maksud yang terkesan ambigu atau dapat juga berisi latar belakang dari tujuan pengujian.
- 2) Paragraf kedua berisi bagaimana pengujian dilakukan oleh penguji dan pemenuhan syarat untuk pemenuhan poin tersebut.
- 3) Status atau hasil pengujian yang memiliki tiga kesimpulan yaitu *PASS*, *FAIL*, dan *N/A*. *PASS* berarti poin pengujian tersebut telah terpenuhi oleh aplikasi. *FAIL* berarti poin pengujian tersebut tidak terpenuhi oleh aplikasi. *N/A* berarti pengujian tidak dapat dilakukan yang berarti tidak *PASS* maupun *FAIL* karena poin pengujian tidak dimiliki atau tidak sesuai dengan penerapan aplikasi.
- 4) *Score* pada kerentanan atau resiko yang dapat terjadi pada kasus status atau hasil pengujian yang fail. Penentuan *score* kerentanan atau resiko didasarkan pada penggunaan *framework* terbuka yaitu *Common Vulnerability Scoring System* (CVSS).
- 5) Bukti dan hasil dari pengujian atau analisis yang dilakukan. Pada bagian ini juga dapat berisi langkah-langkah dari pengujian sebagai salah satu proses pembuktian dari hasil pengujian.
- 6) *Recommendation*, yaitu pemberian saran dan solusi kepada *developer* atau syarat dari aplikasi untuk memperbaiki kerentanan yang ditemukan.

4.4 Hasil Pengujian

4.4.1 MASTG-ARCH: Architecture, Design and Threat Modeling Requirements

4.4.1.1 ARCH-1: “All app components are identified and known to be needed.”

Pengujian untuk poin ini dilakukan dengan cara menganalisis *source code* secara langsung (statis) dan melalui automation *tools* mobSF sebagai basis referensi. Pada poin ini penguji menganalisis beberapa komponen yang dapat dianalisis pada *source code* yang berbasis Bahasa dart dengan framework flutter. Komponen yang dapat dianalisis pada *source code* seperti *function*, *package* yang terpasang, *manifest android* yang berisi *intent*, *activity*, dan *permission* serta komponen-komponen lainnya.

STATUS: “PASS”

```
<manifest android:versionCode="3" android:versionName="1.15.0" android:compileSdkVersion="32" android:compileSdkVersionCodename="12" packages="com.inf1.superapps" platformBuildVersionCode="32" platformBuildVersionName="12"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="32" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.hardware.location.gps" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.SET_ALARM" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES" />
    <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
    <uses-permission android:name="android.permission.USE_FULL_SCREEN_LAYOUT" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
```

Gambar 4.1 Isi File AndroidManifest.xml Aplikasi dari Tools MobSF.

 ACTIVITIES
com.imfi.superapps.MainActivity io.flutter.plugins.urlauncher.WebViewActivity com.wongpiwat.trust_location.TrustLocationPlugin com.google.android.gms.common.api.GoogleApiActivity com.journeyapps.barcodescanner.CaptureActivity

 SERVICES
com.google.firebaseio.components.ComponentDiscoveryService com.baseflow.geolocator.GeolocatorLocationService id.flutter.flutter_background_service.BackgroundService androidx.work.impl.background.systemalarm.SystemAlarmService androidx.work.impl.background.systemjob.SystemJobService androidx.work.impl.foreground.SystemForegroundService androidx.room.MultiInstanceInvalidationService

 RECEIVERS
com.dexterous.flutterlocalnotifications.ScheduledNotificationBootReceiver com.dexterous.flutterlocalnotifications.ScheduledNotificationReceiver id.flutter.flutter_background_service.WatchdogReceiver id.flutter.flutter_background_service.BootReceiver androidx.work.impl.utils.ForceStopRunnable\$BroadcastReceiver androidx.work.impl.background.systemalarm.ConstraintProxy\$BatteryChargingProxy androidx.work.impl.background.systemalarm.ConstraintProxy\$BatteryNotLowProxy androidx.work.impl.background.systemalarm.ConstraintProxy\$StorageNotLowProxy

Gambar 4.2 Daftar Aktivitas dan Layanan Aplikasi dari Tools MobSF.

Sebagai contoh terlihat *permission*, *activity* dan *service* yang digunakan pada aplikasi, hasil *screenshot* di atas didapatkan dari hasil analisis menggunakan *tools* mobSF. Berdasarkan hasil analisis, semua komponen yang berada pada aplikasi memiliki tujuan yang jelas pada setiap kegunaannya, adapun pada *source code* terdapat beberapa komponen yang bersifat tidak wajib namun dapat membantu dan memudahkan developer dalam *flow* kerja aplikasi.

4.4.1.2 ARCH-2: “*Security controls are never enforced only on the client side, but on the respective remote endpoints.*”

Dalam pengembangan aplikasi, sebagian besar proses data dilakukan dengan *remote endpoint* berupa API. Dalam hal ini, aplikasi *mobile* pengguna adalah sebuah *client* sedangkan server adalah *remote endpoint*.

Pada poin pengujian ini, dilakukan pengetesan terhadap *url* pada sisi *remote endpoint* yaitu API. Pengujian dapat dilakukan dengan melakukan pengecekan kebijakan

keamanan pada *remote endpoint* dan dilakukan beberapa serangan seperti *SQL injection*, *brute force*, dan lainnya.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 10 (High)

“AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:H/A:H”

The screenshot shows a POST request to the URL `(URL)/karyawan/IndexKaryawan.ashx`. The 'Body' tab is selected, showing form-data parameters: `NAMA_KARYAWAN` with value `a/**/CREATE/**/TABLE/**/InjectionHauzan4(test/...` and `PAGE` with value `1`. The 'Test Results' tab shows a 200 OK response with a JSON payload containing an error code of 0, a success message, and a data value array. The data value array contains several objects, one of which has a key `NAMA` with a value starting with `N/A`.

Gambar 4.3 Pengujian SQL Injection Pada Field Nama Karyawan.



Gambar 4.4 Bukti hasil SQL Injection pada Database.

Berdasarkan hasil pengujian ditemukan bahwa tidak adanya penggunaan *session token* atau *authorization token* pada pengambilan data karyawan yang dapat memudahkan dilakukannya serangan. Dari hal tersebut ditemukan bahwa

salah satu URL (yang didapatkan dengan *intercept traffic* menggunakan burpsuite) pada *remote endpoint* memiliki kerentanan untuk dilakukan *SQL Injection*. *SQL Injection* dilakukan dengan payload:
`a'/**/CREATE/**/TABLE/**/InjectionHauzan1(test/**/int)**/Select/**/2/**/as/**/NamaKaryawan/**/-.` Untuk lebih meyakinkan diri, penguji meminta ijin kepada mentor untuk dapat melihat *database* bersangkutan apakah *table* berhasil terbuat yang dapat terlihat pada gambar di bawah. Dengan teori yang sama hal ini berarti pelaku serangan dapat menjalankan *query-query* yang sangat fatal seperti *drop table* dan lainnya.

Recommendation: Melakukan sanitasi input atau pencegahan lainnya seperti prepare statement tidak hanya pada client-side melainkan juga pada remote endpoint atau API nya.

4.4.1.3 ARCH-3: “A *high-level architecture* for the mobile app and all connected remote services has been defined and security has been addressed in that architecture .”

High-level architecture adalah sebuah model standar yang digunakan untuk menggambarkan struktur dari bagaimana sebuah sistem berjalan. Dalam pengembangan aplikasi banyak cara untuk mempresentasikan arsitektur pada sistem yang dibuat seperti gambaran terkait presentasi (UI, *Presentation logic*), bisnis (*workflow, component, entity*), data (data access, data *utilities, service tools*) beserta *remote service* yang terhubung pada aplikasi tersebut.

STATUS: “FAIL”

Berdasarkan hasil analisis, tidak ditemukan adanya pendefinisan atau penerapan *high-level architecture* pada aplikasi dan *remote service* yang digunakan. Hal ini berarti segi keamanan pada arsitektur tersebut juga tidak bisa diterapkan.

Recommendation: Membuat atau mendefinisikan bentuk dari *high-level architecture* yang berisi gambaran dari presentasi, bisnis, data dan *remote service* yang terhubung pada aplikasi.

4.4.1.4 ARCH-4: “*Data considered sensitive in the context of the mobile app is clearly identified.*”

Data sensitif pada dasarnya adalah data yang tidak bisa secara bebas dilihat oleh orang lain atau orang umum. Data ini tidak harus bersifat merugikan secara material jika diketahui orang seperti kartu kredit, nomor rekening namun dapat juga yang bersifat pribadi seperti data yang berkaitan dengan keyakinan, Kesehatan, kondisi fisik, data keuangan pribadi dan data lainnya.

STATUS: “PASS”

Berdasarkan hasil analisis yang dilakukan pada aplikasi, data sensitif dapat diidentifikasi dengan mudah karena beberapa data sensitif sudah ditutupi dengan * seperti *password*, nominal dan alasan *reimburse* selain itu beberapa data penting seperti data pribadi karyawan juga tidak ditampilkan secara langsung di aplikasi dan memiliki *privilege* khusus untuk menggunakan fiturnya seperti *scan virtual card*.

4.4.1.5 ARCH-5: “*All app components are defined in terms of the business functions and/or security functions they provide.*”

Pengujian pada poin ini dilakukan dengan menganalisis dokumentasi aplikasi terkait semua komponen aplikasi yang terdefinisi dengan baik. Analisis dilakukan tidak hanya terkait definisi komponen secara umum melainkan juga dari sisi bisnis dan sisi keamanannya.

STATUS: “FAIL”

Berdasarkan hasil analisis, diketahui bahwa tidak ada pendefinisian khusus terhadap komponen-komponen yang ada pada aplikasi dari sisi keamanannya. Namun sudah memenuhi kriteria dalam pendefinisian komponennya dari sisi bisnis.

Recommendation: Mendefinisikan komponen-komponen aplikasi, termasuk juga dari segi fungsi-fungsi keamanan aplikasi.

4.4.1.6 ARCH-6: “A threat model for the mobile app and the associated remote services has been produced that identifies potential threats and countermeasures.”

Pada dasarnya, *threat modeling* adalah elemen penting dalam *Security Development Life Cycle* (SDL). Dengan adanya *threat model* pengembang aplikasi dapat mengidentifikasi ancaman, serangan, kerentanan dan pencegahan serta penanggulangan yang dapat dilakukan terhadap aplikasi. Pada dasarnya untuk melakukan *threat modelling* diperlukan beberapa step seperti:

1. Mendefinisikan *requirement-requirement* pada keamanan aplikasi
2. Membuat diagram untuk aplikasi
3. Mengidentifikasi ancaman
4. Menanggulangi ancaman tersebut
5. Melakukan validasi bahwa ancaman sudah ditanggulangi.

STATUS: “FAIL”

Berdasarkan hasil analisis dan pengecekan pada dokumentasi aplikasi, dapat disimpulkan bahwa kelima step pada *threat modeling* tidak terpenuhi. Hal ini didasarkan pada tidak adanya dokumentasi terkait keamanan aplikasi saat masa perancangan, tidak adanya pendefinisian terhadap *requirement* serta ancaman yang dapat terjadi pada aplikasi. Dengan ini dapat disimpulkan juga bahwa step lainnya tidak dapat dipenuhi.

Recommendation: Membuat threat model sesuai dengan ketentuan yang ada pada saat pengembangan dan pemeliharaan aplikasi untuk kedepannya.

4.4.1.7 ARCH-7: “All security controls have a centralized implementation.”

Pada dasarnya *centralized implementation* berarti implementasi terkait keamanan harus dipusatkan atau dimasukkan sebagai *plugin* atau *library* terpisah. Hal ini untuk memastikan fungsionalitas terkait keamanan dapat dengan mudah dilakukan *maintenance* dan diperbarui. Contohnya adalah ketika aplikasi menggunakan suatu *method cryptography*, daripada mendistribusikan *method* tersebut ke seluruh fitur aplikasi secara masing-masing akan lebih baik jika *method* tersebut dibuat sekali secara terpusat sehingga *method* tersebut hanya perlu dipanggil pada fitur yang memerlukan.

STATUS: “PASS”

Berdasarkan gambar 4.26, aplikasi selalu memanggil *method* yang sama yaitu pada file *SignIn.ashx* setiap kali diperlukan otentikasi atas *login* ulang setiap mengakses beberapa halaman di aplikasi. Hal ini berarti penerapan *centralized implementation* telah dilakukan karena menggunakan *method* yang sama untuk semua implementasi keamanan otentikasinya.

4.4.1.8 ARCH-8: “There is an explicit policy for how cryptographic keys (if any) are managed, and the lifecycle of cryptographic keys is enforced. Ideally, follow a key management standard such as NIST SP 800-57.”

Pengujian pada poin ini dilakukan dengan menganalisis penggunaan *cryptography policy* yang berlaku pada aplikasi dan pada *endpoint*.

STATUS: “N/A”

Pada poin ini pengujian tidak bisa dilakukan karena aplikasi tidak menggunakan enkripsi atau *policy cryptography* sama sekali (subbab 4.3.3).

4.4.1.9 ARCH-9: “A mechanism for enforcing updates of the mobile app exists.”

Pada aplikasi, mekanisme untuk melakukan *update* sangatlah wajib untuk dilakukan. Selain untuk menambahkan fitur, pada perbaruan aplikasi juga biasanya dilakukan *bug fixing* dan perbaikan kerentanan yang ada. Maka dari itu diperlukan adanya mekanisme untuk melakukan *update* secara paksa agar tidak adanya perbedaan antar pengguna demi mencegah terjadinya eksloitasi yang terjadi pada versi yang lama.

Pada poin ini pengujian dilakukan dengan melakukan pengecekan aplikasi secara langsung dengan memasang versi aplikasi yang rendah dan melihat mekanisme apa yang terjadi. Selain itu untuk pengecekan lebih lanjut akan dilakukan pengecekan pada penggunaan *function* atau *package* yang digunakan dalam mekanisme ini pada *source code* baik langsung yaitu pada flutter maupun pada *source* yang didapat setelah *reverse engineering*.

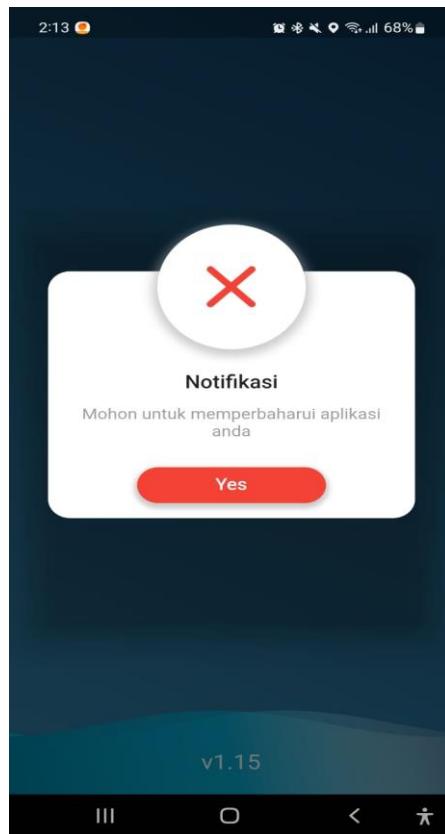
STATUS: “PASS”

```
import android.content.pm.PackageManager;
```

Gambar 4.5 Import Library Untuk Mengecek Version Code Aplikasi.

```
sb.append(".");
if (context != null) {
    try {
        sb.append(c.a(context).c(context.getPackageName(), 0).versionCode);
    } catch (PackageManager.NameNotFoundException unused) {
    }
}
```

Gambar 4.6 Pengecekan Version Code pada Source Code.



Gambar 4.7 Tampilan Pop Up Force Update pada Aplikasi.

Berdasarkan hasil analisis terbukti setelah melakukan pemasangan aplikasi dengan versi rendah, notifikasi untuk melakukan *update* akan muncul yang mengarahkan pengguna ke play store. Selain itu, untuk lebih memastikan maka akan dilakukan analisis pada *source code* dari analisis *tools* mobSF dan terlihat terdapat *package* dan *function* serupa yang digunakan untuk mendapatkan *version code* dari aplikasi.

4.4.1.10 ARCH-10: “*Security is addressed within all parts of the software development lifecycle.*”

Keamanan aplikasi selain dilakukan *addressing* saat aplikasi sudah rilis, tapi diperlukan juga adanya *addressing security* ketika aplikasi sedang dalam tahap pengembangan. Hal ini dilakukan untuk mengurangi risiko dan tindakan pencegahan terkait adanya kelemahan sebelum dilakukan proses rilis.

Pada poin ini, pengujian dilakukan dengan mengecek adanya dokumentasi terkait keperluan keamanan pada aplikasi, apakah aplikasi telah memiliki identifikasi terhadap risiko, ancaman, *requirement* keamanan, *threat modelling*,

STATUS: “FAIL”

Berdasarkan hasil analisis termasuk dengan kesimpulan pada (subbab 4.3.1.6) tidak ditemukan adanya tindakan pengecekan keamanan sama sekali terhadap aplikasi ketika dalam masa pengembangan.

Recommendation: Melakukan addressing keamanan aplikasi pada semua proses software development lifecycle dengan melakukan identifikasi terhadap risiko, ancaman, *requirement* keamanan, *threat modelling*

4.4.1.11 ARCH-11: “A responsible disclosure policy is in place and effectively applied.”

Responsible disclosure policy pada dasarnya adalah sebuah model kebijakan dimana perusahaan memberikan ijin bagi para pelaku keamanan untuk mengemukakan kelemahan yang ditemukan kepada perusahaan secara aman. Lalu perusahaan secara opsional dapat mengemukakan kelemahan atau *bug* yang ditemukan pada *update* sebelumnya kepada umum setelah kelemahan tersebut telah diperbaiki.

Pengujian pada poin ini dilakukan dengan mengecek dokumentasi yang ada terkait aplikasi, berdiskusi dengan developer dan mentor terkait kebijakan yang ada terkait *responsible disclosure*.

STATUS: “FAIL”

Berdasarkan hasil analisis terhadap aplikasi terkait kebijakan *responsible disclosure* yang berlaku ditemukan bahwa aplikasi tidak memiliki kebijakan terkait *responsible disclosure*.

Recommendation: Menerapkan kebijakan terkait *Responsible disclosure* baik secara tertutup maupun terbuka.

4.4.1.12 ARCH-12: “The app should comply with privacy laws and regulations.”

Pengujian pada poin ini dilakukan dengan melakukan diskusi dengan developer serta mentor terkait pemenuhan hukum aplikasi dalam menjagaan privasi dan hak pengguna aplikasi.

STATUS: “PASS”

Berdasarkan hasil analisis dan diskusi ditemukan bahwa aplikasi sudah melalui pemeriksaan terhadap kepatuhan hukumnya. Selain itu, aplikasi yang dimasukkan ke dalam play store sudah melalui proses pengecekan tersendiri dari pihak play store terkait pemenuhan hukum privasi dan regulasi.

4.4.2 MASTG-STORAGE: Data Storage and Privacy Requirements

4.4.2.1 STORAGE-1: “System credential storage facilities need to be used to store sensitive data, such as PII, user credentials or cryptographic keys.”

Pada aplikasi, penggunaan *storage* diperlukan sebagai tempat penyimpanan sementara aplikasi untuk mempermudah kegiatan dari developer dalam sistem kerja aplikasinya. Penggunaan *storage* ini harus diiringi dengan pengamanan data sensitif agar data tidak dapat dilihat oleh orang yang tidak memiliki wewenang.

Pengujian pada poin ini dilakukan dengan cara menganalisis data pada *local smartphone* menggunakan adb shell. Penguji akan masuk ke dalam folder /data/user/0/com.imfi.superapps/ untuk mengecek *permission* dan keamanan data yang tersimpan di Shared Preferences.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 2.3 (Low)

“AV:L/AC:H/PR:H/UI:R/S:C/C:L/I:N/A:N ”

```
SM-G935FD:/data/data/com.imfi.superapps # ls -la
total 48
drwx----- 12 u0_a69 u0_a69 4096 2022-11-23 07:52 .
drwxrwx--x 79 system system 4096 2022-11-23 00:52 ..
drwxrwx--x 2 u0_a69 u0_a69 4096 2022-11-22 09:50 app_flutter
drwxrwx--x 2 u0_a69 u0_a69 4096 2022-11-22 11:05 app_sslcache
drwxrwx--x 2 u0_a69 u0_a69 4096 2022-11-22 11:03 app_textures
drwx----- 4 u0_a69 u0_a69 4096 2022-11-22 11:03 app_webview
drwxrwx--x 5 u0_a69 u0_a69 4096 2022-11-22 11:05 cache
drwxrwx--x 3 u0_a69 u0_a69 4096 2022-11-23 14:07 code_cache
drwxrwx--x 2 u0_a69 u0_a69 4096 2022-11-22 09:50 databases
drwxrwx--x 2 u0_a69 u0_a69 4096 2022-11-22 11:05 files
lrwxrwxrwx 1 root root 38 2022-11-23 07:52 lib -> /data/app/com.imfi.superapps-1/lib/arm
drwxrwx--x 2 u0_a69 u0_a69 4096 2022-11-22 09:50 no_backup
drwxrwx--x 2 u0_a69 u0_a69 4096 2022-11-23 00:52 shared_prefs
```

Gambar 4.8 Permission pada Folder Package Aplikasi.

Berdasarkan gambar di atas *permission* pada folder dalam *package* aplikasi sudah sesuai dimana *permission* untuk *read* dan *write* hanya diberikan ke *user* dan *group*.

```
SM-G935FD:/data/data/com.imfi.superapps/shared_prefs # ls
FlutterSharedPreferences.xml flutter_workmanager_plugin.xml
WebViewChromiumPrefs.xml id.flutter.background_service.xml
SM-G935FD:/data/data/com.imfi.superapps/shared_prefs # cat FlutterS>
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
    <string name="flutter.noDataKaryawan">33</string>
    <string name="flutter.login">{"USER_ID": "20070006", "USER_PASSWORD": "123"}</string>
</map>
SM-G935FD:/data/data/com.imfi.superapps/shared_prefs #
```

Gambar 4.9 Status Keamanan Enkripsi Data Sensitif pada File SharedPreferences.

Sedangkan untuk keamanan pada data yang tersimpan di dalam folder SharedPreferences tidak sesuai ketentuan dimana data sensitif yaitu *password* masih berupa *plain text* dan tidak dilakukan enkripsi.

Recommendation: Menggunakan enkripsi pada data yang tersimpan di shared preference dan menyimpan kunci enkripsi ke dalam *keystore*.

4.4.2.2 STORAGE-2: “*No sensitive data should be stored outside of the app container or system credential storage facilities.*”

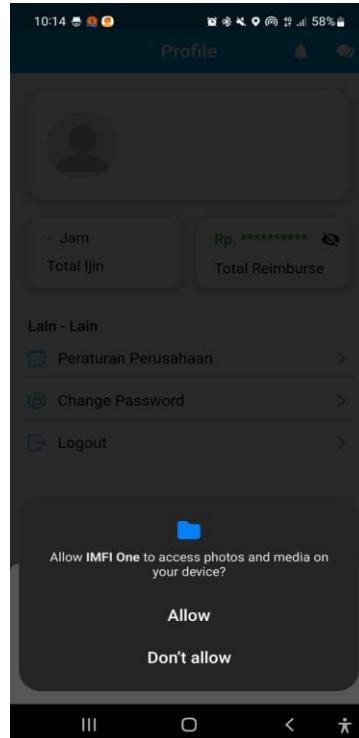
Pada dasarnya, definisi data sensitif adalah informasi yang perlu dilindungi dan dirahasiakan hanya kepada individu yang terkait. Data sensitif mencakup semua data baik asli maupun salinan seperti: opini politik, informasi keuangan (nomor rekening bank, kartu kredit), data *biometric*, etnis atau ras, agama dan informasi rahasia lainnya.

Pengujian pada poin ini dilakukan dengan mengecek keberadaan penggunaan *permission* untuk *read* atau *write external storage*. Selain itu juga dilakukan penganalisisan terkait dimana penggunaan *permission* tersebut serta data apa yang terkait dengan *permission* tersebut.

STATUS: “PASS”

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
```

Gambar 4.10 Storage Permission pada File AndroidManifest.xml

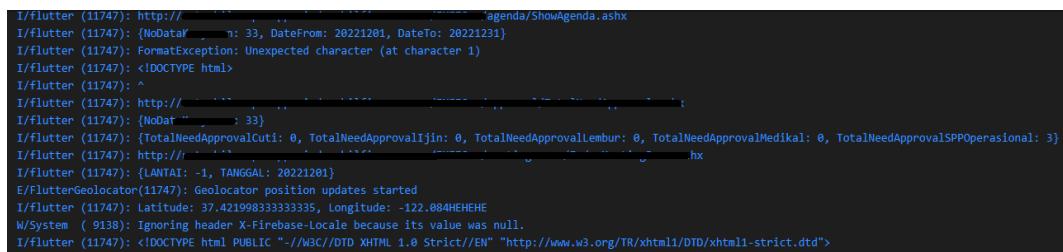


Gambar 4.11 Penggunaan Permission External Storage pada Aplikasi

Berdasarkan hasil analisis pada file *AndroidManifest.xml* terlihat bahwa *permission* untuk *read* dan *write* pada *external storage* digunakan. Namun setelah pengecekan pada aplikasi, terlihat bahwa penggunaan *permission read* dan *write external storage* hanya berada pada fitur dan data gambar profil. Data gambar profil secara umum bukanlah data sensitif, maka dari itu hasil pengujian menyatakan poin ini “*pass*”.

4.4.2.3 STORAGE-3: “No sensitive data is written to application logs.”

Pengujian pada poin ini dilakukan menjalankan *code flutter* apa saja menggunakan emulator android yang telah terpasang aplikasi target yaitu ImfiOne. Setelah emulator dan proses *debugging* berjalan, penguji berpindah kepada aplikasi ImfiOne dan menjalankan setiap fitur pada emulator.



```
I/flutter (11747): http://... ...
I/flutter (11747): agenda/ShowAgenda.ashx
I/flutter (11747): {NoData": ... : 33, DateFrom: 20221201, DateTo: 20221231}
I/flutter (11747): FormatException: Unexpected character (at character 1)
I/flutter (11747): <!DOCTYPE html>
I/flutter (11747): ^
I/flutter (11747): http://...
I/flutter (11747): {NoData": ... : 33}
I/flutter (11747): {TotalNeedApprovalCuti: 0, TotalNeedApprovalIjin: 0, TotalNeedApprovalLembur: 0, TotalNeedApprovalMedikal: 0, TotalNeedApprovalSPPOperasional: 3}
I/flutter (11747): http://...
I/flutter (11747): {LANTAI: -1, TANGGAL: 20221201}
E/flutterGeolocator(11747): Geolocator position updates started
I/flutter (11747): Latitude: 37.421998333333335, Longitude: -122.084HEHEHE
W/System ( 9138): Ignoring header X-Firebase-Locale because its value was null.
I/flutter (11747): <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Gambar 4.12 Log Aplikasi pada Debug Console di VSCode.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 2.4 (Low)

“AV:N/AC:L/PR:H/UI:R/S:U/C:L/I:N/A:N”

Berdasarkan hasil pengujian, pada gambar 4.12 terlihat bahwa data sensitif berupa *field name* yang berhubungan dengan data karyawan, url API, serta data lokasi *longitude latitude* pengguna ditampilkan pada *console log*.

Recommendation: Menghilangkan semua tampilan log dari *console*. Salah satu cara yang bisa dilakukan adalah dengan menerapkan proses *code review* pada tahapan pengembangan aplikasi.

4.4.2.4 STORAGE-4: “No sensitive data is shared with third parties unless it is a necessary part of the architecture.”

Pengujian pada poin ini dilakukan dengan 2 tahap, yaitu analisis dan pencarian layanan *third-party* yang dapat dilakukan dengan menganalisis *traffic* pada seluruh fitur aplikasi menggunakan proses *intercept* pada *burpsuite*. Lalu, jika ditemukan adanya layanan *third-party* yang digunakan maka izin atas layanan tersebut dicek apakah hanya berupa layanan yang diperlukan dan tidak ada data sensitif di dalamnya.

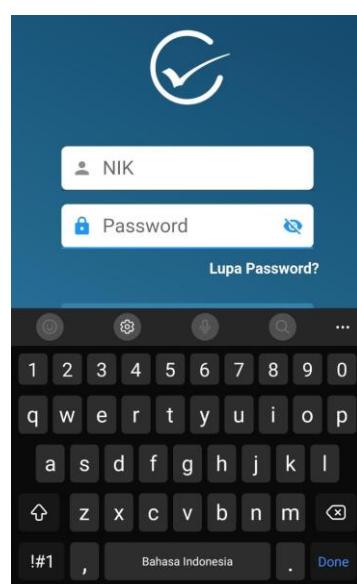
STATUS: “PASS”

Berdasarkan hasil analisis yang dilakukan dengan *intercept request* pada semua fitur yang ada, tidak ditemukan adanya data yang terkirim pada *third-party* yang tidak dikenal. Selain itu pada file *AndroidManifest.xml* juga tidak ditemukan *permission* yang mengurus terkait *third-party* tersebut.

4.4.2.5 STORAGE-5: “The keyboard cache is disabled on text inputs that process sensitive data.”

Pada poin ini pengujian dilakukan dengan mengecek setiap inputan apakah memiliki mekanisme yang menyimpan *cache keyboard* pada data sensitif.

STATUS: “PASS”



Gambar 4.13 Tampilan Keyboard Cache (di Atas Keyboard).

Berdasarkan hasil analisis dan gambar di atas, aplikasi tidak memiliki mekanisme untuk menampilkan *keyboard cache* sama sekali baik pada inputan data sensitif maupun inputan data lainnya.

4.4.2.6 STORAGE-6: “*No sensitive data is exposed via IPC mechanisms.*”

IPC atau *Inter Process Communication* adalah mekanisme yang disediakan sistem operasi untuk memungkinkan terjadinya pertukaran data dari dua atau lebih program atau proses. Pada sebuah aplikasi android, terdapat sebuah komponen yang bernama *content provider* (dapat dilihat pada AndroidManifest) untuk mengizinkan sebuah program atau aplikasi lain untuk mengakses dan mengubah data yang disimpan oleh aplikasi.

Pengujian pada poin ini diawali dengan dilakukannya pengecekan status *android:exported* pada setiap *content provider*. Jika terdapat status *true* pada *android:exported* maka data yang terekspor harus dicek apakah mengandung data sensitif atau tidak.

STATUS: "PASS"

```
<provider android:name="vn.hungnd.flutterdownloader.DownloadedFileProvider" android:exported="false" android:authorities="com.imfi.superapps.flutter_downloader.provider" android:grantUriPermissions="true" >
    <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/provider_paths" />
</provider>
<provider android:name="io.flutter.plugins.imagepicker.ImagePickerFileProvider" android:exported="false" android:authorities="com.imfi.superapps.flutter.image_provider" android:grantUriPermissions="true" >
    <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/flutter_image_picker_file_paths" />
</provider>
<activity android:theme="@android:style/Theme.NoTitleBar.Fullscreen" android:name="io.flutter.plugins.url_launcher.WebViewActivity" android:exported="false" />
<provider android:name="com.crazeader.openfile.FileProvider" android:exported="false" android:authorities="com.imfi.superapps.file_provider" android:grantUriPermissions="true" >
    <meta-data android:name="android.support.FILE_PROVIDER_PATHS" android:resource="@xml/filepath" />
</provider>
<provider android:name="com.google.firebase.provider.FirebaseInitProvider" android:exported="false" android:authorities="com.imfi.superapps.firebaseioinitprovider" android:initOrder="100" android:order="100" />
<provider android:name="androidx.startup.InitializationProvider" android:exported="false" android:authorities="com.imfi.superapps.androidx-startup" >
    <meta-data android:name="androidx.work.WorkManagerInitializer" android:value="androidx.startup" />
    <meta-data android:name="androidx.emoji2.text.EmojiCompatInitializer" android:value="androidx.startup" />
    <meta-data android:name="androidx.lifecycle.ProcessLifecycleInitializer" android:value="androidx.startup" />
</provider>
```

Gambar 4.14 Nilai Permission Export untuk Provider pada File AndroidManifest.xml

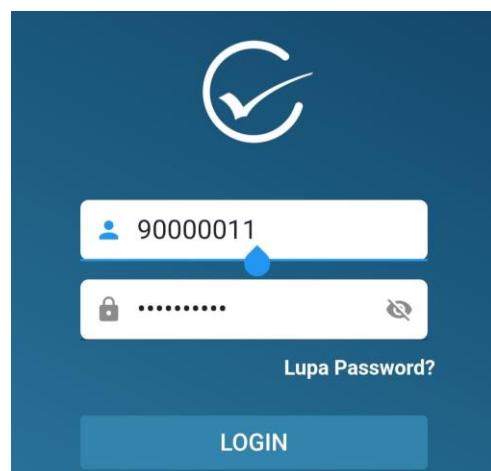
Berdasarkan gambar di atas dapat disimpulkan bahwa status `android:exported` bernilai `false` pada setiap `provider` yang ada. Hal ini berarti aplikasi tidak mengizinkan adanya data yang diekspor ke luar aplikasi lain.

4.4.2.7 STORAGE-7: “No sensitive data, such as passwords or pins, is exposed through the user interface.”

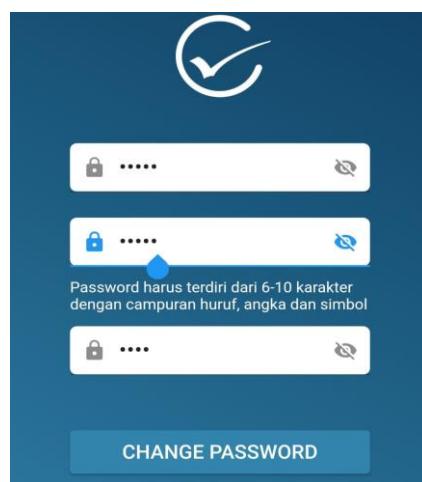
Data sensitif seperti *password* dan *pin* harus ditutup agar tidak bisa dilihat oleh pihak ketiga baik melalui aplikasi maupun terlihat oleh orang lain saat *input* data.

Pada poin ini penguji melakukan pengecekan dengan mencoba melakukan *input* ke semua *field input* untuk mengecek apakah *password* atau *pin* ditutup.

STATUS: “PASS”



Gambar 4.15 Input Field pada Halaman Login.



Gambar 4.16 Input Field pada Halaman Lupa Password.

Berdasarkan gambar 4.15 dan 4.16, *input field* yang memiliki data sensitif seperti *password* sudah dilakukan mekanisme penutupan *password* agar *password* tidak bisa terlihat oleh orang lain.

4.4.2.8 STORAGE-8: “*No sensitive data is included in backups generated by the mobile operating system.*”

Pada dasarnya, Android telah menyediakan fitur *backup* untuk aplikasi sebagai penjagaan jika sewaktu-waktu terjadi kerusakan atau kehilangan data. Namun, disisi lain data yang sensitif jika tersimpan di tempat *backup* data juga memiliki risiko untuk terjadi kebocoran data.

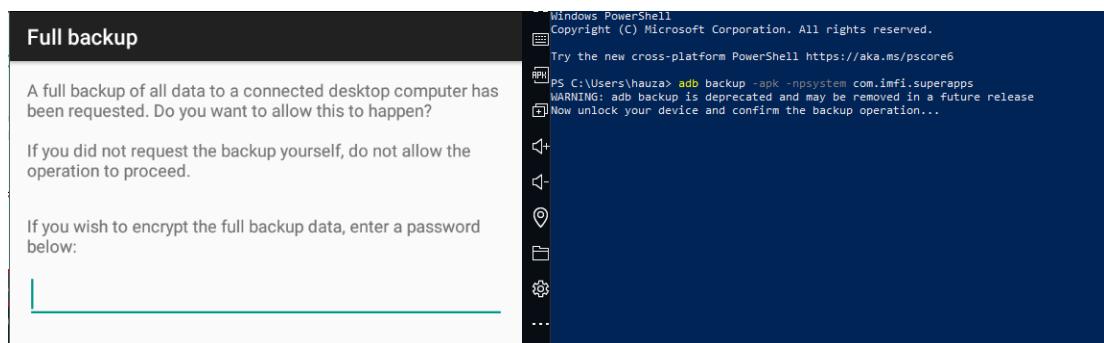
Pengujian pada poin ini dilakukan dengan mengecek *file* AndroidManifest.xml apakah memberikan ijin untuk melakukan *backup* pada bagian *application*. Selain itu juga dilakukan *backup* melalui adb.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 4.0 (Medium)

“AV:L/AC:L/PR:H/UI:R/S:U/C:L/I:L/A:L”



Gambar 4.17 Penggunaan Command “Adb Backup” pada Aplikasi.

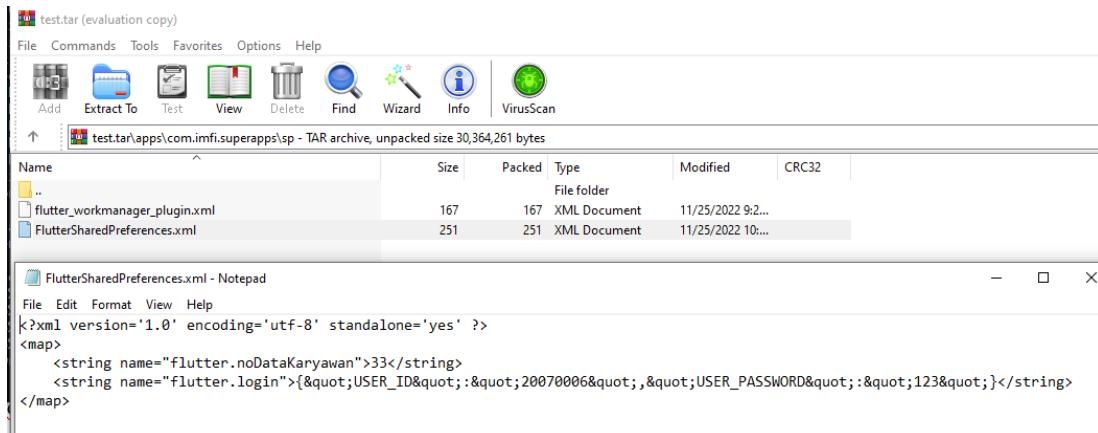
Berdasarkan analisis terhadap *file* AndroidManifest.xml tidak ditemukan adanya *permission* untuk mekanisme *back up* yang tertulis. Setelah itu, penguji melakukan uji coba *back up* menggunakan adb. ADB *back up* pada dasarnya akan berfungsi jika terdapat izin dari AndroidManifest.xml yang berupa *<application android:allowBackup="True">* atau tidak ada penulisan ijin *allowBackup* sama sekali.

Setelah dicoba, *back up* dapat dilakukan kepada seluruh sistem data dan *app* data namun bukan aplikasi itu sendiri. Hasil dari *backup* berupa *file* backup.ab yang kemudian dapat di-*unpack* menggunakan abe.jar (sebuah android *backup extractor*) dan jika dilihat terdapat *file local SharedPreferences* yang masih menyimpan data *user* yang

login. Maka dari itu pengujian mendapatkan kesimpulan “fail” pada poin ini.

```
C:\Users\hauza>java -jar abe.jar unpack backup.ab test.tar ""
This backup is encrypted, please provide the password
Password:
Calculated MK checksum (use UTF-8: true): 89A05B0F92390F511C1B62C0C341008B8CD8CECBDFEB20CEEDB3F92568B991E
0% 1% 2% 3% 4% 5% 6% 7% 8% 9% 10% 11% 12% 13% 14% 15% 16% 17% 18% 19% 20% 21% 22% 23% 24% 25% 26% 27% 28% 29% 30% 31% 32%
% 33% 34% 35% 36% 37% 38% 39% 40% 41% 42% 43% 44% 45% 46% 47% 48% 49% 50% 51% 52% 53% 54% 55% 56% 57% 58% 59% 60% 61% 62%
% 63% 64% 65% 66% 67% 68% 69% 70% 71% 72% 73% 74% 75% 76% 77% 78% 79% 80% 81% 82% 83% 84% 85% 86% 87% 88% 89% 90% 91% 92%
% 93% 94% 95% 96% 97% 98% 99% 100%
30371840 bytes written to test.tar.
```

Gambar 4.18 Unpack File Backup Aplikasi Menggunakan abe.jar



Gambar 4.19 Isi dari File Backup SharedPreferences

Recommendation: Menuliskan ijin dan mendefinisikan nilai dari allowBackup pada file AndroidManifest.xml menjadi *false*.

4.4.2.9 STORAGE-9: “*The app removes sensitive data from views when moved to the background.*”

Pengujian pada poin ini dilakukan dengan membuka halaman yang memiliki data sensitif seperti halaman yang mengandung *password*, data nominal *reimburse*, peraturan perusahaan yang tidak bisa dilihat oleh pihak lain.

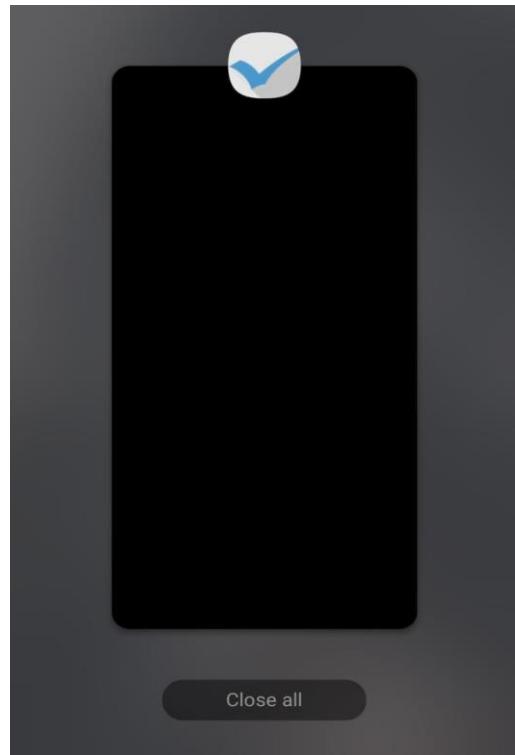
STATUS: “PASS”

```
private void a(d dvar) {
    new l(dvar, "flutter_windowmanager").e(this);
}
```

Gambar 4.20 Penggunaan Package yang Memiliki Fitur FLAG_SECURE.

```
@SuppressWarnings("deprecation")
private boolean validLayoutParams(int flag) {
    switch (flag) {
        case WindowManager.LayoutParams.FLAG_ALLOW_LOCK_WHILE_SCREEN_ON:
        case WindowManager.LayoutParams.FLAG_ALT_FOCUSABLE_IM:
        case WindowManager.LayoutParams.FLAG_DIM_BEHIND:
        case WindowManager.LayoutParams.FLAG_FORCE_NOT_FULLSCREEN:
        case WindowManager.LayoutParams.FLAG_FULLSCREEN:
        case WindowManager.LayoutParams.FLAG_HARDWARE_ACCELERATED:
        case WindowManager.LayoutParams.FLAG_IGNORE_CHEEK_PRESSES:
        case WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON:
        case WindowManager.LayoutParams.FLAG_LAYOUT_INSET_DECOR:
        case WindowManager.LayoutParams.FLAG_LAYOUT_IN_SCREEN:
        case WindowManager.LayoutParams.FLAG_LAYOUT_NO_LIMITS:
        case WindowManager.LayoutParams.FLAG_NOT_FOCUSABLE:
        case WindowManager.LayoutParams.FLAG_NOT_TOUCHABLE:
        case WindowManager.LayoutParams.FLAG_NOT_TOUCH_MODAL:
        case WindowManager.LayoutParams.FLAG_SCALED:
        case WindowManager.LayoutParams.FLAG_SECURE:
```

Gambar 4.21 Isi Package Window Manager pada Flutter.



Gambar 4.22 Tampilan View pada Halaman Peraturan Perusahaan Saat Dipindahkan ke Background.

Berdasarkan hasil analisis, penguji mendapatkan bahwa pada halaman peraturan perusahaan mekanisme untuk membuat *blank screen* saat aplikasi dipindahkan ke *background*. Sedangkan untuk halaman lain yang mengandung *password*, nominal *reimburse*, dan lainnya tidak perlu dilakukan mekanisme ini karena sudah ditutupi dengan *. Setelah itu, penguji juga mencari *package* pada *source code* hasil *reverse*

engineering dan *package* yang berkaitan di flutter yang digunakan untuk membuat mekanisme tersebut. Hasilnya ditemukan penggunaan FLAG_SECURE yang dikenal untuk menjaga halaman dari *screenshot* dan penampilan yang tidak aman (*background*).

4.4.2.10 STORAGE-10: “*The app does not hold sensitive data in memory longer than necessary, and memory is cleared explicitly after use.*”

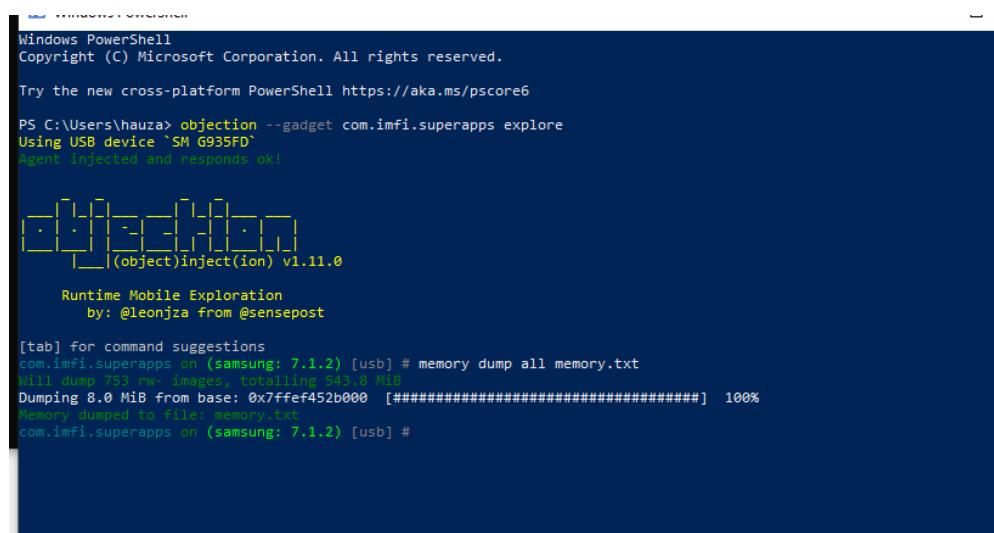
Pengujian pada poin ini dilakukan dengan menjalankan keseluruhan fitur pada aplikasi, lalu menggunakan *tools* objection untuk melakukan pengecekan pada *memory* menggunakan *command memory dump*. Jika aplikasi dibuktikan masih menyimpan data sensitif setelah aplikasi tidak digunakan maka poin dapat dinyatakan sebagai “fail”.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 4.0 (Medium)

“AV:L/AC:L/PR:H/UI:R/S:U/C:L/I:L/A:L”



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\hauza> objection --gadget com.imfi.superapps explore
Using USB device `SM G935FD`
Agent injected and responds ok!

[object]inject(i0n) v1.11.0

Runtime Mobile Exploration
by: @leonjza from @sensepost

[tab] for command suggestions
com.imfi.superapps on (samsung: 7.1.2) [usb] # memory dump all memory.txt
will dump 753 rw+ images, totalling 543.8 MiB
Dumping 8.0 MiB from base: 0x7ffef452b000 [#####] 100%
Memory dumped to file: memory.txt
com.imfi.superapps on (samsung: 7.1.2) [usb] #
```

Gambar 4.23 Dump Memory Menggunakan Objection pada Aplikasi.

Gambar 4.24 Isi File Memory yang Telah Diambil dalam Bentuk Teks.

Berdasarkan gambar 4.24 dapat disimpulkan bahwa setelah dilakukan *memory dump* pada aplikasi yang sudah tidak digunakan, ditemukan bahwa aplikasi masih menyimpan data sensitif berupa user_id dan user_password pada memori.

Recommendation: Menghapus atau melakukan enkripsi pada data sensitif yang kemungkinan terdapat pada memori untuk mencegah kebocoran data.

4.4.2.11 STORAGE-11: “*The app enforces a minimum device-access-security policy, such as requiring the user to set a device passcode.*”

Pengujian dilakukan dengan menjalankan aplikasi pada sebuah perangkat yang tidak memenuhi minimum *device-access-security policy* seperti tidak terpasang *passcode* pada *device*, telah di root, dan lainnya.

STATUS: “FAIL”

Berdasarkan hasil analisis yang dilakukan, aplikasi dapat berjalan dengan baik pada perangkat yang tidak memiliki *passcode* dan atau sudah di root.

Recommendation: Membuat kebijakan minimum dengan mewajibkan penggunaan passcode pada device.

4.4.2.12 STORAGE-12: “*The app educates the user about the types of personally identifiable information processed, as well as security best practices the user should follow in using the app.*”

Pengujian dilakukan dengan menjalankan semua fitur dan mengecek isi ketentuan, panduan dan deskripsi dari aplikasi. Pada dasarnya aplikasi harus memiliki panduan atau policy yang disampaikan kepada penggunanya yang berisi ketentuan-ketentuan terkait bagaimana data sensitif pengguna diolah dan bagaimana keamanan diterapkan pada aplikasi.

STATUS: “FAIL”

Berdasarkan hasil analisis tidak ditemukan adanya edukasi dalam bentuk apapun terkait bagaimana informasi diolah dan bagaimana kerja keamanan pada aplikasi.

Recommendation: Membuat edukasi baik dengan memberikan pop up berisi term and condition serta kebijakan-kebijakan terkait penggunaan dan penyimpanan data pada aplikasi saat pertama kali user mengakses aplikasi.

4.4.2.13 STORAGE-13: “*No sensitive data should be stored locally on the mobile device. Instead, data should be retrieved from a remote endpoint when needed and only be kept in memory.*”

Pada poin ini penguji mencoba menjalankan semua fitur di aplikasi untuk mengumpulkan data yang terambil. Setelah itu penguji melakukan pengecekan terhadap folder *local* android menggunakan adb shell untuk melihat data apa saja yang tersimpan pada *local storage*.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 4.0 (Medium)

“AV:L/AC:L/PR:H/UI:R/S:U/C:L/I:L/A:L”

Berdasarkan gambar 4.9 dapat terlihat bahwa terdapat data sensitif yaitu *password* yang tersimpan secara *local* pada folder SharedPreferences.

Recommendation: Tidak menyimpan data sensitif pada local storage. Namun, data dapat diambil secara berkala saat diperlukan saja oleh aplikasi melalui remote endpoint.

4.4.2.14 STORAGE-14: *“If sensitive data is still required to be stored locally, it should be encrypted using a key derived from hardware backed storage which requires authentication.”*

Pada poin ini, penguji dapat melihat terhadap hasil tes poin MASTG-STORAGE-13 untuk mengetahui apakah data sensitif yang tersimpan (jika ada) pada *local storage* telah dilakukan enkripsi.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 4.0 (Medium)

“AV:L/AC:L/PR:H/UI:R/S:U/C:L/I:L/A:L”

Berdasarkan gambar 4.9 dapat terlihat bahwa terdapat data sensitif yaitu *username* dan *password* yang tersimpan secara *local* tanpa adanya mekanisme enkripsi pada folder SharedPreferences.

Recommendation: Penyimpanan data sensitif pada local storage, jika diperlukan untuk mempermudah fungsionalitas aplikasi maka dilakukan enkripsi pada data tersebut. Lalu, key dari enkripsi tersebut disimpan di tempat penyimpanan yang aman.

4.4.2.15 STORAGE-15: *“The app’s local storage should be wiped after an excessive number of failed authentication attempts.”*

Pengujian dilakukan dengan melakukan percobaan *login* secara acak pada aplikasi, lalu setelah percobaan *login* yang gagal telah dilakukan beberapa kali, dilakukan pengecekan pada *local storage* aplikasi pada kasus ini dalam SharedPreferences yang sebelumnya menyimpan data seperti nodata, userid, password.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 4.0 (Medium)

“AV:L/AC:L/PR:H/UI:R/S:U/C:L/I:L/A:L”

```

12-06 09:33:17.060 3191 3191 I art      : at void android.os.Handler.dispatchMessage(android.os.Message)           com.android.phone
(Handler.java:95)                                com.android.printservice.recommendation com.google.android.syncadapters.calendar
12-06 09:34:14.859 3191 3088 I Flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           com.google.android.contacts
isServerErron: true, isOvertime: false, message: Failed<br />Password yang Anda masukan tidak sesuai.), stackTrace com.google.android.dts
null)                                              com.android.providers.calendar com.jet.parkir_let
12-06 09:34:20.204 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           com.joeykrim.app
isServerErron: true, isOvertime: false, message: Failed<br />Password yang Anda masukan tidak sesuai.), stackTrace com.joeykrim.rootcheck
null)                                              com.android.providers.downloads com.microvirt.download
12-06 09:34:26.648 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           com.microvirt.guide
isServerErron: true, isOvertime: false, message: Failed<br />Password yang Anda masukan tidak sesuai.), stackTrace com.microvirt.insider
null)                                              com.android.providers.media com.microvirt.launcher
12-06 09:34:30.073 5089 5089 W FA       : Service connection failed: ConnectionResult{statusCode=SERVICE_INVALID,           com.android.providers.settings com.microvirt.memime
resolution=null, message=null} com.android.server.telecom com.microvirt.tools
com.google.android.gms.common.internal.zzh.handleMessage(Unknown Source) com.android.settings com.txteapps.wifidb
12-06 09:34:30.984 5089 5133 W flayer_android_utils: at com.google.android.gms.common.internal.zzh.handleMessage(Unknown Source) com.google.android.storagebackup
12-06 09:34:33.515 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           SM-G935FD:/data/data # cd com.lmfi.superapps
isServerErron: true, isOvertime: false, message: Failed<br />Password yang Anda masukan tidak sesuai.), stackTrace SM-G935FD:/data/data/com.lmfi.superapps # ls
null)                                              app_flutter app_textures cache databases lib shared_prefs
12-06 09:34:37.020 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           SM-G935FD:/data/data/com.lmfi.superapps # cd shared_prefs
isServerErron: true, isOvertime: false, message: Failed<br />Password yang Anda masukan tidak sesuai.), stackTrace SM-G935FD:/data/data/com.lmfi.superapps/shared_prefs # ls
null)                                              FlutterSharedPreferences.xml flutter_image_picker_shared_preference.xml id.flutter
12-06 09:34:40.833 1847 1860 E RemotePrintpooler: at android.os.Handler.dispatchMessage(Handler.java:95)           WebViewChromPref.xml flutter_workmanager_plugin.xml
12-06 09:34:46.933 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           <map>
isServerErron: true, isOvertime: false, message: Failed<br />Password yang Anda masukan tidak sesuai.), stackTrace <string name="flutter_noDataKaryawan">3</string>
null)                                              SM-G935FD:/data/data/com.lmfi.superapps/shared_prefs #
12-06 09:35:17.256 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           </map>
isServerErron: true, isOvertime: false, message: Failed<br />Password yang Anda masukan tidak sesuai.), stackTrace
null)                                              <xml version="1.0" encoding="utf-8" standalone="yes" >
12-06 09:35:18.485 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           <map>
isServerErron: true, isOvertime: false, message: Failed<br />Password yang Anda masukan tidak sesuai.), stackTrace <string name="flutter_noDataKaryawan">3</string>
null)                                              SM-G935FD:/data/data/com.lmfi.superapps/shared_prefs #
12-06 09:35:33.523 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           </map>
isServerErron: true, isOvertime: false, message: Your NIK Is Not Registered), stackTrace null
12-06 09:35:37.786 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           SM-G935FD:/data/data/com.lmfi.superapps/shared_prefs #
isServerErron: true, isOvertime: false, message: Failed<br />Password yang Anda masukan tidak sesuai.), stackTrace
null)                                              <map>
12-06 09:35:43.721 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           <string name="flutter_noDataKaryawan">3</string>
isServerErron: true, isOvertime: false, message: Your NIK Is Not Registered), stackTrace null
12-06 09:35:54.944 3191 3088 I flutter : ProviderError<AuthModel>(error: ErrorState<isNoInternet: false,           </map>
isServerErron: true, isOvertime: false, message: Failed<br />Password yang Anda masukan tidak sesuai.), stackTrace
null)

```

Gambar 4.25 Log Percobaan Login (Kiri) dan Isi File SharedPreferences (Kanan)

Pada gambar 4.25 di bagian kiri telah ditampilkan sebuah log yang berisi percobaan *login* yang telah dilakukan lebih dari sepuluh kali. Berdasarkan gambar tersebut dilakukan pengecekan pada *local storage* yang berada pada gambar 4..25 di bagian kanan yang masih mengandung nodatakaryawan sedangkan data *user id* dan *password* telah terhapus. Hal ini berarti data pada *local storage* tidak sepenuhnya terhapus yang menyebabkan poin ini “fail”.

Recommendation: Menghapus data pada local storage ketika aplikasi mengalami gagal login sebanyak beberapa kali untuk menghindari adanya serangan bruteforce.

4.4.3 MASTG-CRYPTO: Cryptography Requirements

Pengujian tidak bisa dilakukan karena aplikasi tidak menggunakan *cryptography* pada datanya sama sekali.

4.4.4 MASTG-AUTH: Authentication and Session Management Requirements

4.4.4.1 AUTH-1: “*If the app provides users access to a remote service, some form of authentication, such as username/password authentication, is performed at the remote endpoint.*”

Pengujian pada poin ini dilakukan dengan menganalisis mekanisme otentikasi yang ada pada aplikasi. Setelah itu penguji dapat melihat apakah proses otentikasi dilakukan pada *remote endpoint*.

STATUS: “PASS”

The screenshot shows a NetworkMiner capture of a POST request to the URL `/IMFIOne/auth/SignIn.ashx`. The request is made with the following headers and body:

```

1 POST /IMFIOne/auth/SignIn.ashx HTTP/1.1
2 user-agent: Dart/2.18 (dart:io)
3 content-type: application/x-www-form-urlencoded;
  charset=utf-8
4 Accept-Encoding: gzip, deflate
5 Content-Length: 34
6 host: *
7 Connection: close
8
9 USER_ID=12345&USER_PASSWORD=123
  
```

The response is a JSON object containing the following data:

```

1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Length: 1038
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Tue, 29 Nov 2022 06:29:21 GMT
11 Connection: close
12
13 [
  {
    "ErrorCode": "0",
    "ErrorMessage": "SUCCESS",
    "DataValue": [
      {
        "NoD": "1",
        "NOD": "1",
        "NOD": "1",
        "isNotifikasiInternal": true,
        "KodeSanggah": "100",
        "isNotifikasiExternal": false,
        "isNotifikasi": "1",
        "isNotifikasi": "1",
        "isNotifikasi": "1"
      }
    ]
  }
]
  
```

Gambar 4.26 Intercept Jaringan yang Berisi Otentikasi pada Aplikasi.

Berdasarkan hasil analisis pada gambar di atas, dapat disimpulkan bahwa aplikasi menggunakan *remote endpoint* untuk otentikasinya setiap pengguna membuka halaman-halaman yang berisi data-data pribadi karyawan.

4.4.4.2 AUTH-2: “*If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user's credentials.*”

Penguji tidak dapat melakukan pengujian pada bagian ini karena aplikasi tidak menerapkan penggunaan *session* sama sekali

STATUS: “N/A”

4.4.4.3 AUTH-3: “*If stateless token-based authentication is used, the server provides a token that has been signed using a secure algorithm.*”

Penguji tidak dapat melakukan pengujian pada bagian ini karena aplikasi tidak menerapkan penggunaan *session* sama sekali

STATUS: “N/A”

4.4.4.4 AUTH-4: “*The remote endpoint terminates the existing session when the user logs out.*”

Penguji tidak dapat melakukan pengujian pada bagian ini karena aplikasi tidak menerapkan penggunaan *session* sama sekali

STATUS: “N/A”

4.4.4.5 AUTH-5: “*A password policy exists and is enforced at the remote endpoint.*”

Pengujian pada poin ini dilakukan dengan menganalisis *remote endpoint* pada fitur yang seharusnya menggunakan *policy password* dalam pembuatan kata sandinya dalam kasus aplikasi ini adalah halaman *change password*.

STATUS: “FAIL”

```

Request
Pretty Raw Hex
1 POST /IMFIOne/auth/ChangePassword.ashx HTTP/1.1
2 user-agent: Dart/2.18 (dart:io)
3 content-type: application/x-www-form-urlencoded;
charset=utf-8
4 Accept-Encoding: gzip, deflate
5 Content-Length: 94
6 host: [REDACTED]
7 Connection: close
8
9 USER_ID=1&USER_PASSWORD=1234234&USER_OLD_PASSWORD=1234234&USER_NEW_PASSWORD=123|
```

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Length: 91
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Tue, 29 Nov 2022 06:40:48 GMT
11 Connection: close
12
13 [
  {
    "ErrorCode":1,
    "ErrorMessage":
    "Password harus berisi minimal 6 karakter.",
    "DataValue":[
    ]
  }
]
```

Gambar 4.27 Intercept Jaringan pada Halaman Ubah Password Untuk Policy Minimal 6 Karakter.

```

Request
Pretty Raw Hex
1 POST /IMFIOne/auth/ChangePassword.ashx HTTP/1.1
2 user-agent: Dart/2.18 (dart:io)
3 content-type: application/x-www-form-urlencoded;
charset=utf-8
4 Accept-Encoding: gzip, deflate
5 Content-Length: 97
6 host: [REDACTED]
7 Connection: close
8
9 USER_ID=1&USER_PASSWORD=1234234&USER_OLD_PASSWORD=1234234&USER_NEW_PASSWORD=123123|
```

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Length: 73
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Tue, 29 Nov 2022 06:41:46 GMT
11 Connection: close
12
13 [
  {
    "ErrorCode":0,
    "ErrorMessage":"Berhasil Ganti Password",
    "DataValue":[
    ]
  }
]
```

Gambar 4.28 Intercept Jaringan pada Halaman Ubah Password untuk Policy Mengandung Karakter Special, Huruf Besar dan Alphanumeric.

Berdasarkan gambar 4.27 penerapan *password policy* sudah dilakukan untuk ketentuan *password* harus memiliki minimal 6 karakter, namun pada gambar 4.28 penerapan *password* yang seharusnya pada aplikasi diberlakukan ketentuan harus memiliki huruf besar, angka dan karakter *special* tidak diterapkan.

Recommendation: Melakukan penerapan kebijakan password yang sama pada remote endpoint dengan aplikasi. Pada kasus ini remote endpoint harus menggunakan kebijakan yang sama yaitu minimal 6 karakter, memiliki huruf besar, angka dan karakter *special*.

4.4.4.6 AUTH-6: “*The remote endpoint implements a mechanism to protect against the submission of credentials an excessive number of times.*”

Brute-force attack adalah percobaan untuk mencari *password* menggunakan mekanisme untuk mencari setiap kemungkinan huruf, angka dan simbol sampai kombinasi yang tepat ditemukan.

Pengujian ini dilakukan dengan tindakan *bruteforce* pada *remote endpoint* menggunakan *tools* burpsuite. *Bruteforce* dilakukan sebanyak 300-500 kali untuk menemukan *rate limit* yang dibuat pada proses *otentifikasi* aplikasi.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 5.3 (Medium)

"AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L"

Request ^	Payload 1	Payload 2	Status	Error	Timeout	Length	Comment
298	20445900	username	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
299	2044253	oliver	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
300	2044551	prince	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
301	2044850	beach	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
302	2045150	amateur	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
303	2045451	7777777	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
304	2045753	muffin	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
305	2046056	redsox	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
306	2046360	star	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
307	2046665	testing	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
308	2046971	shannon	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
309	2047278	murphy	200	<input type="checkbox"/>	<input type="checkbox"/>	378	
310	2047586	frank	200	<input type="checkbox"/>	<input type="checkbox"/>	270	

Request	Response
Pretty	Raw
Hex	

```

1 POST /IMFOne/auth/Signin.ashx HTTP/1.1
2 user-agent: Dart/2.18 (dartio)
3 content-type: application/x-www-form-urlencoded; charset=utf-8
4 Accept-Encoding: gzip, deflate
5 Content-Length: 36
6 Host: [REDACTED]
7 Connection: close
8
9 USER_ID=2047278&USER_PASSWORD=murphy

```

352 of 497 0 matches

Gambar 4.29 Uji Bruteforce pada Halaman Login Menggunakan Burpsuite.

Berdasarkan hasil percobaan serangan *brute-force*, dapat disimpulkan bahwa tidak ada penerapan *rate limit* pada percobaan yang melewati batas wajar sebanyak 300-500 percobaan. Maka dari itu hasil pengujian menyatakan poin ini “fail”.

Recommendation: Menerapkan mekanisme rate limit pada setiap request percobaan login pada remote endpoint. Seberapa banyak remote endpoint memberikan limit dapat bergantung pada kebijakan perusahaan.

4.4.4.7 AUTH-7: “*Sessions are invalidated at the remote endpoint after a predefined period of inactivity and access tokens expire.*”

Penguji tidak dapat melakukan pengujian pada bagian ini karena aplikasi tidak menerapkan penggunaan *session* sama sekali

STATUS: “N/A”

4.4.4.8 AUTH-8: “*Biometric authentication, if any, is not event-bound (i.e. using an API that simply returns "true" or "false"). Instead, it is based on unlocking the keychain/keystore.*”

Penguji tidak dapat melakukan pengujian pada bagian ini karena aplikasi tidak menerapkan penggunaan *biometric authentication* sama sekali.

STATUS: “N/A”

4.4.4.9 AUTH-9: “*A second factor of authentication exists at the remote endpoint and the 2FA requirement is consistently enforced.*”

Two-Factor Authentication (2FA) diibaratkan sebagai lapisan tambahan pada keamanan dimana 2FA menggunakan metode yang mewajibkan 2 jenis otentikasi untuk mendapatkan akses ke data.

Pengujian pada poin ini dilakukan dengan mengecek terkait keberadaan mekanisme yang menggunakan 2FA untuk otentikasinya seperti pada fitur *login* dan lainnya.

STATUS: “FAIL”

Berdasarkan hasil analisis, tidak ditemukan adanya penerapan 2FA pada setiap proses *login* dan transaksi. Maka hasil pengujian pada poin ini dinyatakan ”fail”.

Recommendation: Menerapkan penggunaan 2FA pada proses-proses aplikasi yang dianggap sensitif seperti login, ganti password, dan transaksi sensitif.

4.4.4.10 AUTH-10: “*Sensitive transactions require step-up authentication.*”

Pengujian pada poin ini dilakukan dengan cara menganalisis aplikasi pada fitur yang termasuk ke dalam transaksi sensitif. Analisis dapat dilakukan dengan melakukan *intercept* pada jaringan menggunakan burpsuite untuk melihat apakah fitur yang termasuk ke dalam transaksi sensitif telah menggunakan otentikasi tambahan.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 2.6 (Low)

“AV:N/AC:H/PR:L/UI:R/S:U/C:N/I:L/A:N”

```

Request
Pretty Raw Hex
1 POST /UpdateApproval.ashx HTTP/1.1
2 user-agent: Dart/2.18 (dart:io)
3 content-type: application/x-www-form-urlencoded;
charset=utf-8
4 Accept-Encoding: gzip, deflate
5 Content-Length: 90
6 host: [REDACTED].com
7 Connection: close
8
9 NO_DATA_APPROVAL=<REDACTED>&DisubmitOleh=<REDACTED>&DECISION=
Revise&USER_ID=<REDACTED>&USER_PASSWORD=<REDACTED>

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Length: 127
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Wed, 04 Jan 2023 07:23:27 GMT
11 Connection: close
12
13 [
  {
    "ErrorCode": 0,
    "ErrorMessage": "Revise Success",
    "DataValue": [
      {
        "NoApproval": "<REDACTED>/100/<REDACTED>",
        "NoDataApproval": "<REDACTED>"
      }
    ]
  }
]

```

Gambar 4.30 Intercept Request pada Halaman Approver.

Pada aplikasi telah ditemukan fitur yang termasuk ke dalam transaksi sensitif yaitu proses pengajuan dari karyawan yang dapat disetujui oleh *user* yang memiliki wewenang pada perusahaan seperti kepala departemen, kepala divisi dan lainnya. Hasil pengujian dengan melakukan *intercept* membuktikan bahwa tidak ada penerapan *step up authentication*, namun hanya menerapkan proses otentifikasi ulang seperti *login*.

Recommendation : Menerapkan mekanisme otentifikasi tambahan seperti menggunakan *authorization*. Pada dasarnya *authorization* termasuk dalam mekanisme *step-up authentication* dimana sistem akan memberikan *authorization token* berdasarkan *user*, *role*, dan hak akses yang berbeda.

4.4.4.11 AUTH-11: *“The app informs the user of all sensitive activities with their account. Users are able to view a list of devices, view contextual information (IP address, location, etc.), and to block specific devices.”*

Untuk memperkuat keamanan akun, aplikasi perlu menyediakan fitur dimana pengguna dapat melihat daftar data perangkat yang mengikat akun tersebut pada perangkat mereka. Data perangkat yang ditampilkan dapat seperti nama perangkat, *ip address* dan lokasi dimana perangkat yang mengikat akun tersebut berada. Dengan ini jika terjadi hal yang tidak diinginkan seperti *login* tidak jelas, pelaku dapat dengan mudah memblokir perangkat tersebut.

Pengujian pada poin ini dilakukan dengan menganalisis aplikasi dan mencari fitur yang sesuai dengan poin ini.

STATUS: “FAIL”

Berdasarkan hasil analisis, tidak ditemukan adanya penggunaan fitur yang sesuai dengan poin ini.

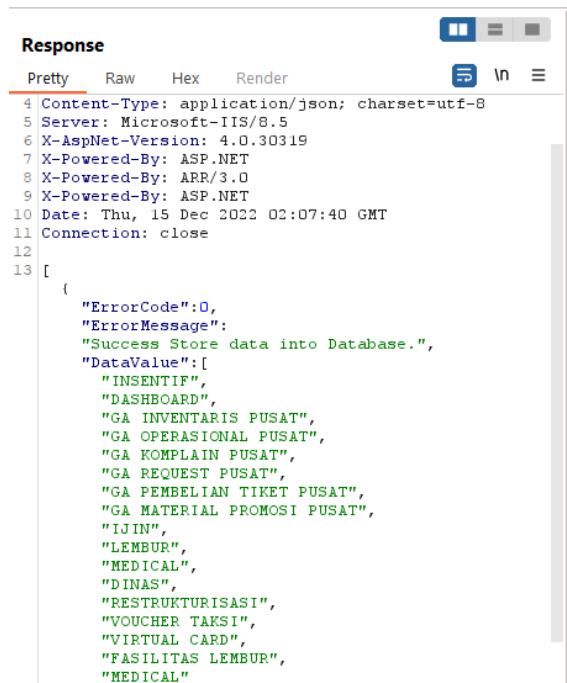
Recommendation: Menerapkan mekanisme yang dapat melihat dan mengatur aktivitas pada setiap device.

4.4.4.12 AUTH-12: “Authorization models should be defined and enforced at the remote endpoint.”

Authorization model merupakan sebuah kontrol pada aplikasi yang berisi kumpulan *privilege* atau fitur apa saja yang dapat dilakukan atau diakses setiap pengguna. Sebagai contoh, karyawan dengan jabatan supervisor dapat mengakses halaman insentif sedangkan staf biasa tidak bisa.

Pengujian pada poin ini dilakukan dengan menganalisis fitur pada aplikasi yang dimiliki oleh pengguna dengan *privilege* tertentu dan dengan pengguna yang lain. Setelah itu dilakukan analisis pada hasil *intercept request* terhadap *remote endpoint* sebelum melakukan akses pada fitur tersebut.

STATUS: “PASS”



```

Response
Pretty Raw Hex Render
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Thu, 15 Dec 2022 02:07:40 GMT
11 Connection: close
12
13 [
  {
    "ErrorCode": 0,
    "ErrorMessage": "Success Store data into Database.",
    "DataValue": [
      "INSENTIF",
      "DASHBOARD",
      "GA INVENTARIS PUSAT",
      "GA OPERASIONAL PUSAT",
      "GA KOMPLAIN PUSAT",
      "GA REQUEST PUSAT",
      "GA PEMBELIAN TIKET PUSAT",
      "GA MATERIAL PROMOSI PUSAT",
      "IJIN",
      "LEMBUR",
      "MEDICAL",
      "DINAS",
      "RESTRUKTURISASI",
      "VOUCHER TAKSI",
      "VIRTUAL CARD",
      "FASILITAS LEMBUR",
      "MEDICAL"
    ]
  }
]

```

Gambar 4.31 Respon yang Berisi Daftar Authority yang Dimiliki Setiap User.

Berdasarkan gambar 4.31 dapat disimpulkan bahwa *authorization model* telah terdefinisi pada *remote endpoint* saat pengguna melakukan *request* pada menu yang berarti hasil pengujian pada poin ini yaitu “pass”.

4.4.5 MASTG-NETWORK: Network Communication Requirements

4.4.5.1 NETWORK-1: “Data is encrypted on the network using TLS. The secure channel is used consistently throughout the app.”

Pengujian dilakukan dengan melihat *protocol* yang digunakan pada jaringan *remote endpoint* dengan melakukan *intercept request* menggunakan burp suite untuk melihat apakah aplikasi termasuk *remote endpoint* telah menggunakan SSL/TLS.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 5.4 (Medium)

“AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N”

#	Host	Method	URL	Params
42	http://[REDACTED]	GET	/IMFIOne/[REDACTED]	
41	http://[REDACTED]	POST	/IMFIOne/[REDACTED]	... ✓
40	http://[REDACTED]	POST	/IMFIOne/[REDACTED]	... ✓
39	http://[REDACTED]	POST	/IMFIOne/[REDACTED]	... ✓
38	http://[REDACTED]	POST	/IMFIOne/[REDACTED]	... ✓
37	http://[REDACTED]	POST	/IMFIOne/[REDACTED]	... ✓
36	http://[REDACTED]	POST	/IMFIOne/[REDACTED]	✓
35	http://[REDACTED]	POST	/IMFIOne/[REDACTED]	✓
34	http://[REDACTED]	POST	/IMFIOne/[REDACTED]	✓
33	http://[REDACTED]	POST	/IMFIOne/[REDACTED]	✓
31	http://[REDACTED]	GET	/IMFIOne/[REDACTED]	
32	http://[REDACTED]	GET	/IMFIOne/[REDACTED]	

Gambar 4.32 Daftar History Hasil Intercept yang Berisi Host dari API.

Hasil pengujian mendapatkan bahwa *remote endpoint* masih menggunakan HTTP yang berarti tidak ada penggunaan *protocol* SSL/TLS pada aplikasi.

Recommendation: Menggunakan *protocol* SSL/TLS pada aplikasi dengan menggunakan HTTPS.

4.4.5.2 NETWORK-2: “*The TLS settings are in line with current best practices, or as close as possible if the mobile operating system does not support the recommended standards.”*

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 5.4 (Medium)

“AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N”

Berdasarkan poin NETWORK-1, maka dapat disimpulkan aplikasi tidak memenuhi poin ini.

Recommendation: Menggunakan *protocol* SSL/TLS pada aplikasi dengan menggunakan HTTPS.

4.4.5.3 NETWORK 3: “*The app verifies the X.509 certificate of the remote endpoint when the secure channel is established.*

Only certificates signed by a trusted CA are accepted.”

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 5.4 (Medium)

“AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N”

Berdasarkan poin NETWORK-1, maka dapat disimpulkan aplikasi tidak memenuhi poin ini.

Recommendation: Menggunakan *protocol* SSL/TLS pada aplikasi dengan menggunakan HTTPS.

4.4.5.4 NETWORK-4: “*The app either uses its own certificate store, or pins the endpoint certificate or public key, and subsequently does not establish connections with endpoints that offer a different certificate or key, even if signed by a trusted CA.*”

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 5.4 (Medium)

“AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N”

Berdasarkan poin NETWORK-1, maka dapat disimpulkan aplikasi tidak memenuhi poin ini.

Recommendation: Menggunakan *protocol SSL/TLS* pada aplikasi dengan menggunakan HTTPS.

4.4.5.5 NETWORK-5: “*The app doesn't rely on a single insecure communication channel (email or SMS) for critical operations, such as enrollments and account recovery.*”

Pengujian pada poin ini dilakukan dengan cara melakukan analisis dan pengetesan pada fitur yang kemungkinan memiliki transaksi atau operasi yang *critical* seperti lupa *password*.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 4.6 (Medium)

“AV:N/AC:H/PR:L/UI:R/S:U/C:L/I:L/A:L”

```

Request
Pretty Raw Hex
1 POST /IMF...ForgotPassword.ashx HTTP/1.1
2 user-agent: Dart/2.18 (dart:io)
3 content-type: application/x-www-form-urlencoded;
charset=utf-8
4 Accept-Encoding: gzip, deflate
5 Content-Length: 16
6 host: [REDACTED]
7 Connection: close
8
9 [REDACTED]
10 [REDACTED]

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Length: 243
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Fri, 02 Dec 2022 08:06:35 GMT
11 Connection: close
12
13 [
  {
    "ErrorCode": 0,
    "ErrorMessage": "Password Sementara sudah dikirimkan ke ri*****r.*****@████████.com. Password Sementara hanya berlaku selama 15 menit. Segera cek Email untuk melakukan login dan mengganti Password Sementara Anda.",
    "DataValue": []
  }
]

```

Gambar 4.33 Intercept Request pada Halaman Lupa Password.

Berdasarkan gambar 4.33, ditemukan bahwa aplikasi masih menggunakan *single insecure communication channel* yaitu email sebagai metode untuk melakukan akun *recovery*.

Recommendation: Tidak menggunakan saluran komunikasi yang tidak aman seperti email dan sms yang mengirimkan *password* sementara atau *link* untuk mengganti *password*. Sebagai gantinya, aplikasi dapat menggunakan multiple factor *authentication* seperti kode otp, *biometric* dan lain-lain. Selain itu, dapat juga dilakukan mekanisme manual oleh administrator untuk me-reset kata sandi karena aplikasi merupakan aplikasi internal.

4.4.5.6 NETWORK-6: “*The app only depends on up-to-date connectivity and security libraries.*”

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 5.4 (Medium)

“AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N”

Berdasarkan poin NETWORK-1, maka dapat disimpulkan aplikasi tidak menggunakan jaringan yang dianggap *up-to-date* dari segi keamanannya dengan tidak menggunakan tls atau ssl sama sekali

Recommendation: Menggunakan *protocol* SSL/TLS pada aplikasi dengan menggunakan HTTPS.

4.4.6 MASTG-PLATFORM: Platform Interaction Requirements

4.4.6.1 PLATFORM-1: “The app only requests the minimum set of permissions necessary.”

Pengujian dilakukan dengan menganalisis penggunaan *permission* yang ada pada aplikasi. Hal ini dilakukan dengan menganalisis data pada file AndroidManifest.xml (yang diperoleh dari tools mobSF) untuk melihat daftar *permission*, lalu mencari tahu kegunaan pada setiap *permission* dan fitur apa yang menggunakan *permission* tersebut.

STATUS: “PASS”

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.hardware.location.gps" />
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.SET_ALARM" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES" />
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
<uses-permission android:name="android.permission.USE_FULL_SCREEN_INTENT" />
<uses-permission android:name="android.permission.SCHEDULE_EXACT_ALARM" />
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
```

Gambar 4.34 Daftar Permission pada File AndroidManifest.xml

Berdasarkan hasil analisis yang dilakukan dengan mempelajari kebutuhan dan penggunaan setiap *permission* pada aplikasi maka dapat dinyatakan setiap *permission* penting dan digunakan dengan baik. Sebagai contoh, *permission* untuk lokasi, kamera, *storage*, digunakan untuk absen. Lalu alarm, *vibrate*, *post notification* untuk *notification*, dan lainnya.

4.4.6.2 PLATFORM-2: “*All inputs from external sources and the user are validated and if necessary sanitized. This includes data received via the UI, IPC mechanisms such as intents, custom URLs, and network sources.*”

Pengujian dilakukan dengan melakukan injeksi kepada setiap inputan baik yang terlihat atau melalui UI maupun dari yang lainnya. Injeksi yang dilakukan dapat berupa *SQL Injection*, *XML Injection*, *Fragment Injection*. Untuk memenuhi poin ini, inputan baik dari *user* maupun dari luar harus sudah divalidasi dan disanitasi sehingga tidak bisa dilakukan injeksi sama sekali.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 10 (High)

“AV:N/AC:L/PR:N/UI:N/S:C/C:N/I:H/A:H”

Pada poin MASTG-ARCH-2, dapat disimpulkan bahwa aplikasi tidak memiliki validasi dan sanitasi tambahan yang menyebabkan *remote endpoint* dapat dilakukan *SQL Injection*.

Recommendation: Melakukan sanitasi *input* atau pencegahan seperti *prepare statement*, *remove html tag* dan lain-lain, pada semua *field* inputan.

4.4.6.3 PLATFORM-3: “*The app does not export sensitive functionality via custom URL schemes, unless these mechanisms are properly protected.*”

Custom URL Scheme adalah sebuah konsep atau metode pada pengembangan android yang memungkinkan pengguna untuk membuka aplikasi android melalui *hyperlink*.

STATUS: “N/A”

Aplikasi tidak menggunakan *custom URL scheme* sehingga tidak dapat dilakukan pengujian.

4.4.6.4 PLATFORM-4: “*The app does not export sensitive functionality through IPC facilities, unless these mechanisms are properly protected.*”

Inter Process Communication (definisi bisa dilihat pada subbab 4.3.1.) biasanya digunakan dalam pengembangan aplikasi *mobile* sebagai teknik untuk proses berkomunikasi dengan hal seperti soket jaringan atau *shared file*.

Pengujian pada poin ini dilakukan dengan melakukan penganalisisan terhadap file AndroidManifest.xml apakah melakukan *export* pada setiap elemen yaitu *<intent>*, *<service>*, *<provider>* dan *<receiver>*. Jika dilakukan export pada elemen tersebut, perlu diperhatikan fungsi atau data apa yang diekspor jika tidak mengandung hal yang sensitif maka poin ini dapat dikatakan “*pass*”.

STATUS: "PASS"



Gambar 4.35 Daftar Activity, Service, Receiver dan Provider yang Terexport.

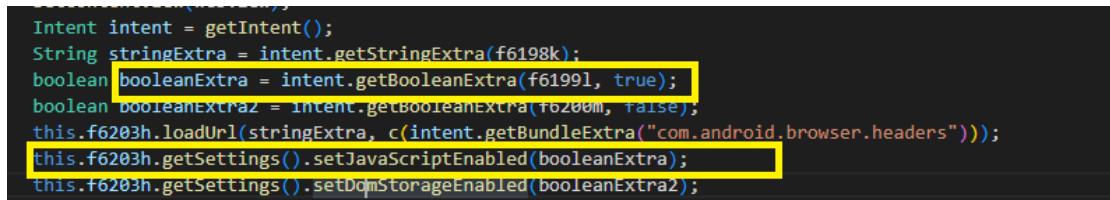
Gambar 4.36 Pengecekan IPC yang memiliki nilai android:exported=true.

Berdasarkan kedua gambar tersebut dapat disimpulkan bahwa IPC yang dapat dieksplor tidak memiliki data atau fungsi sensitif, sebagai contoh MainActivity hanya berisi *intent filter* untuk *launcher* dan main yang merupakan *function* umum pada aplikasi.

4.4.6.5 PLATFORM-5: “*JavaScript is disabled in WebViews unless explicitly required.*”

Class *WebView* adalah ekstensi dari class *View* android yang memungkinkan penampilan halaman web pada aplikasi *mobile*. *WebView* secara *default* tidak menyertakan fitur apapun dari browser web seperti kontrol navigasi atau kolom url melainkan hanya tampilan halaman web/url tersebut.

STATUS: “PASS”



```

Intent intent = getIntent();
String stringExtra = intent.getStringExtra(f6198k);
boolean booleanExtra = intent.getBooleanExtra(f61991, true);
boolean booleanExtra2 = intent.getBooleanExtra(f6200m, false);
this.f6203h.loadUrl(stringExtra, c(intent.getBundleExtra("com.android.browser.headers")));
this.f6203h.getSettings().setJavaScriptEnabled(booleanExtra);
this.f6203h.getSettings().setDomStorageEnabled(booleanExtra2);

```

Gambar 4.37 Nilai Webview Setting untuk Penggunaan Javascript dari File Java Hasil Reverse Engineering.

```

/* renamed from: l reason: collision with root package name */
private static String f61991 = "enableJavaScript";

```

Gambar 4.38 Nilai String dari Variable f61991 yang Digunakan pada Gambar 4.37.

```

return WebView(
    gestureRecognizers: {}...addAll([
        Factory<VerticalDragGestureRecognizer>(
            () => VerticalDragGestureRecognizer(),
        ), // or null // Factory
        Factory<OneSequenceGestureRecognizer>(
            () => EagerGestureRecognizer(),
        ), // Factory
    ]),
    initialUrl: "about:blank",
    javascriptMode: JavascriptMode.unrestricted,
    navigationDelegate: (NavigationRequest request) {
        return NavigationDecision.navigate;
    },
)

```

Gambar 4.39 Nilai Javascript Mode pada File .dart untuk Class Webview yang Dipanggil.

Berdasarkan gambar 4.37 terdapat penggunaan *class webview* yang terlihat melakukan pembuatan *intent* yang berisi *setting* untuk *enable javascript*. Selain itu, setelah melakukan konfirmasi pada *source code* di flutter terlihat bahwa terdapat penggunaan parameter yang memberikan nilai javascript mode *unrestricted* atau tidak dibatasi. Namun setelah hasil diskusi dengan pihak pengembang diketahui bahwa mode javascript diperlukan untuk fitur *report approval*.

4.4.6.6 PLATFORM-6: “*WebViews are configured to allow only the minimum set of protocol handlers required (ideally, only https is supported). Potentially dangerous handlers, such as file, tel and app-id, are disabled.*”

Pengujian pada poin ini, dilakukan dengan melakukan *intercept* pada *traffic* jaringan halaman *webview*. Pengujian diawali dengan menggunakan *burpsuite* untuk mengecek apakah *protocol handler* yang digunakan telah dilengkapi dengan *tls* atau *ssl*. Hal ini berarti *protocol handler* yang digunakan adalah *HTTPS* dan bukan *HTTP*. Jika hal ini tidak sesuai ketentuan maka tidak perlu dilakukan pengujian lainnya.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 4.2 (Medium)

"AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:L/A:N"

383	http://[REDACTED]	GET	/NotAutho		✓	200	6261	HTML	aspx	Title
382	http://[REDACTED]	GET	/NotAutho			302	513	HTML	aspx	Object moved
381	http://[REDACTED]	GET	/InsentifA		✓	302	511	HTML	aspx	Object moved
380	http://[REDACTED]	GET	/Login.aspx			200	69004	HTML	aspx	IMFI Dashboard Branch
379	http://[REDACTED]	POST	/NotAutho		✓	302	439	HTML	aspx	Object moved
378	http://[REDACTED]	GET	/NotAutho			200	6261	HTML	aspx	Title
377	http://[REDACTED]	GET	/NotAutho			302	513	HTML	aspx	Object moved
376	http://[REDACTED]	GET	/InsentifA		✓	302	511	HTML	aspx	Object moved
375	http://[REDACTED]	GET	/Login.aspx			200	69004	HTML	aspx	IMFI Dashboard Branch
374	http://[REDACTED]	POST	/NotAutho		✓	302	439	HTML	aspx	Object moved
373	http://[REDACTED]	GET	/NotAutho			200	6261	HTML	aspx	Title
372	http://[REDACTED]	GET	/NotAutho			302	513	HTML	aspx	Object moved

Gambar 4.40 Daftar Host dari History Intercept pada Halaman Insentif yang Menggunakan Webview.

Berdasarkan gambar 4.40 dapat dibuktikan yaitu halaman pada aplikasi yang menggunakan *webview* yaitu halaman insentif yang masih menggunakan *protocol handler* HTTP yang berarti tidak adanya penggunaan *protocol* TLS ataupun SSL. Hal ini berarti hasil pengujian tidak memenuhi ketentuan poin yang menyebabkan “fail”.

Recommendation: Menggunakan *protocol* SSL/TLS pada aplikasi dengan menggunakan *protocol handler* HTTPS.

4.4.6.7 PLATFORM-7: *“If native methods of the app are exposed to a WebView, verify that the WebView only renders JavaScript contained within the app package.”*

Pengujian pada poin ini, dilakukan pengecekan pada aplikasi melalui *source code* untuk mencari penggunaan javascript, lalu dilakukan *intercept* pada *request* atau respon *handler* untuk melihat apakah *webview* hanya melakukan *rendering* pada javascript yang terkandung di *app package* atau juga melakukan *rendering* pada javascript dari tempat lain.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 4.6 (Medium)

“AV:N/AC:H/PR:L/UI:R/S:U/C:L/I:L/A:L”

Request

Pretty Raw Hex

1 GET
/NotAuthorized/NotAuthorized/SessionExpired.aspx
HTTP/1.1
2 Host: [REDACTED].com
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Linux; Android 7.1.2; SM-G935FD Build/N2G48H; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/68.0.3440.70 Mobile Safari/537.36
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US
8 Cookie: ASP.NET_SessionId=cn4p4234q1qndzj1ylahaqrk
9 X-Requested-With: com.in
10 Connection: close
11
12

Original response

Pretty Raw Hex Render

```
checkKey()" enctype="multipart/form-data">
<div class="aspNetHidden">
  <input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
  <input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
  <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="" />
  /wEPDwU0JTc3ODAxNTI0D2QWagID2QWAmYpZBYCZg9kFQnCBQ88KwAGAQ88KwAHQAQWAH4KQ3NzUG9zdGZpeAULQnV0dG9u1FRydWVkgAEFH19fQ29udHJvbHNSZXF1aXJUG5zdEjhYztLZLx1fxYBBQ1pbWdcQG55ZUxvZ2luIUba4BxO+lqTDjPlyYuM1TDwOWW+j9jDWp7z9xbYdo=" />
</div>
<script type="text/javascript">
//<![CDATA[
var theForm = document.forms['form1'];
if (!theForm) {
    theForm = document.form1;
}
function __doPostBack(eventTarget, eventArgument) {
    if (!theForm.onsubmit || (theForm.onsubmit() != false)) {
        theForm.__EVENTTARGET.value = eventTarget;
        theForm.__EVENTARGUMENT.value = eventArgument;
        theForm.submit();
    }
}
//]]>
</script>

<script src="/WebResource.axd?d=Mhaae-SevpthMc-kDdwVR19vrigfxMvav_CipOCWG2YwInM_oPUN3V4g2Uvu5VsrgSWzT" type="text/javascript">
</script>
```

Gambar 4.41 Respon dari Intercept Request pada Halaman Webview.

Request

Pretty Raw Hex ⌂ In ⌂

1 GET /NotAuthorized/NotAuthorized/SessionExpired.aspx
HTTP/1.1
2 Host: [REDACTED].com
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Linux; Android 7.1.2; SM-G935FD Build/N2G48H; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/4.0 Chrome/68.0.3440.70 Mobile Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US
8 Cookie: ASP.NET_SessionId=cn4p4234qlqndzjlylahaqrk
9 X-Requested-With: [REDACTED]
10 Connection: close
11
12

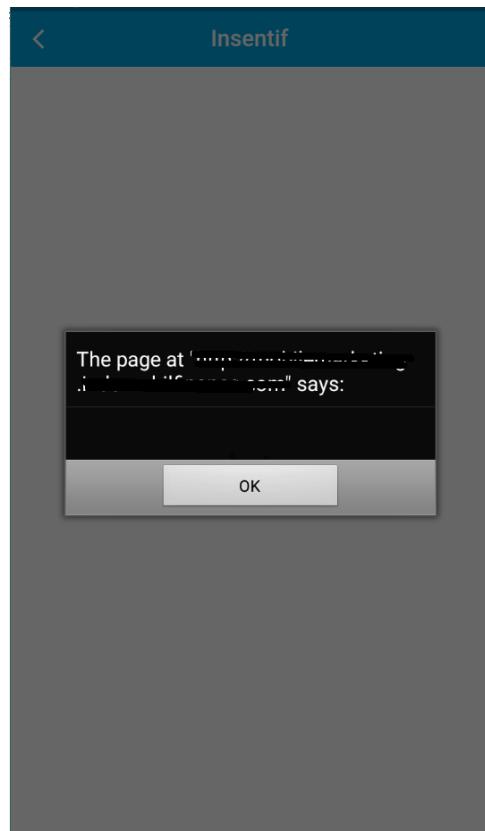
Edited response ⌂

Pretty Raw Hex Render ⌂ In ⌂

24 <input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value="" />
25 <input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
26 <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value=""/>
27 </div>
28
29 <script type="text/javascript">
30 //<![CDATA[
31 var theForm = document.forms['form1'];
32 if (!theForm) {
33 theForm = document.form1;
34 }
35 function __doPostBack(eventTarget, eventArgument) {
36 if (!theForm.onsubmit || (theForm.
37 onsubmit) != false) {
38 theForm.__EVENTTARGET.value =
39 eventTarget;
40 theForm.__EVENTARGUMENT.value =
41 eventArgument;
42 theForm.submit();
43 }
44 }
45 //]]>
46 alert(document.cookie);
47 </script>

Gambar 4.42 Percobaan Tamper pada Script untuk Webview.

Berdasarkan hasil pengujian ditemukan salah satu halaman yang menggunakan *webview* yaitu *insentif* yang masih menggunakan *script* berupa javascript kiriman dari hasil *request* ke *endpoint*. Hal ini berarti respons dapat di-*tamper* menggunakan *script* yang berbahaya.



Gambar 4.43 Hasil Percobaan Tamper pada Halaman Insentif.

Setelah dilakukan pengetesan dengan menambahkan *script alert* sederhana untuk mengecek apakah *script* dapat berjalan dengan normal. Dibuktikan pada gambar 4.43 bahwa *script* yang diedit melalui respons dapat berjalan tanpa adanya masalah. Hal ini berarti *webview* masih melakukan *rendering javascript code* dari luar paket aplikasi yang menyebabkan poin ini “fail”.

Recommendation: Hanya melakukan *rendering javascript* pada *script* yang terkandung di paket aplikasi Android dan tidak menerima *script* dari luar.

4.4.6.8 PLATFORM-8: “Object deserialization, if any, is implemented using safe serialization APIs.”

Serialization merujuk kepada proses konversi dari sebuah data *object* (*class*, *dictionary*) ke dalam format tertentu seperti JSON, XML, HDF5 agar dapat disimpan ke dalam suatu *file* atau dikirim melalui jaringan. *Object* dalam format tertentu tadi lalu dibuat kembali sebagai data *object* dalam bentuk seperti *class* melalui proses *deserialization*.

Pengujian pada poin ini dilakukan dengan melakukan analisis pada kemungkinan proses *deserialization* yang tidak aman seperti: 1) penerimaan respon yang berupa data tidak dilakukan filter atau dimasukkan ke dalam sebuah *class* khusus terlebih dahulu yang menyebabkan aplikasi menerima data begitu saja. 2) tidak adanya sanitasi terhadap data yang dikirim seperti penggunaan *special character*.

STATUS: “PASS”

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Length: 12042
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Wed, 04 Jan 2023 07:31:04 GMT
11 Connection: close
12 [
13   {
    "ErrorCode":0,
    "ErrorMessage":"Success",
    "DataValue":[
      {
        "NoDataBerita":9,
        "JudulBerita":"Test Bulletin",
        "TanggalTerbit":"28/06/2021 1:00:00",
        "KontenMobile":"
<u003chtml><u003e<u003chead><u003e<n> \u003ctitle><u003etitle<u003e</title><u003e<n> \u003c/head><u003e<n> \u003cbody><u003e<u003c
</body><u003e<u003c/html><u003e",
        "BeritaThumbnailURL":
        "~/UploadDocument/Media/pngguru.com (1).png",
        "Attachment":[
          {
            "Deskripsi":"Test",
            "FileName":"8P597e8En7A272W63jV427L5X83mCR.jpg",
            "FilePath":
            "C:\\inetpub\\wwwroot\\IMS\\UploadDocument\\DocumentAttachment\\Berita\\2\\W3HbRIWg3P07QlyNBMxCBQ2H4W837T.xls"
          }
        ],
        "NoDataKategoriBerita":3
      }
    ]
  }
]

```

Gambar 4.44 Keadaan Awal Saat Nilai Key yang Dikirim Benar pada Halaman iBulletin.

```

Edited response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Length: 12039
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Wed, 04 Jan 2023 07:31:43 GMT
11 Connection: close
12 [
13   {
    "ErrorCode":0,
    "ErrorMessage":"Success",
    "DataValue":[
      {
        "NoDataBerita":9,
        "NamaBerita":"Test Bulletin",
        "TanggalTerbit":"28/06/2021 1:00:00",
        "KontenMobile":"
<u003chtml><u003e<u003chead><u003e<n> \u003ctitle><u003etitle<u003e</title><u003e<n> \u003c/head><u003e<n> \u003cbody><u003e<u003c
</body><u003e<u003c/html><u003e",
        "BeritaThumbnailURL":
        "~/UploadDocument/Media/pngguru.com (1).png",
        "Attachment":[
          {
            "Deskripsi":"Test",
            "FileName":"8P597e8En7A272W63jV427L5X83mCR.jpg",
            "FilePath":
            "C:\\inetpub\\wwwroot\\IMS\\UploadDocument\\DocumentAttachment\\Berita\\2\\W3HbRIWg3P07QlyNBMxCBQ2H4W837T.xls"
          }
        ],
        "NoDataKategoriBerita":3
      }
    ]
  }
]

```

Gambar 4.45 Keadaan Setelah Dilakukan Pengubahan Pada Nilai Key yang Dikirim ke Halaman iBulletin.

Penguji melakukan *tampering* pada respon dengan mengubah salah satu *key* pada json yaitu judul berita untuk mengecek apakah aplikasi melakukan *handle* dengan baik dan telah melakukan *filtering* untuk mengambil *key* yang diperlukan saja pada aplikasi. Hasil pengujian membuktikan bahwa aplikasi tetap berjalan normal ketika menerima *key* tambahan pada json dan menampilkan data kosong pada *key* yang seharusnya diambil yaitu judul berita.

4.4.6.9 PLATFORM-9: “*The app protects itself against screen overlay attacks. (Android only).*”

Pengujian dilakukan dengan mengecek pemberian *permission* SYSTEM_ALERT_WINDOW pada file AndroidManifest.xml. SYSTEM_ALERT_WINDOW adalah *permission* yang memungkinkan aplikasi untuk berjalan pada *background* di atas aplikasi lain. Contoh aplikasi yang paling umum adalah *icon* lingkaran dari aplikasi *facebook messenger*.

STATUS: “PASS”

Berdasarkan gambar 4.34, tidak ditemukan adanya penggunaan *permission* SYSTEM_ALERT_WINDOW

4.4.6.10 PLATFORM-10: “*A WebView's cache, storage, and loaded resources (JavaScript, etc.) should be cleared before the WebView is destroyed.*”

Pengujian pada poin ini dilakukan dengan melakukan pengecekan pada *cache* dan *storage* lokal milik aplikasi. Selain itu pengecekan terhadap *resource* yang telah di-load seperti javascript dapat dicek pada penyimpanan memori aplikasi. Poin ini dapat terpenuhi bila data terkait *webview* pada *cache*, *storage* dan *resource* terkait javascript telah dihapus saat halaman *webview* tidak digunakan lagi.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 1.8 (Low)

“AV:L/AC:H/PR:H/UI:R/S:U/C:L/I:N/A:N”

Gambar 4.46 Dump Memory untuk Mengecek String dari Script yang Tadi Ditambahkan (subbab 4.2.6.7).

Berdasarkan hasil pengujian menggunakan *memory dump*, *resource* yang berupa *script code javascript* (yang telah ditamper) melalui respons masih tersimpan pada memori setelah halaman *webview* tidak digunakan.

Gambar 4.47 Isi File Cache Webview pada Local Storage.

```
2|SM-G935FD:/data/data/com.imfi.superapps/app_webview # cat Cookies
A=8@t@?tablecookiescookies*CREATE TABLE cookies (creation_utc INTEGER NOT NULL,host_key TEXT NOT NULL,name TEXT NOT NULL,value TEXT NOT NULL,path TEXT NOT NULL,expires_utc INTEGER NOT NULL,is_secure INTEGER NOT NULL,is_httponly INTEGER NOT NULL,last_access_utc INTEGER NOT NULL,has_expired INTEGER NOT NULL DEFAULT 1,is_persistent INTEGER NOT NULL DEFAULT 1,priority INTEGER NOT NULL DEFAULT 1,encrypted_value BLOB DEFAULT '',firstpartyonly INTEGER NOT NULL DEFAULT 0,UNIQUE (host_key, name, path))-*@A@ndexsqlite_autoindex_cookies_1cookies*#P@5$0@/tablemeta*CREATE TABLE meta(key LONGVARCHAR,value LONGVARCHAR,*last_compatible_version@Q@version@G@#human_status@autoindex_meta@met@0000
@0@0@0@0@0@U@=ost/_NQ@B@#com.imfi.superapps.comASP.NET_SessionIdcn4p4234qlqndzjlylahaqrk//N@q@#a
@0@0@0@0@0@U@o
w #
```

Gambar 4.48 Isi File Cookies Webview pada Local Storage.

Berdasarkan gambar 4.47 dan 4.48, terlihat pada penyimpanan lokal yang masih tersimpan data cache dari *webview* (gambar 4.47) yang berisi *header* dari *request* termasuk url API, path *file API*, dan lainnya. Selain itu juga terlihat data *cookie* (gambar 4.48) yang berisi data seperti url API dan *session ID*

Recommendation: Menghapus isi file pada *local storage* ketika *webview* sudah tidak digunakan. Selain itu, data sensitif yang tersimpan pada *local storage* telah dienkripsi terlebih dahulu.

4.4.6.11 PLATFORM-11: “*Verify that the app prevents usage of custom third-party keyboards whenever sensitive data is entered.*”

Penggunaan *custom keyboard* pada aplikasi android dapat menyebabkan munculnya potensi disisipkannya kode berbahaya yang dapat berjalan tanpa disadari pengguna. *Custom keyboard* yang disediakan oleh *third-party* dapat tersisipi *script* yang dapat mengirimkan semua *history* ketikan *user* kepada pelaku yang biasanya disebut sebagai *keylogger*.

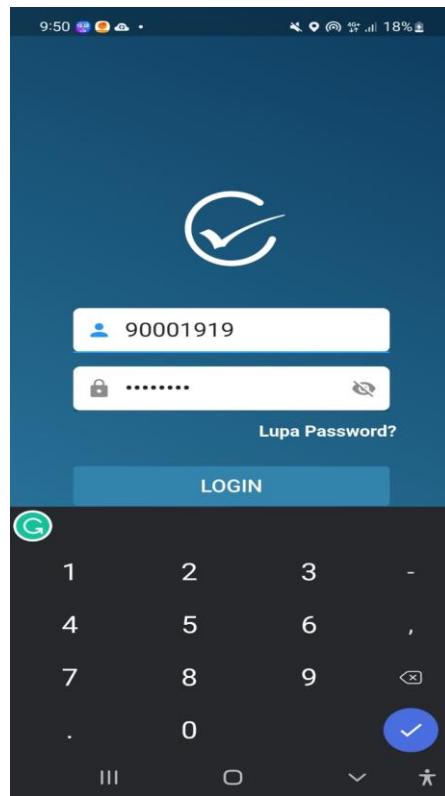
Pengujian dilakukan dengan menjalankan aplikasi saat sedang menggunakan aplikasi *keyboard* tambahan.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 2.6 (Low)

“AV:N/AC:H/PR:L/UI:R/S:U/C:L/I:N/A:N”



Gambar 4.49 Penggunaan Keyboard dari Custom Third-Party.

Berdasarkan gambar 4.49 dapat dibuktikan bahwa aplikasi *third-party*, dalam kasus ini yaitu aplikasi *custom keyboard* Grammarly yang masih dapat digunakan termasuk pada penulisan data sensitif seperti *user id* dan *password*.

Recommendation: Developer dapat menonaktifkan akses ke *third-party* keyboard dari konfigurasi aplikasi. Lalu, aplikasi dapat melakukan *mask* pada data sensitif seperti *password* dengan *string random* sebagai representasi *password* asli.

4.4.7 MASTG-CODE: *Code Quality and Build Setting Requirements*

4.4.7.1 CODE-1: “*The app is signed and provisioned with a valid certificate, of which the private key is properly protected.*”

Pengujian dilakukan dengan mengecek status *signature* dari aplikasi menggunakan *tools* seperti mobSF dan jarsigner. Aplikasi minimal harus memenuhi skema v1 dan v2 yaitu v1 digunakan untuk penandatanganan JAR dan v2 digunakan untuk skema penandatanganan APK yang mendukung versi Android 7.0 ke atas. Sedangkan untuk skema v3 yang mendukung versi Android 9.0 ke atas tidak terlalu di wajibkan.

STATUS: “PASS”

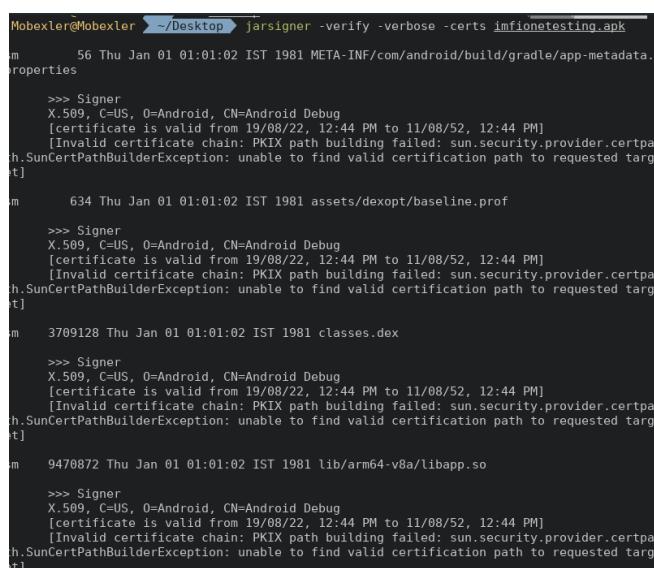


```

SIGNER CERTIFICATE

APK is signed
v1 signature: True
v2 signature: True
v3 signature: False
Found 1 unique certificates
Subject: CN=Android Debug, O=Android, C=US
  
```

Gambar 4.50 Status Penandatanganan Aplikasi pada Skema v1, v2 dan v3 dari Tool MobSF.



```

Mobexler@Mobexler ~/Desktop > jarsigner -verify -verbose -certs imfionetesting.apk
m      56 Thu Jan 01 01:01:02 IST 1981 META-INF/com/android/build/gradle/app-metadata.properties
>>> Signer
X.509, C=US, O=Android, CN=Android Debug
[certificate is valid from 19/08/22, 12:44 PM to 11/08/52, 12:44 PM]
[Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]

m      634 Thu Jan 01 01:01:02 IST 1981 assets/dexopt/baseline.prof
>>> Signer
X.509, C=US, O=Android, CN=Android Debug
[certificate is valid from 19/08/22, 12:44 PM to 11/08/52, 12:44 PM]
[Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]

m      3709128 Thu Jan 01 01:01:02 IST 1981 classes.dex
>>> Signer
X.509, C=US, O=Android, CN=Android Debug
[certificate is valid from 19/08/22, 12:44 PM to 11/08/52, 12:44 PM]
[Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]

m      9470872 Thu Jan 01 01:01:02 IST 1981 lib/arm64-v8a/libapp.so
>>> Signer
X.509, C=US, O=Android, CN=Android Debug
[certificate is valid from 19/08/22, 12:44 PM to 11/08/52, 12:44 PM]
[Invalid certificate chain: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target]
  
```

Gambar 4.51 Status Signer dan Certificate pada Setiap File dengan Tool Jarsigner.

Berdasarkan gambar 4.50 dan 4.51 aplikasi sudah memenuhi ketentuan minimum aplikasi dalam penandatanganan *certificate* yaitu skema v1 dan v2.

4.4.7.2 CODE-2: “The app has been built in release mode, with settings appropriate for a release build (e.g. non-debuggable).”

Pengujian dilakukan dengan mengecek status *android:debuggable* yang berada pada file *AndroidManifest.xml*. Jika nilai *debuggable* aplikasi *false* atau tidak ada status *android:debuggable* sama sekali yang berarti nilai *android:debuggable* adalah *false* mengikuti nilai *default*-nya. Hal ini berarti dapat disimpulkan bahwa aplikasi dalam keadaan non *debuggable*.

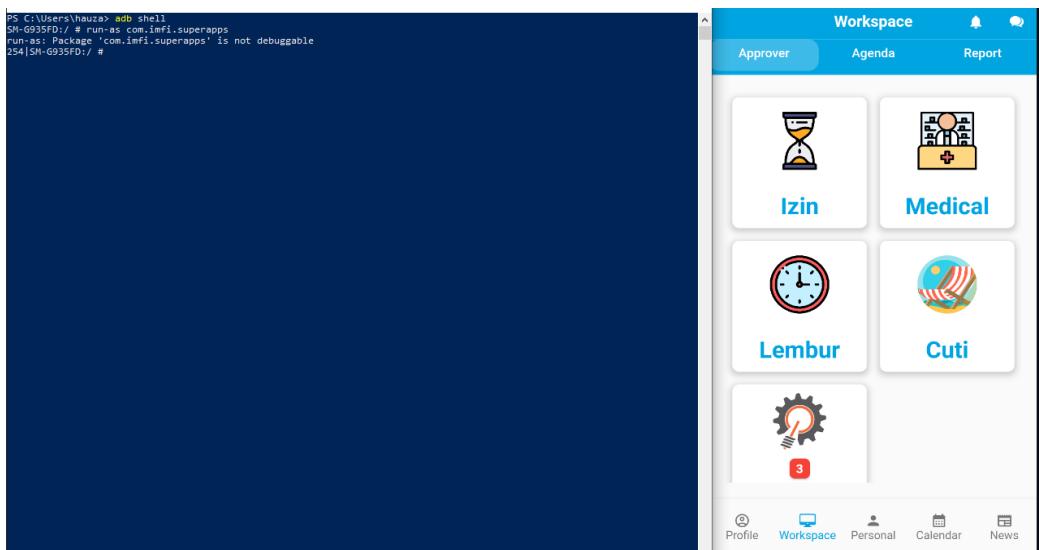
STATUS: “PASS”

```

1.  <?xml version="1.0" encoding="utf-8"?>
2.  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3.    android:versionName="1.15.8" android:compileSdkVersion="32" android:compileSdkVersionCodename="12" package="com
4.      xiaomi.superapps">
5.      <uses-sdk android:minSdkVersion="21" android:targetSdkVersion="32" />
6.      <uses-permission android:name="android.permission.INTERNET" />
7.      <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
8.      <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
9.      <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
10.     <uses-permission android:name="android.permission.CAMERA" />
11.     <uses-permission android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS" />
12.     <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
13.     <uses-permission android:name="android.permission.WAKE_LOCK" />
14.     <uses-permission android:name="android.hardware.location.gps" />
15.     <uses-permission android:name="android.permission.SET_ALARM" />
16.     <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
17.     <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
18.     <uses-permission android:name="android.permission.WRITE_CONTACTS" />
19.     <uses-permission android:name="android.permission.READ_CONTACTS" />
20.     <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
21.     <uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES" />
22.     <uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
23.     <uses-permission android:name="android.permission.USE_FULL_SCREEN_INTENT" />
24.     <uses-permission android:name="android.permission.SCHEDULE_EXACT_ALARM" />
25.     <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
26.   </queries>
27.   <intent>
28.     <action android:name="android.intent.action.VIEW" />
29.   </intent>
30.   <package android:name="com.google.android.apps.maps" />
31. </queries>
32. <uses-feature android:glesVersion="0x00020000" android:required="true" />
33. <uses-feature android:name="android.hardware.camera" android:required="false" />
34. <uses-feature android:name="android.hardware.camera.front" android:required="false" />
35. <uses-feature android:name="android.hardware.camera.autofocus" android:required="false" />
36. <uses-feature android:name="android.hardware.camera.flash" android:required="false" />

```

Gambar 4.52 Isi File *AndroidManifest.xml* untuk Mencari Nilai *Android:debuggable*



Gambar 4.53 Penggunaan Command *run-as* Pada Adb Shell untuk Status Debug Aplikasi.

Pada gambar 4.52, setelah dilakukan pencarian pada nilai *android:debuggable*, hasil pencarian menunjukkan tidak ditemukan nilai *android:debuggable*. Hal ini berarti aplikasi menggunakan nilai *default*-nya yaitu *android:debuggable false* yang berarti aplikasi tidak dapat dilakukan debug. Selain itu, pada gambar 4.53 juga terlihat bahwa aplikasi dalam status *non-debuggable* dengan menggunakan *command run-as* pada adb shell.

4.4.7.3 CODE-3: “*Debugging symbols have been removed from native binaries.*”

Pada dasarnya saat *native binary* dikompilasi menjadi *shared object* atau file berekstensi .so terdapat sebuah simbol *debug* yang terkandung di dalamnya. *Debugging Symbols* sendiri merepresentasikan sebuah informasi debug atau informasi tambahan yang umumnya digunakan untuk memudahkan proses *debugging* pada tahap pengembangan. Informasi tambahan yang terkandung dapat berupa informasi seperti nama fungsi, variabel, dan beberapa bit yang terkait dengan *source code* asli.

Pengujian dilakukan dengan melakukan pengecekan pada *file shared object* untuk melihat apakah masih terkandung debug symbol dan informasi *debugging* yang seharusnya dihilangkan saat proses *build*,

STATUS: “PASS”

```
Mobexler@Mobexler: ~/Desktop/app-release/lib/arm64-v8a objdump --syms libapp.so
libapp.so:      file format elf64-little
SYMBOL TABLE:
no symbols

Mobexler@Mobexler: ~/Desktop/app-release/lib/arm64-v8a objdump --syms libflutter.so
libflutter.so:   file format elf64-little
SYMBOL TABLE:
no symbols

Mobexler@Mobexler: ~/Desktop/app-release/lib/arm64-v8a objdump --syms libimage_processing_util_jni.so
libimage_processing_util_jni.so:    file format elf64-little
SYMBOL TABLE:
no symbols

Mobexler@Mobexler: ~/Desktop/app-release/lib/arm64-v8a objdump --syms libbarhopper_v3.so
libbarhopper_v3.so:   file format elf64-little
SYMBOL TABLE:
no symbols
```

Gambar 4.54 Isi Native Debug Symbol pada File Shared Libraries .so.

Pada dasarnya aplikasi yang menggunakan Android SDK telah menghilangkan *native debug symbol* secara *default*. Hal ini berarti, aplikasi Imfi One yang tidak menggunakan android NDK sudah tidak memiliki debug *symbol* pada file .so (*shared object*). Untuk membuktikan lebih lanjut, penguji telah melakukan pengecekan pada tabel simbol di file *shared object* (gambar 4.54) untuk melihat apakah terdapat debug simbol dan terbukti bahwa tidak ada debug *symbol* sama sekali.

4.4.7.4 CODE-4: “*Debugging code and developer assistance code (e.g. test code, backdoors, hidden settings) have been removed. The app does not log verbose errors or debugging messages.*”

Pada poin ini, pengujian dilakukan dengan menjalankan *command* pada adb untuk melihat log aplikasi, pada kasus ini penguji akan mencoba mencari sesuatu yang berhubungan dengan pesan dan *error*. Selain itu penguji juga melakukan pengujian dengan menjalankan proses *debugging* pada sebuah aplikasi flutter acak, lalu *exit* dan menjalankan aplikasi ImfiOne pada emulator tersebut.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 2.6 (Low)

“AV:N/AC:H/PR:L/UI:R/S:U/C:L/I:N/A:N”

```
PS C:\Users\hauza> adb logcat -v threadtime 127.0.0.1:21523 | select-string "message" | select-string "error"
12-06 09:34:14.854 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Failed<br />Password yang Anda masukkan tidak sesuai.), stackTrace:
null)
12-06 09:34:20.204 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Failed<br />Password yang Anda masukkan tidak sesuai.), stackTrace:
null)
12-06 09:34:26.640 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Failed<br />Password yang Anda masukkan tidak sesuai.), stackTrace:
null)
12-06 09:34:33.515 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Failed<br />Password yang Anda masukkan tidak sesuai.), stackTrace:
null)
12-06 09:34:37.020 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Failed<br />Password yang Anda masukkan tidak sesuai.), stackTrace:
null)
12-06 09:34:46.033 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Failed<br />Password yang Anda masukkan tidak sesuai.), stackTrace:
null)
12-06 09:35:17.256 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Failed<br />Password yang Anda masukkan tidak sesuai.), stackTrace:
null)
12-06 09:35:18.485 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Failed<br />Password yang Anda masukkan tidak sesuai.), stackTrace:
null)
12-06 09:35:33.523 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Your NIK Is Not Registered), stackTrace: null)
12-06 09:35:37.786 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Failed<br />Password yang Anda masukkan tidak sesuai.), stackTrace:
null)
12-06 09:35:43.721 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Your NIK Is Not Registered), stackTrace: null)
12-06 09:35:54.944 3191 3988 I flutter : ProviderError<AuthModel>(error: ErrorState(isNoInternet: false,
isServerError: true, isOvertime: false, message: Failed<br />Password yang Anda masukkan tidak sesuai.), stackTrace:
null)
```

Gambar 4.55 Menampilkan Log dengan Adb Logcat untuk Mencari Error atau Debugging Message.

Berdasarkan gambar 4.12 pada subbab 4.4.2.3, aplikasi masih menampilkan beberapa data seperti *url API*, data karyawan dan lainnya pada *debug console* setelah menjalankan aplikasi dengan emulator pada VS Code. Lalu, pada gambar 4.55 aplikasi masih menampilkan *verbose error* atau *debugging message*.

Recommendation: Menghilangkan semua tampilan *log* dari *console*. Salah satu cara yang bisa dilakukan adalah dengan menerapkan proses *code review* pada tahapan pengembangan aplikasi.

4.4.7.5 CODE-5: “All third party components used by the mobile app, such as libraries and frameworks, are identified, and checked for known vulnerabilities.”

Pengujian pada poin ini dilakukan dengan melakukan analisis menggunakan *tools* owasp dependency check pada setiap *library* dan *framework* yang digunakan pada aplikasi apakah *library* dan *framework* yang digunakan dapat teridentifikasi

dan sudah melalui pemeriksaan terhadap kelemahan-kelemahan yang ada.

STATUS: “PASS”

Gambar 4.56 Hasil Penggunaan Plugin OWASP Dependency Check.

Berdasarkan gambar di atas, dapat terlihat bahwa semua *third-party component* dapat diidentifikasi dan telah melalui pemeriksaan terkait kelemahan-kelemahan yang ada pada setiap komponennya.

4.4.7.6 CODE-6: “*The app catches and handles possible exceptions.*”

Pengujian dilakukan dengan melakukan *intercept traffic request* yang dilakukan pada semua fitur aplikasi khususnya pada proses otentikasi dan melakukan *tampering* pada respons dari *request*.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 6.4 (Medium)

“AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:N”

The screenshot shows the 'Original response' tab in Postman. The status code is 200 OK. The response body is a JSON object with an 'Error' key containing an array of objects. Each object has properties: 'ErrorCode' (5), 'ErrorMessage' ('Your NIK Is Not Registered'), and 'DataValue' (an empty array). The JSON structure is as follows:

```

1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Length: 87
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Thu, 01 Dec 2022 09:04:16 GMT
11 Connection: close
12
13 [
  {
    "ErrorCode": "5",
    "ErrorMessage": "Your NIK Is Not Registered",
    "DataValue": []
  }
]

```

Gambar 4.57 Tampilan Bentuk Awal Hasil Respon dari Login.

The screenshot shows the 'Edited response' tab in Postman. The status code is 200 OK. The response body is a JSON object with an 'Error' key containing an array of objects. Each object has properties: 'ErrorCode' (0), 'ErrorMessage' ('SUCCESS'), and 'DataValue' (an empty array). The JSON structure is as follows:

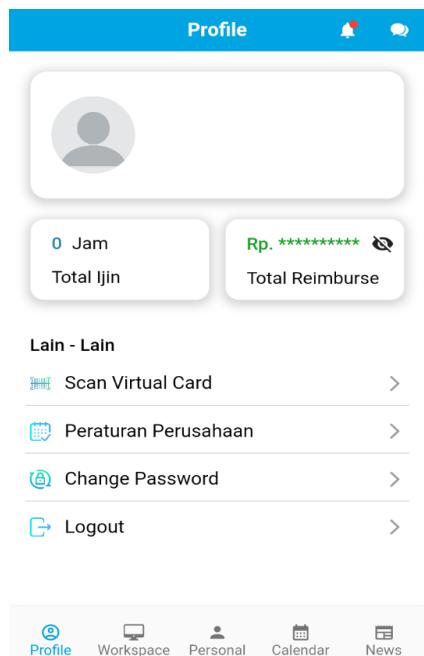
```

1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Length: 61
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Thu, 01 Dec 2022 07:41:02 GMT
11 Connection: close
12
13 [
  {
    "ErrorCode": "0",
    "ErrorMessage": "SUCCESS",
    "DataValue": []
  }
]

```

Gambar 4.58 Penguji Mencoba Tamper Respon dengan Respon Sukses.

Terlihat pada gambar 4.57 yang merupakan respons asli dari hasil *request* otentikasi yang salah. Ketika dilakukan edit pada respons (gambar 4.58) dengan *status code* dan *error message* yang umum lalu dikirimkan Kembali ke aplikasi jika *login* dapat tertembus berarti terjadi kesalahan aplikasi dalam *handle* kemungkinan *exception*.



Gambar 4.59 Tampilan Aplikasi Setelah Tamper Respon Saat Login.

Berdasarkan gambar 4.59, terbukti bahwa hasil edit respons tidak diurus dengan baik oleh aplikasi yang menyebabkan *login* dapat ditembus dengan tampilan data *user* yang kosong. Hal ini berarti aplikasi tidak memenuhi syarat yang menyebabkan hasil pengujian “*fail*”.

Recommendation: Menerapkan *session management*, agar developer dapat *handle* respons dengan menambahkan kondisi tambahan untuk mencocokkan *session* dengan *database*.

4.4.7.7 CODE-7: “Error handling logic in security controls denies access by default.”

Pada dasarnya *exception* biasanya terjadi saat aplikasi masuk ke dalam status abnormal atau *error* pada *logic code*. Pengujian pada poin ini dilakukan untuk memastikan bahwa aplikasi dapat mengurus adanya *exception* secara aman dengan cara menolak akses secara *default* dan tidak menampilkan informasi sensitif via UI atau log aplikasi.

Pengujian pada poin ini dapat dilakukan dengan beberapa cara seperti: 1) mengetik *value* yang tidak terduga ke dalam field yang berada pada UI aplikasi. 2) berinteraksi dengan aplikasi menggunakan *intent*, *public provider* dan *value* yang tidak terduga. 3) melakukan *tampering* pada komunikasi jaringan dan file yang tersimpan pada aplikasi.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 5.0 (Medium)

“AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:N/A:N”

```

Edited request
Pretty Raw Hex
1 POST /auth/SignIn.ashx HTTP/1.1
2 user-agent: Dart/2.18 (dart:io)
3 content-type: application/x-www-form-urlencoded;
charset=utf-8
4 Accept-Encoding: gzip, deflate
5 Content-Length: 37
6 host: [REDACTED].com
7 Connection: close
8
9 USER_ID='1 OR 1=1'&USER_PASSWORD='1'

Response
Pretty Raw Hex Render
1 HTTP/1.1 500 Internal Server Error
2 Cache-Control: private
3 Content-Length: 4590
4 Content-Type: text/html; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Sun, 25 Dec 2022 11:22:12 GMT
11 Connection: close
12
13 <!DOCTYPE html>
14 <html>
15   <head>
16     <title>
          Conversion failed when converting the varchar
          value 'admindev' to data type int.
        </title>
      ...
    ...
  ...

```

Gambar 4.60 SQL Injection dengan Mengandalkan Error-Based Respon.

Berdasarkan hasil pengujian (gambar 4.60) dengan melakukan *intercept* terhadap *field* dengan memasukkan *value* yang tidak terduga, pada kasus ini memasukkan nilai “+” (yang sudah dilakukan *decode*) pada *field* userid dan userpassword dapat terlihat adanya *error* yang belum terkontrol pada *remote endpoint*. Hal ini menyebabkan nilai *userid* yang diduga sebagai admin yaitu admindev dapat terlihat pada hasil responnya.

```

Request
Pretty Raw Hex
1 POST /file/StoreImageProfile.ashx
HTTP/1.1
2 user-agent: Dart/2.18 (dart:io)
3 content-type: application/x-www-form-urlencoded;
charset=utf-8
4 Accept-Encoding: gzip, deflate
5 Content-Length: 89
6 host: [REDACTED].com
7 Connection: close
8
9 USER_ID=[REDACTED]&USER_PASSWORD=[REDACTED]&base64=i5&
IMAGE_NAME=scaled_image_picker1917819796.exe

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Cache-Control: private
3 Content-Length: 84
4 Content-Type: application/json; charset=utf-8
5 Server: Microsoft-IIS/8.5
6 X-AspNet-Version: 4.0.30319
7 X-Powered-By: ASP.NET
8 X-Powered-By: ARR/3.0
9 X-Powered-By: ASP.NET
10 Date: Sun, 25 Dec 2022 11:45:29 GMT
11 Connection: close
12
13 [
  {
    "ErrorCode": 0,
    "ErrorMessage": "Success Store Image into Database.",
    "DataValue": [
    ]
  }
]

```

Gambar 4.61 Tamper Request dengan Mengirimkan Data atau File Berekstensi .exe.

Selain itu, pada gambar 4.61 dilakukan pengujian dengan memasukkan *file executable* berbentuk .exe yang dimasukan ke dalam *upload* gambar pada gambar profil pengguna yang memberikan respons sukses tersimpan di *database*.

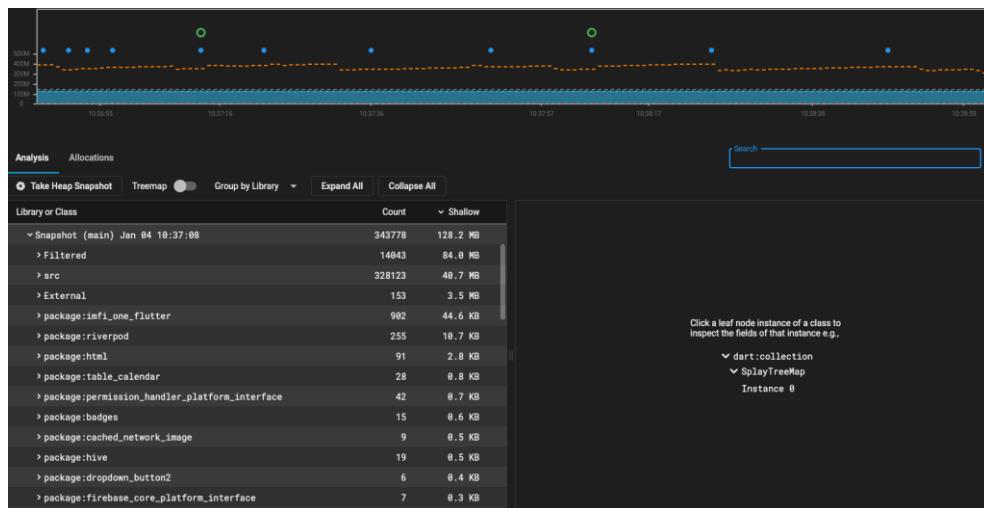
Recommendation: Melakukan sanitasi pada setiap inputan di aplikasi, termasuk dengan *input* file pada fitur *upload*.

4.4.7.8 CODE-8: “*In unmanaged code, memory is allocated, freed*

and used securely.”

Pengujian pada poin ini dilakukan dengan melakukan analisis terhadap *memory profile* pada aplikasi yang sedang berjalan, jika pada grafik *memory* dan isi *snapshot* pada *package* memiliki ukuran yang besar atau abnormal maka dapat dikatakan terjadi memori leak. Pengujian dilakukan menggunakan aplikasi VSCode dalam fitur *memory page* ketika menjalankan aplikasi pada *device*.

STATUS: “PASS”



Gambar 4.62 Memory Page pada VS Code Terhadap Aplikasi yang Sedang Berjalan.

Setelah melakukan pengecekan pada *memory page* sambil menjalankan aplikasi, dapat terlihat bahwa tidak ditemukan adanya *memory leak* pada aplikasi. Hal ini dibuktikan dari bentuk grafik yang stabil rendah dan heap snapshot pada setiap *package* yang memiliki ukuran yang normal.

4.4.7.9 CODE-9: “Free security features offered by the toolchain, such as byte-code minification, stack protection, PIE support and automatic reference counting, are activated.”

Pada poin ini ada beberapa contoh fitur keamanan dari *toolchain* yang harus dipenuhi, seperti:

1. *PIE (Position Independent Executable)*

Adalah *executable binary* yang dibuat dari PIC. PIC sendiri adalah *code* yang ditempatkan di *primary memory* yang dijalankan pada setiap atau beberapa program yang tidak bertumpukan dengan *memory* lain yang sedang digunakan, contohnya dengan *shared library* atau *shared object* lainnya.

2. *Stack Smashing Protection*

Stack smashing protection atau *stack canaries* digunakan untuk mencegah adanya serangan *stack buffer overflow* yang dilakukan dengan cara memasukkan nilai integer tersembunyi pada *stack*. Jika *code* memiliki mekanisme *stack smashing protection*, nilai yang telah *di-tamper* akan ditimpas dan muncul respons bahwa terjadi *tamper* pada *memory*.

Pengujian dilakukan dengan melakukan pengecekan terhadap keberadaan mekanisme perlindungan *binary* pada *code*.

STATUS: “PASS”

```
defaultConfig {
    applicationId "com.ik. ...._aps"
    // You can update the following values to match your application needs.
    // For more information, see: https://docs.flutter.dev/deployment/android#reviewing-the-build-configuration
    minSdkVersion 21
    targetSdkVersion 32
    versionCode flutterVersionCode.toInt()
    versionName flutterVersionName
    multiDexEnabled true
}
```

Gambar 4.63 Isi File Config untuk Mengecek Nilai Minimal SDK Device.

Berdasarkan hasil analisis ditemukan bahwa aplikasi menggunakan ketentuan minsdkversion 21 yang berarti aplikasi sudah memenuhi ketentuan PIE. Hal ini didasarkan dengan Android 5.0 (API level 21) yang sudah memberhentikan penggunaan *native library* yang tidak mendukung PIE. Sedangkan berdasarkan pada gambar subbab code 6 ditemukan bahwa aplikasi menggunakan *framework* flutter dengan Bahasa dart yang tidak melakukan *compile* menggunakan *stack canary*. Hal ini berarti Bahasa dart dapat mencegah *buffer overflow*.



```
Mobexler@Mobexler: ~/Desktop/app-release/lib/arm64-v8a$ rabin2 -I libflutter.so | grep -E "canary|pic"
ERROR: Cannot determine entrypoint, using 0x00316e40
canary false
pic true
```

Gambar 4.64 Mengecek Nilai Protection Canary dan Pic dengan r2 (radare2).

Lalu untuk lebih memastikan, penguji menggunakan *tools rabin2* dalam pengecekan *native library* untuk menentukan bahwa aplikasi telah memiliki *protection* PIE dan *stack smashing* yang di-set sebagai *true*. Berdasarkan gambar 4.64, dapat terlihat bahwa nilai PIC telah di-set *true* yang berarti aplikasi sudah menggunakan perlindungan pada PIE. Namun, pada *canary* memiliki nilai *false* karena pada dasarnya aplikasi yang menggunakan *framework* flutter dan Bahasa dart tidak melakukan *compile* menggunakan *stack canary*.

4.4.8 MASTG-RESILIENCE: Resilience Requirements

4.4.8.1 RESILIENCE-1: “The app detects, and responds to, the presence of a rooted or jailbroken device either by alerting the user or terminating the app.”

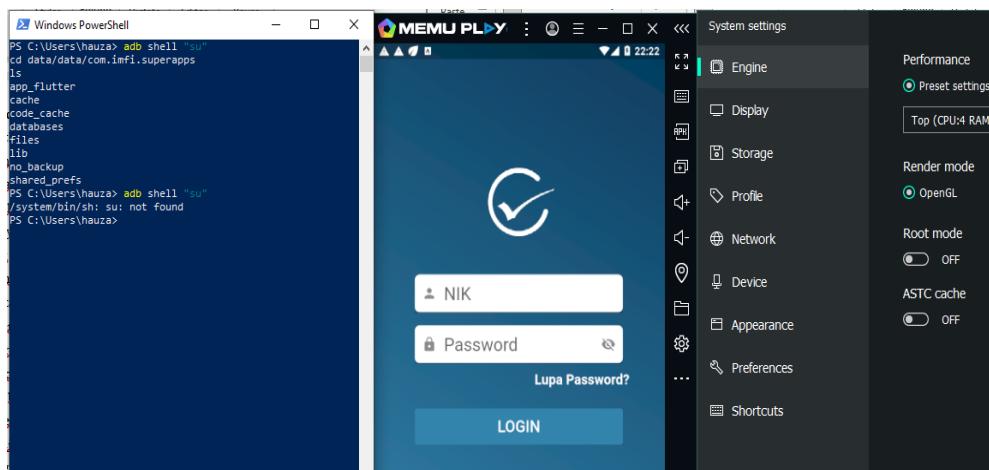
Pengujian pada poin ini dilakukan dengan menjalankan aplikasi pada perangkat yang sudah dilakukan root untuk mengetahui apakah aplikasi memberikan respons atau tindakan deteksi.

STATUS: “FAIL”

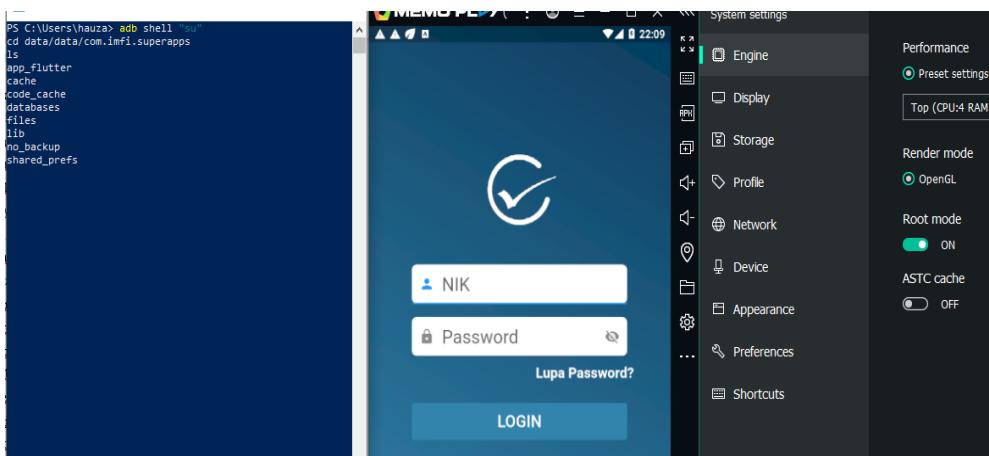
CVSS VECTOR STRING:

Score: 4.5 (Medium)

“AV:L/AC:H/PR:N/UI:N/S:C/C:L/I:L/A:N”



Gambar 4.65 Keadaan Aplikasi Saat Status Device Tidak Root.



Gambar 4.66 Keadaan Aplikasi Saat Status Device Root.

Berdasarkan gambar 4.66 aplikasi dijalankan pada sebuah perangkat emulator yang telah di root. Status root pada perangkat dapat terlihat pada terminal yang telah menjalankan *command* untuk root yaitu ”su” untuk membuat adb shell. Sedangkan pada gambar 4.65 perangkat yang memiliki status bukan root tidak bisa menjalankan *command* adb shell “su”. Kesimpulannya aplikasi tetap berjalan seperti biasa pada kedua kondisi dan tidak memberikan *alert* maupun mekanisme exit paksa.

Recommendation: Menggunakan *library* pada penulisan *code* terhadap Android dan iOS yang digunakan untuk mendeteksi apakah aplikasi sudah di-root. Selain itu, aplikasi dapat menerapkan *binding device* pada *server-side* untuk memastikan aplikasi hanya mengizinkan *device* yang memiliki ijin untuk mengakses aplikasi.

4.4.8.2 RESILIENCE-2: “*The app prevents debugging and/or detects, and responds to, a debugger being attached. All available debugging protocols must be covered.*”

Pada poin ini, pengujian dilakukan dengan menjalankan *command adb* untuk melihat status *package* apakah dapat dilakukan debug atau tidak.

STATUS: “PASS”

Berdasarkan gambar 4.12 (subbab 4.4.2.3) pada aplikasi yang dijalankan *command run-as*. Dapat diambil kesimpulan bahwa aplikasi dalam keadaan *non-debugable*.

4.4.8.3 RESILIENCE-3: “*The app detects, and responds to, tampering with executable files and critical data within its own sandbox.*”

Pada dasarnya terdapat 2 hal yang berkaitan dengan integritas file yaitu file *storage* yang berarti file-file pada aplikasi yang tersimpan di SD *card* atau *public storage* dan *key-value* yang tersimpan di *SharedPreferences*. Lalu, file *code integrity* yang berarti isi yang tersimpan seperti pada file *AndroidManifest.xml*, *class* file **.dex*, *native libraries* (**.so*).

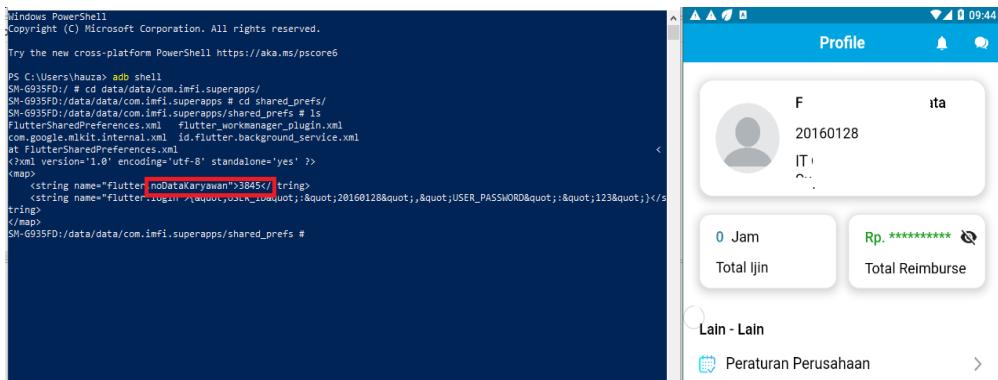
Pengujian dilakukan dengan melakukan *tampering* pada 2 hal yaitu *storage* yang berarti isi pada file seperti *SharedPreference* pada SD *card*, lalu mengubah *permission* serta konten lainnya pada *file* seperti *AndroidManifest.xml*, *classes.dex*, dan file *shared object .so*.

STATUS: “FAIL”

CVSS VECTOR STRING:

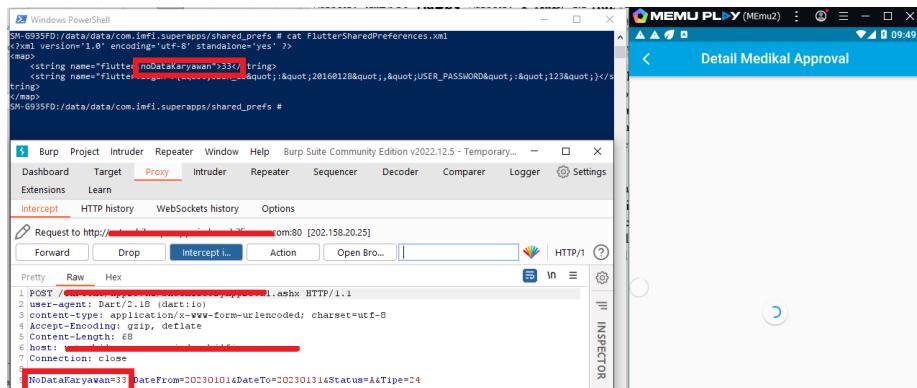
Score: 5.1 (Medium)

“AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N”



Gambar 4.67 Keadaan Aplikasi Sebelum Nilai NoDataKaryawan Pada File SharedPreference Diubah.

Untuk pengujian pada file *storage* yang dilakukan pada file SharedPreferences yang menggunakan nodatakaryawan 3845 yaitu milik *user F*. Lalu, setelah dilakukan pengeditan pada file sharedpreference dilakukan *intercept* untuk melihat apakah nilai nodatakaryawan yang baru diubah telah digunakan untuk proses aplikasi seperti *retrieve* data.



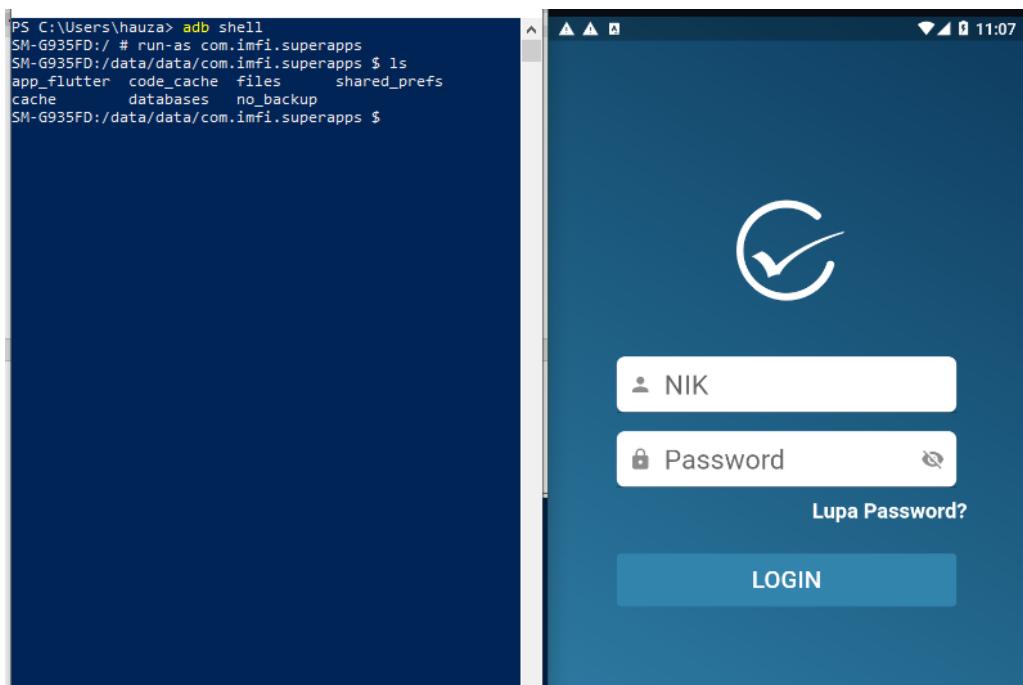
Gambar 4.68 Intercept Request untuk Mengecek Nilai NoDataKaryawan yang Digunakan oleh Aplikasi.

Setelah dicek pada gambar 4.68, dapat terlihat aplikasi menggunakan data nodatakaryawan yang telah diubah.

```
<application android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:icon="@mipmap/launcher_icon" android:label="INFI One" android:name="android.app.Application" android:requestLegacyExternalStorage="true" android:usesCleartextTraffic="true" android:debuggable="true">
```

Gambar 4.69 Tamper Isi File AndroidManifest.xml lalu Rebuild aplikasi.

Lalu, penguji juga menambahkan *value android:debuggable* yang sebelumnya memiliki *value default false* (subbab 4.4.7.2) menjadi *true* untuk melihat apakah *state* aplikasi berubah menjadi dapat dilakukan debug.



Gambar 4.70 Menjalankan Command Run-as Menggunakan Adb Shell.



Gambar 4.71 Nilai Android:debuggable pada AndroidManifest.xml Melalui Reverse Engineering.

Berdasarkan hasil pengecekan, dibuktikan bahwa hasil *tamper* pada file AndroidManifest.xml tetap ada pada file AndroidManifest hasil *reverse engineering* apk dan berefek terhadap *state* aplikasi yang dapat dijalankan *command run-as* yang berarti aplikasi dalam *state debuggable*.

Recommendation: Menerapkan mekanisme deteksi dari terjadinya tamper pada file dan bytecode dengan menambahkan code signature atau melakukan checksum. Checksum dapat dilakukan menggunakan hash function atau checksum library yang tersedia pada Bahasa pemrograman. Checksum yang dihasilkan dapat ditambahkan pada bytecode sebagai string ataupun disimpan terpisah.

4.4.8.4 RESILIENCE-4: “The app detects, and responds to, the presence of widely used reverse engineering tools and frameworks on the device.”

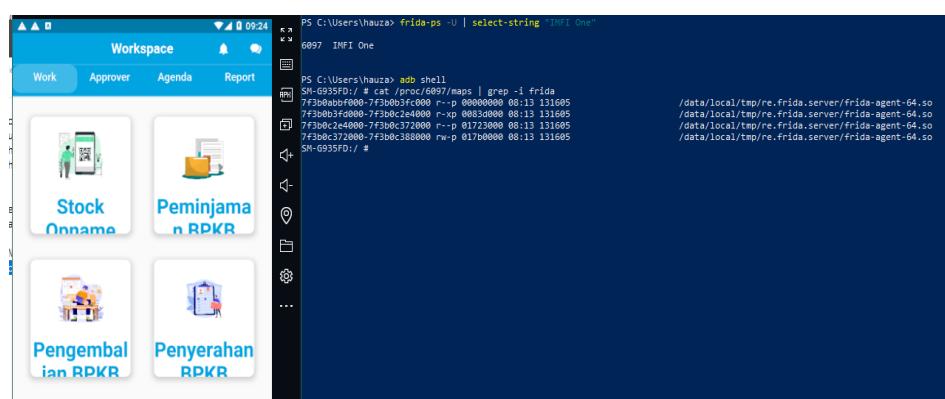
Pengujian pada poin ini dilakukan dengan menjalankan *tools* atau *framework* untuk *reverse engineering* seperti Xposed, Frida, Drozer pada aplikasi. Pada kasus ini, penguji menjalankan Frida-trace pada aplikasi ImfiOne, lalu mengecek *process* yang sedang berjalan pada aplikasi ImfiOne dengan melihat isi file proc/pid/maps pada adb shell.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 5.4 (Medium)

“AV:N/AC:H/PR:N/UI:N/S:C/C:L/I:L/A:N”



Gambar 4.72 Membuka File Mapped Memory pada Process Id IMFI One.

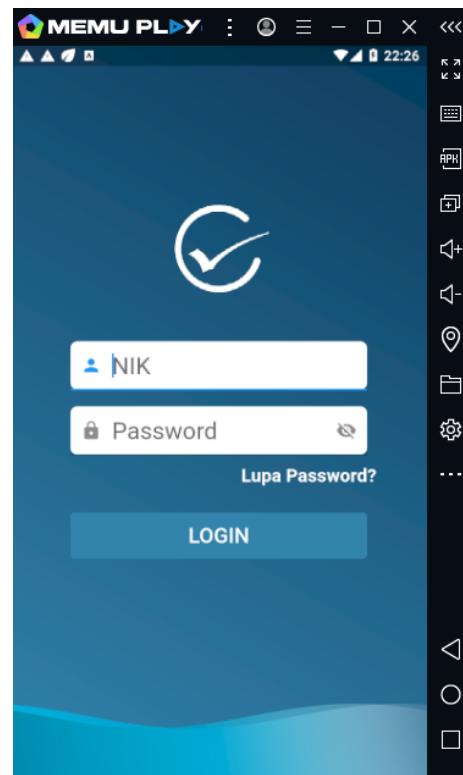
Berdasarkan gambar 4.72, dapat diambil kesimpulan bahwa aplikasi tetap berjalan dengan sempurna walaupun penggunaan alat untuk *reverse engineering* sedang berjalan pada aplikasi (Frida-server) yang dibuktikan dengan adanya data terkait frida-agent pada file proc/[pid-app]/maps.

Recommendation: Menerapkan deteksi pada teknik-teknik *tampering* seperti debugging, pendekripsi root, *code patching*. Terdapat beberapa *library* atau *package* pada setiap Bahasa pemrograman yang dapat digunakan untuk mencegah dan mendekripsi *reverse engineering tool*. Pada flutter terdapat beberapa *library* seperti Obfuscator dan *device info*. Setelah terdeteksi, aplikasi dapat melakukan aksi seperti menonaktifkan fitur tertentu dan mengakhiri proses.

4.4.8.5 RESILIENCE-5: “*The app detects, and responds to, being run in an emulator.*”

Pengujian dilakukan dengan melakukan pemasangan aplikasi pada sebuah emulator untuk menentukan apakah aplikasi dapat berjalan dengan normal tanpa ada *alert* atau deteksi terkait penggunaan emulator.

STATUS: “FAIL”



Gambar 4.73 Menjalankan Aplikasi dengan Emulator.

Berdasarkan gambar 4.73, dapat disimpulkan aplikasi dapat berjalan dengan normal dalam sebuah aplikasi emulator bernama “Memu”.

Recommendation: Menerapkan *device binding* dan melakukan pengecekan pada informasi *device* seperti *device id*, *imei*, dan lain-lain. Aplikasi dapat menggunakan package seperti *DeviceInfo*, *TelephonyManager* pada flutter untuk mengetahui dan membandingkan apakah aplikasi memiliki SIM card, *fingerprint* sensor dan lainnya.

4.4.8.6 RESILIENCE-6: “The app detects, and responds to, tampering the code and data in its own memory space.”

Pengujian dilakukan dengan melakukan injeksi *code* menggunakan *tool* Frida untuk melakukan *hook method* pada setiap *class* yang ada pada aplikasi. Lalu, dilakukan modifikasi pada *method* yang dapat dilakukan dengan cara: 1) melakukan *override* pada *return value* di *method* tersebut. 2) mengubah nilai parameter yang dikirim ke *method*. 3) menambahkan *script* atau menghilangkan *script* yang sudah ada sebelumnya pada *method*.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 5.5 (Medium)

“AV:N/AC:H/PR:L/UI:R/S:C/C:L/I:L/A:L”

```
PS C:\Users\hauza> frida -U -l hook.js -p 3229
[ / \ ] Frida 15.2.2 - A world-class dynamic instrumentation toolkit
| ( ) |
| > - | Commands:
/ / / | help      -> Displays the help system
. . . | object?   -> Display information about 'object'
. . . | exit/quit -> Exit
. . . | More info at https://frida.re/docs/home/
. . . | Connected to SM G935FD (id=127.0.0.1:21523)

[SM G935FD::PID::3229 ]-> message: {'type': 'send', 'payload': 'Inject script berhasil'} data: None
message: {'type': 'send', 'payload': 'Inject script berhasil'} data: None
```

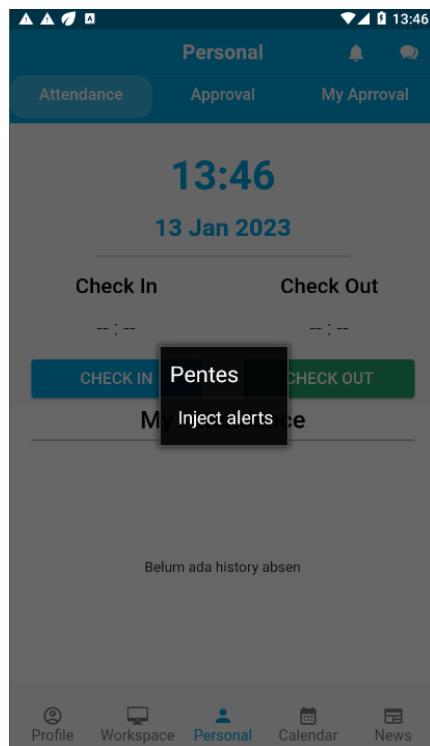
Gambar 4.74 Command Untuk Melakukan Hook Method Dengan Frida.

```
JS hook.js  X

C: > Users > hauza > JS hook.js > ...
● 1 ˇ Java.perform(function () {
  2    const JavaString = Java.use("java.lang.String");
  3    Java.use("android.app.Activity").onUserInteraction.implementation = function () {
  4      var AlertDialogBuilder = Java.use("android.app.AlertDialog$Builder");
  5      send('Inject script berhasil');
  6      var title= JavaString.$new.overload("java.lang.String").call(JavaString, "Pentes");
  7      var text=JavaString.$new.overload("java.lang.String").call(JavaString, "Inject alerts");
  8      var alert = AlertDialogBuilder.$new(this);
  9      alert.setTitle(title);
 10      alert.setMessage(text);
 11      alert.create().show();
 12      return this.onUserInteraction();
 13    };
 14  });


```

Gambar 4.75 Script Untuk Melakukan Hook Method.



Gambar 4.76 Hasil Tes Injeksi Script Dengan Hook Method.

Pengujian yang dilakukan dengan melakukan *hook method* onUserInteraction pada *activity* bawaan android.app.Activity. Berdasarkan hasil pengujian, aplikasi tidak mendeteksi dan memberikan respons khusus ketika dilakukan *tamper code* dengan melakukan *hook method* pada aplikasi yang dapat terlihat pada gambar 4.76.

Recommendation: Menerapkan deteksi pada teknik-teknik *tampering* seperti *debugging*, *pendekteksian root*, *code patching*. Terdapat beberapa *library* atau *package* pada setiap Bahasa pemrograman yang dapat digunakan untuk mencegah dan mendeteksi *reverse engineering tool*.

4.4.8.7 RESILIENCE-7: “*The app implements multiple mechanisms in each defense category (8.1 to 8.6). Note that resiliency scales with the amount, diversity of the originality of the mechanisms used.*”

Pengujian dilakukan dengan menganalisis mekanisme yang digunakan sebagai pertahanan pada poin 4.4.8.1 ke 4.4.8.6.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 5.4 (Medium)

“AV:N/AC:H/PR:N/UI:N/S:C/C:L/I:L/A:N”

Aplikasi ini tidak memiliki mekanisme deteksi dan respons khusus pada pencegahan root, emulator, penggunaan *reverse engineering tool*, dan lainnya.

Recommendation: Menerapkan mekanisme untuk memenuhi poin pada subbab 4.4.8.1 – 4.4.8.6.

4.4.8.8 RESILIENCE-8: “*The detection mechanisms trigger responses of different types, including delayed and stealthy responses.*”

Pada dasarnya terdapat beberapa jenis mekanisme pendeteksian seperti mekanisme respons saat terjadi perbedaan tipe atau data. Hal ini termasuk dengan *delayed* dan *stealthy response*.

- 1) *Delayed Response* adalah teknik untuk menghindari deteksi penyerang dengan membuat aplikasi memberikan respons dengan menunggu beberapa waktu sebelum aplikasi melakukan aksi.
- 2) *Stealthy Response* adalah teknik yang digunakan dengan mengirimkan respons yang tidak terlihat jelas pada penyerang. Respons ini termasuk tindakan seperti mengirimkan log aktivitas mencurigakan, memberikan *notif* atau *alert* kepada personel keamanan, melakukan *disable* pada fitur tertentu tanpa memberitahukan *user*.

STATUS: “FAIL”

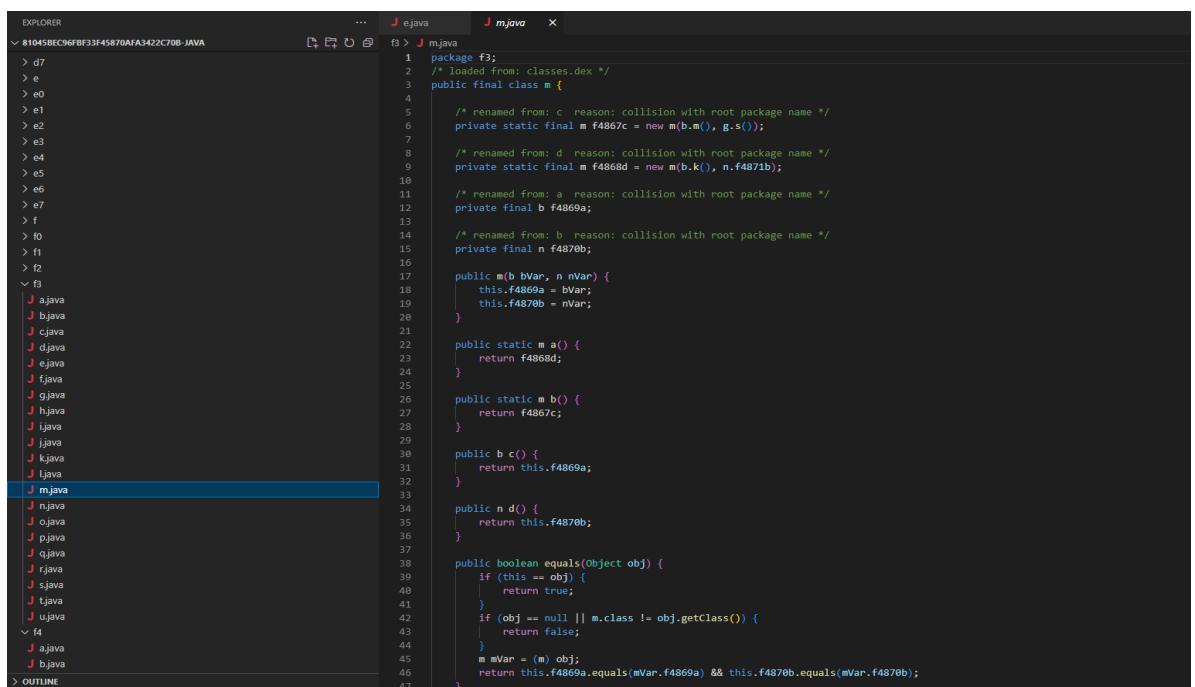
Berdasarkan subbab 4.4.8.3 – 4.4.8.6, aplikasi tidak memiliki mekanisme deteksi seperti pemberian respons saat terjadi *tampering* dari sisi file, *code* dan pada penggunaan root *device* dan emulator.

Recommendation: Setelah diterapkan pendeteksian pada subbab 4.4.8.3 – 4.4.8.6, aplikasi dapat menerapkan mekanisme yang berupa respons untuk mencegah terjadinya penyerangan. Respons yang diberikan dapat berupa *delayed* respons, *stealthy* respons dan lainnya.

4.4.8.9 RESILIENCE-9: “*Obfuscation is applied to programmatic defenses, which in turn impede de-obfuscation via dynamic analysis.*”

Obfuscation adalah proses untuk membuat aplikasi menjadi lebih sulit untuk dipahami oleh manusia dengan mentransformasi *source code* menjadi *string* acak dan *redundant logic*. Penggunaan *obfuscation* pada aplikasi berguna untuk mencegah penyerang dari *reverse engineering* untuk melindungi data atau informasi pada aplikasi dan *source codenya*.

STATUS: “PASS”



```

EXPLORER ... J e.java J m.java ×
↳ 81045BEC96FBF33F45870AFA3422C70B-JAVA f3 > J m.java
> d7
> e
> e0
> e1
> e2
> e3
> e4
> e5
> e6
> e7
> f
> f0
> f1
> f2
< f3
J ajava
J bjava
J cjava
J djava
J ejava
J fjava
J gjava
J hjava
J ijava
J jjava
J kjava
J ljava
J mjava
J njava
J ojava
J pjava
J qjava
J rjava
J sjava
J tjava
J ujava
< f4
J ajava
J bjava
> OUTLINE

```

```

1 package f3;
2 /* loaded from: classes.dex */
3 public final class m {
4
5     /* renamed from: c reason: collision with root package name */
6     private static final m f4867c = new m(b.m(), g.s());
7
8     /* renamed from: d reason: collision with root package name */
9     private static final m f4868d = new m(b.k(), n.f4871b);
10
11    /* renamed from: a reason: collision with root package name */
12    private final b f4869a;
13
14    /* renamed from: b reason: collision with root package name */
15    private final n f4870b;
16
17    public m(b bVar, n nVar) {
18        this.f4869a = bVar;
19        this.f4870b = nVar;
20    }
21
22    public static m a() {
23        return f4868d;
24    }
25
26    public static m b() {
27        return f4867c;
28    }
29
30    public b c() {
31        return this.f4869a;
32    }
33
34    public n d() {
35        return this.f4870b;
36    }
37
38    public boolean equals(Object obj) {
39        if (this == obj) {
40            return true;
41        }
42        if (obj == null || m.class != obj.getClass()) {
43            return false;
44        }
45        m mVar = (m) obj;
46        return this.f4869a.equals(mVar.f4869a) && this.f4870b.equals(mVar.f4870b);
47    }
}

```

Gambar 4.77 Source Code Java Hasil Reverse Engineering.

Pada dasarnya, hasil *compile* pada *code* dart telah melalui proses abstraksi pada setiap *code*-nya dengan menambahkan operasi kode menjadi lebih banyak, mengganti penggunaan *variable* dan nama *function* dengan nama acak serta menggunakan banyak *pointer* pada penggunaan variabel atau *function*-nya.

4.4.8.10 RESILIENCE-10: “*The app implements a ‘device binding’ functionality using a device fingerprint derived from multiple properties unique to the device.*”

Device binding adalah metode yang digunakan untuk mengikat sebuah aplikasi dan *statenya* pada satu *device* khusus. Hal ini bertujuan untuk menghindari tindakan serangan yang dilakukan dengan menyalin sebuah data aplikasi (seperti *local storage* dan *cache*) ke perangkat lainnya. Untuk memenuhi poin ini, aplikasi dapat menerapkan mekanisme yang dapat mengikat aplikasi dengan *device* milik *user* menggunakan nilai yang unik seperti IMEI, *device id*, dan lainnya.

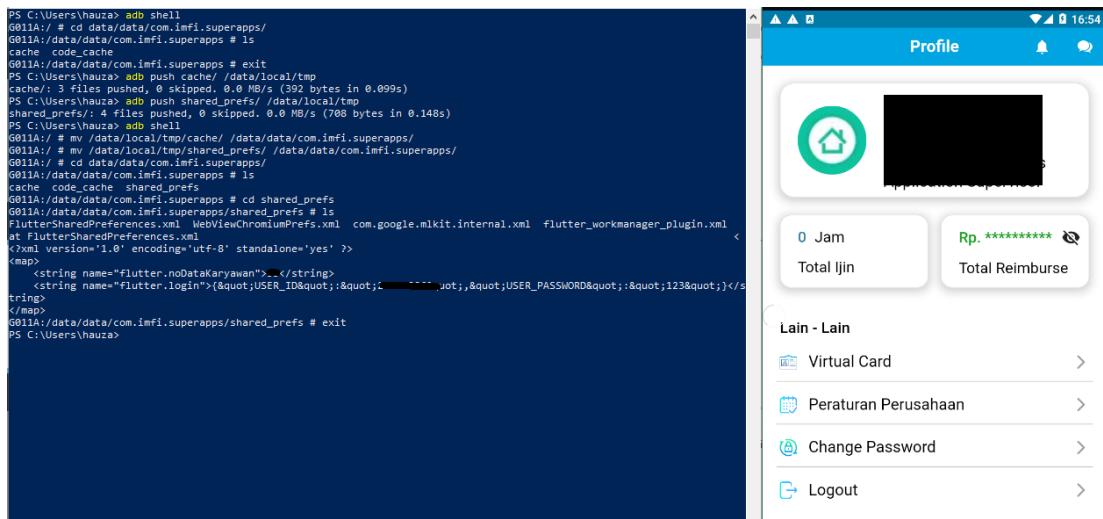
Pengujian dilakukan dengan menyalin isi dari file *cache* dan *SharedPreferences* ke dalam *SD card* yang kemudian dipindahkan ke perangkat lainnya. Jika perangkat lain tersebut melanjutkan *state* berdasarkan data yang tersalin maka pengujian dapat dianggap sebagai “fail”.

STATUS: “PASS”

```
SM-G935FD:/ # cd data/data/com.imfi.superapps
SM-G935FD:/data/data/com.imfi.superapps # ls
app_flutter app_textures app_webview cache code_cache databases files no_backup shared_prefs
SM-G935FD:/data/data/com.imfi.superapps # exit
PS C:\Users\hauza> adb devices
List of devices attached
127.0.0.1:21523 device

PS C:\Users\hauza> adb pull /data/data/com.imfi.superapps/cache/ cache
/data/data/com.imfi.superapps/cache/: 3 files pulled, 0 skipped. 0.0 MB/s (392 bytes in 0.415s)
PS C:\Users\hauza> adb pull /data/data/com.imfi.superapps/shared_prefs/ shared_prefs
/data/data/com.imfi.superapps/shared_prefs/: 4 files pulled, 0 skipped. 0.0 MB/s (708 bytes in 0.270s)
```

Gambar 4.78 Menyalin Data Cache dan SharedPreferences.



Gambar 4.79 Memindahkan File Cache dan SharedPreference ke Emulator Device Baru.

Berdasarkan gambar 4.76 dapat terlihat bahwa aplikasi yang dijalankan pada perangkat baru dan masih melanjutkan *state* dari perangkat sebelumnya yang memiliki informasi *login* pada file shared-preferences.

4.4.8.11 RESILIENCE-11: “*All executable files and libraries belonging to the app are either encrypted on the file level and/or important code and data segments inside the executables are encrypted or packed. Trivial static analysis does not reveal important code or data.”*

Pengujian dilakukan dengan melakukan *reverse engineering* pada file apk. Penguji lalu mengecek apakah *library-library* dan *executable* file sudah melalui mekanisme enkripsi pada *source code* hasil *reverse engineering*.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 2.2 (Low)

“AV:L/AC:H/PR:L/UI:R/S:U/C:L/I:N/A:N”

```

EXPLORE   ...  J ImagePickerPlugin.java
< /Flutter/plugins/imagepicker > J ImagePickerPlugin.java
> i3    275
> i4    276
> i5    277
> i6    278
> i7    279
> id    280
> io.flutter 281
> embedding 282
> plugin 283
> firebase 284
> googlemaps 285
> imagepicker 286
> urlauncher 287
> webviewflutter 288
> < /Flutter/plugins/imagepicker>
> view 289
> j0 290
> j1 291
> j2 292
> j3 293
> j4 294
> j5 295
> j6 296
> j7 297
> k 298
> k0 299
> k1 300
> k2 301
> k3 302
> k4 303
> k5 304
> k6 305
> k7 306
> k8 307
> k9 308
> l0 309
> l1 310
> l2 311
> l3 312
> l4 313
final E b(Activity activity) {
    D dVar = new D(activity);
    File cacheDir = activity.getCacheDir();
    return new E(activity, cacheDir, new G(cacheDir, new Io.flutter.plugins.imagepicker.B()), dVar);
}

@Override // v5.0
public void onAttachedToActivity(C cVar) {
    C(this.f6152f.b(), (Application) this.f6152f.a(), cVar.d(), null, cVar);
}

@Override // v5.0
public void onAttachedToEngine(D dVar) {
    this.f6152f = dVar;
}

@Override // v5.0
public void onDetachedFromActivity() {
    d();
}

@Override // v5.0
public void onDetachedFromActivityForConfigChanges() {
    onDetachedFromActivity();
}

@Override // v5.0
public void onDetachedFromEngine(D dVar) {
    this.f6152f = null;
}

@Override // f6.1.c
public void onMethodCall(K kVar, L dVar) {
    A aVar = this.f6152f;
    if (aVar == null || aVar.a() == null) {
        dVar.b("no_activity", "image_picker plugin requires a foreground activity.", null);
        return;
    }
}

```

Gambar 4.80 File Library dari Source Code Hasil Reverse Engineering.

Berdasarkan gambar 4.77, aplikasi terlihat tidak melalui proses enkripsi sama sekali pada setiap *code* dan data yang berada pada setiap file *library* seperti *google maps*, *image picker*, dan lainnya.

Recommendation: Menerapkan enkripsi pada executable file dan library-library yang digunakan pada aplikasi.

4.4.8.12 RESILIENCE-12: “*If the goal of obfuscation is to protect sensitive computations, an obfuscation scheme is used that is both appropriate for the particular task and robust against manual and automated de-obfuscation methods, considering currently published research. The effectiveness of the obfuscation scheme must be verified through manual testing. Note that hardware-based isolation features are preferred over obfuscation whenever possible.”*

Pengujian dilakukan dengan mengecek file *source code* berbentuk java untuk mengecek apakah skema yang digunakan pada *obfuscation* telah memenuhi syarat agar tidak mudah dilakukan tindakan *deobfuscation*.

STATUS: “PASS”

Terlihat pada gambar 4.79 (subbab 4.4.8.9), bahwa *source code* sudah menerapkan mekanisme *obfuscation* dengan melakukan abstraksi seperti menyembunyikan nama file, *function* dan variabel, menggunakan penerapan logika lebih banyak dengan menambahkan *pointer* dan parameter. Selain itu aplikasi berbahasa .dart dan *framework* flutter telah melalui proses *compile* menjadi .java yang pada dasarnya sudah sulit untuk dibaca.

4.4.8.13 RESILIENCE-13: “As a defense in depth, next to having solid hardening of the communicating parties, application level payload encryption can be applied to further impede eavesdropping.”

Pengujian pada poin ini, dilakukan dengan melakukan *intercept* pada *traffic* jaringan untuk melihat *payload* data yang terkirim. Hal ini berguna untuk menghindari adanya tindakan memata-matai atau *tampering* pada *payload* data yang terkirim pada komunikasi seperti komunikasi *client* dan server pada *remote endpoint*.

STATUS: “FAIL”

CVSS VECTOR STRING:

Score: 2.6 (Low)

“AV:N/AC:H/PR:L/UI:R/S:U/C:L/I:N/A:N”

Berdasarkan gambar 4.41 (subbab 4.4.6.7), terbukti bahwa tidak ada penggunaan enkripsi pada *payload* atau data yang terkirim antara *client* dan server yang menyebabkan data baik itu informasi, *script*, dan lainnya dapat dimata-matai serta ditamper oleh penyerang.

Recommendation: Menggunakan HTTPS sebagai *protocol handler*. HTTPS dapat mengenkripsi komunikasi dari client dan server, hal membuat penyerang lebih sulit untuk melakukan *intercept* dan membaca data. Selain itu Teknik yang dapat digunakan adalah *certificate pinning*. *Certificate pinning* digunakan untuk mengikat *certificate* tertentu kepada *host* tertentu secara khusus.

4.5 Evaluasi Hasil Pengujian

4.5.1 Tabel Status Pengujian

Berdasarkan hasil implementasi pengujian sebagai solusi, dapat disimpulkan hasil pada tiap poinnya yang digambarkan melalui tabel. Tabel tersebut dibagi menjadi 8 bagian berdasarkan kategorinya masing-masing.

Table 4.1 Evaluasi Kategori Architecture, Design, and Threat Modelling

MASTG-ID	Requirement	Hasil
MASTG-ARCH-1	<i>All app components are identified and known to be needed.</i>	PASS
MASTG-ARCH-2	<i>Security controls are never enforced only on the client side, but on the respective remote endpoints.</i>	FAIL
MASTG-ARCH-3	<i>A high-level architecture for the mobile app and all connected remote services has been defined and security has been addressed in that architecture.</i>	FAIL
MASTG-ARCH-4	<i>Data considered sensitive in the context of the mobile app is clearly identified.</i>	PASS
MASTG-ARCH-5	<i>All app components are defined in terms of the business functions and/or security functions they provide.</i>	FAIL
MASTG-ARCH-6	<i>A threat model for the mobile app and the associated remote services has been produced that identifies potential threats and countermeasures.</i>	FAIL
MASTG-ARCH-7	<i>All security controls have a centralized implementation.</i>	PASS
MASTG-ARCH-8	<i>There is an explicit policy for how cryptographic keys (if any) are managed, and the lifecycle of cryptographic keys is enforced. Ideally, follow a key management standard such as NIST SP 800-57.</i>	N/A
MASTG-ARCH-9	<i>A mechanism for enforcing updates of the mobile app exists.</i>	PASS
MASTG-ARCH-10	<i>Security is addressed within all parts of the software development lifecycle.</i>	FAIL
MASTG-ARCH-11	<i>A responsible disclosure policy is in place and effectively applied.</i>	FAIL
MASTG-ARCH-12	<i>The app should comply with privacy laws and regulations.</i>	PASS

Table 4.2 Evaluasi Kategori Data Storage and Privacy

MASTG-ID	Requirement	Hasil
MASTG-STORAGE-1	<i>System credential storage facilities need to be used to store sensitive data, such as PII, user credentials or cryptographic keys.</i>	FAIL
MASTG-STORAGE-2	<i>No sensitive data should be stored outside of the app container or system credential storage facilities.</i>	PASS
MASTG-STORAGE-3	<i>No sensitive data is written to application logs.</i>	FAIL
MASTG-STORAGE-4	<i>No sensitive data is shared with third parties unless it is a necessary part of the architecture.</i>	PASS
MASTG-STORAGE-5	<i>The keyboard cache is disabled on text inputs that process sensitive data.</i>	PASS
MASTG-STORAGE-6	<i>No sensitive data is exposed via IPC mechanisms.</i>	PASS
MASTG-STORAGE-7	<i>No sensitive data, such as passwords or pins, is exposed through the user interface.</i>	PASS
MASTG-STORAGE-8	<i>No sensitive data is included in backups generated by the mobile operating system.</i>	FAIL
MASTG-STORAGE-9	<i>The app removes sensitive data from views when moved to the background.</i>	PASS
MASTG-STORAGE-10	<i>The app does not hold sensitive data in memory longer than necessary, and memory is cleared explicitly after use.</i>	FAIL
MASTG-STORAGE-11	<i>The app enforces a minimum device-access-security policy, such as requiring the user to set a device passcode.</i>	FAIL
MASTG-STORAGE-12	<i>The app educates the user about the types of personally identifiable information processed, as well as security best practices the user should follow in using the app.</i>	FAIL
MASTG-STORAGE-13	<i>No sensitive data should be stored locally on the mobile device. Instead, data should be retrieved from a remote endpoint when needed and only be kept in memory.</i>	FAIL
MASTG-STORAGE-14	<i>If sensitive data is still required to be stored locally, it should be encrypted using a key derived from hardware backed storage which requires authentication.</i>	FAIL
MASTG-STORAGE-15	<i>The app's local storage should be wiped after an excessive number of failed authentication attempts.</i>	FAIL

Table 4.3 Evaluasi Kategori Cryptography

MASTG-ID	Requirement	Hasil
MASTG-CRYPTO-1	<i>The app does not rely on symmetric cryptography with hardcoded keys as a sole method of encryption.</i>	N/A
MASTG-CRYPTO-2	<i>The app uses proven implementations of cryptographic primitives.</i>	N/A
MASTG-CRYPTO-3	<i>The app uses cryptographic primitives that are appropriate for the particular use-case, configured with parameters that adhere to industry best practices.</i>	N/A
MASTG-CRYPTO-4	<i>The app does not use cryptographic protocols or algorithms that are widely considered deprecated for security purposes.</i>	N/A
MASTG-CRYPTO-5	<i>The app doesn't re-use the same cryptographic key for multiple purposes.</i>	N/A
MASTG-CRYPTO-6	<i>All random values are generated using a sufficiently secure random number generator.</i>	N/A

Table 4.4 Evaluasi Kategori Authentication and Session Management

MASTG-ID	Requirement	Hasil
MASTG-AUTH-1	<i>If the app provides users access to a remote service, some form of authentication, such as username/password authentication, is performed at the remote endpoint.</i>	PASS
MASTG-AUTH-2	<i>If stateful session management is used, the remote endpoint uses randomly generated session identifiers to authenticate client requests without sending the user's credentials.</i>	N/A
MASTG-AUTH-3	<i>If stateless token-based authentication is used, the server provides a token that has been signed using a secure algorithm.</i>	N/A
MASTG-AUTH-4	<i>The remote endpoint terminates the existing session when the user logs out.</i>	N/A
MASTG-AUTH-5	<i>A password policy exists and is enforced at the remote endpoint.</i>	FAIL
MASTG-AUTH-6	<i>The remote endpoint implements a mechanism to protect against the submission of credentials an excessive number of times.</i>	FAIL
MASTG-AUTH-7	<i>Sessions are invalidated at the remote endpoint after a predefined period of inactivity and access tokens expire.</i>	N/A
MASTG-AUTH-8	<i>Biometric authentication, if any, is not event-bound (i.e. using an API that simply returns "true" or "false"). Instead, it is based on unlocking the keychain/keystore.</i>	N/A
MASTG-AUTH-9	<i>A second factor of authentication exists at the remote endpoint and the 2FA requirement is consistently enforced.</i>	FAIL
MASTG-AUTH-10	<i>Sensitive transactions require step-up authentication.</i>	FAIL
MASTG-AUTH-11	<i>The app informs the user of all sensitive activities with their account. Users are able to view a list of devices, view contextual information (IP address, location, etc.), and to block specific devices.</i>	FAIL
MASTG-AUTH-12	<i>Authorization models should be defined and enforced at the remote endpoint.</i>	PASS

Table 4.5 Evaluasi Kategori Network Communication

MASTG-ID	Requirement	Hasil
MASTG-NETWORK-1	<i>Data is encrypted on the network using TLS. The secure channel is used consistently throughout the app.</i>	FAIL
MASTG-NETWORK-2	<i>The TLS settings are in line with current best practices, or as close as possible if the mobile operating system does not support the recommended standards.</i>	FAIL
MASTG-NETWORK-3	<i>The app verifies the X.509 certificate of the remote endpoint when the secure channel is established. Only certificates signed by a trusted CA are accepted.</i>	FAIL
MASTG-NETWORK-4	<i>The app either uses its own certificate store, or pins the endpoint certificate or public key, and subsequently does not establish connections with endpoints that offer a different certificate or key, even if signed by a trusted CA.</i>	FAIL
MASTG-NETWORK-5	<i>The app doesn't rely on a single insecure communication channel (email or SMS) for critical operations, such as enrollments and account recovery.</i>	FAIL
MASTG-NETWORK-6	<i>The app only depends on up-to-date connectivity and security libraries.</i>	FAIL

Table 4.6 Evaluasi Kategori Platform Interaction

MASTG-ID	Requirement	Hasil
MASTG-PLATFORM-1	<i>The app only requests the minimum set of permissions necessary.</i>	PASS
MASTG-PLATFORM-2	<i>All inputs from external sources and the user are validated and if necessary sanitized. This includes data received via the UI, IPC mechanisms such as intents, custom URLs, and network sources.</i>	FAIL
MASTG-PLATFORM-3	<i>The app does not export sensitive functionality via custom URL schemes, unless these mechanisms are properly protected.</i>	N/A
MASTG-PLATFORM-4	<i>The app does not export sensitive functionality through IPC facilities, unless these mechanisms are properly protected.</i>	PASS
MASTG-PLATFORM-5	<i>JavaScript is disabled in WebViews unless explicitly required.</i>	PASS
MASTG-PLATFORM-6	<i>WebViews are configured to allow only the minimum set of protocol handlers required (ideally, only https is supported). Potentially dangerous handlers, such as file, tel and app-id, are disabled.</i>	FAIL
MASTG-PLATFORM-7	<i>If native methods of the app are exposed to a WebView, verify that the WebView only renders JavaScript contained within the app package.</i>	FAIL
MASTG-PLATFORM-8	<i>Object deserialization, if any, is implemented using safe serialization APIs.</i>	PASS
MASTG-PLATFORM-9	<i>The app protects itself against screen overlay attacks. (Android only)</i>	PASS
MASTG-PLATFORM-10	<i>A WebView's cache, storage, and loaded resources (JavaScript, etc.) should be cleared before the WebView is destroyed.</i>	FAIL
MASTG-PLATFORM-11	<i>Verify that the app prevents usage of custom third-party keyboards whenever sensitive data is entered (iOS only).</i>	FAIL

Table 4.7 Code Quality and Build Setting

MASTG-ID	Requirement	Hasil
MASTG-CODE-1	<i>The app is signed and provisioned with a valid certificate, of which the private key is properly protected.</i>	PASS
MASTG-CODE-2	<i>The app has been built in release mode, with settings appropriate for a release build (e.g. non-debuggable).</i>	PASS
MASTG-CODE-3	<i>Debugging symbols have been removed from native binaries.</i>	PASS
MASTG-CODE-4	<i>Debugging code and developer assistance code (e.g. test code, backdoors, hidden settings) have been removed. The app does not log verbose errors or debugging messages.</i>	FAIL
MASTG-CODE-5	<i>All third party components used by the mobile app, such as libraries and frameworks, are identified, and checked for known vulnerabilities.</i>	PASS
MASTG-CODE-6	<i>The app catches and handles possible exceptions.</i>	FAIL
MASTG-CODE-7	<i>Error handling logic in security controls denies access by default.</i>	FAIL
MASTG-CODE-8	<i>In unmanaged code, memory is allocated, freed and used securely.</i>	PASS
MASTG-CODE-9	<i>Free security features offered by the toolchain, such as byte-code minification, stack protection, PIE support and automatic reference counting, are activated.</i>	PASS

Table 4.8 Evaluasi Kategori Resilience

MASTG-ID	Requirement	Hasil
MASTG-RESILIENCE-1	<i>The app detects, and responds to, the presence of a rooted or jailbroken device either by alerting the user or terminating the app.</i>	FAIL
MASTG-RESILIENCE-2	<i>The app prevents debugging and/or detects, and responds to, a debugger being attached. All available debugging protocols must be covered.</i>	PASS
MASTG-RESILIENCE-3	<i>The app detects, and responds to, tampering with executable files and critical data within its own sandbox.</i>	FAIL
MASTG-RESILIENCE-4	<i>The app detects, and responds to, the presence of widely used reverse engineering tools and frameworks on the device.</i>	FAIL
MASTG-RESILIENCE-5	<i>The app detects, and responds to, being run in an emulator.</i>	FAIL
MASTG-RESILIENCE-6	<i>The app detects, and responds to, tampering the code and data in its own memory space.</i>	FAIL
MASTG-RESILIENCE-7	<i>The app implements multiple mechanisms in each defense category (8.1 to 8.6). Note that resiliency scales with the amount, diversity of the originality of the mechanisms used.</i>	FAIL
MASTG-RESILIENCE-8	<i>The detection mechanisms trigger responses of different types, including delayed and stealthy responses.</i>	FAIL
MASTG-RESILIENCE-9	<i>Obfuscation is applied to programmatic defenses, which in turn impede de-obfuscation via dynamic analysis.</i>	PASS
MASTG-RESILIENCE-10	<i>The app implements a 'device binding' functionality using a device fingerprint derived from multiple properties unique to the device.</i>	PASS
MASTG-RESILIENCE-11	<i>All executable files and libraries belonging to the app are either encrypted on the file level and/or important code and data segments inside the executables are encrypted or packed. Trivial static analysis does not reveal important code or data.</i>	FAIL
MASTG-RESILIENCE-12	<i>If the goal of obfuscation is to protect sensitive computations, an obfuscation scheme is used that is both appropriate for the particular task and robust against manual and automated de-obfuscation methods, considering currently published research. The effectiveness of the obfuscation scheme must be verified through manual testing. Note that hardware-based isolation features are preferred over obfuscation whenever possible.</i>	PASS
MASTG-RESILIENCE-13	<i>As a defense in depth, next to having solid hardening of the communicating parties, application level payload encryption can be applied to further impede eavesdropping.</i>	FAIL

4.5.2 Ringkasan Hasil Pengujian

Ringkasan ditentukan berdasarkan hasil perhitungan dengan membandingkan nilai kelulusan poin pada setiap kategori. Perhitungan didasarkan pada perbandingan poin yang berstatus PASS dan keseluruhan poin (kecuali status N/A) pada setiap kategori.

Table 4.9 Ringkasan Hasil Pengujian

Android	P	F	N/A	%
V1: Architecture, Design and Threat Modelling	5	6	1	45.45%
V2: Data Storage and Privacy	6	9	0	40.00%
V3: Cryptography Verification	0	0	6	-
V4: Authentication and Session Management	2	5	5	28.57%
V5: Network Communication	0	6	0	0.00%
V6: Platform Intercation	5	5	1	50.00%
V7: Code Quality and Build Settings	6	3	0	66.67%
V8: Resiliency Against Reverse Engineering	4	9	0	30.77%

Berdasarkan tabel ringkasan tersebut, dapat dihitung *compliance score* atau nilai kepatuhan dari hasil pengujian berdasarkan panduan OWASP MASVS sebagai berikut:

$$(45.45 + 40 + 0 + 28.57 + 0 + 50 + 66.67 + 30.77) / 8 = 32.68$$

Compliance Score = 32.68/100

4.5.3 Evaluasi Dengan Wawancara Terhadap Hasil Pengujian

Berikut merupakan wawancara yang dilakukan dengan perwakilan tim IT Infrastructure & Development dari PT Indomobil Finance Indonesia yang dilakukan pada tanggal 17 Januari 2023. Wawancara ini terdiri dari 7 pertanyaan, berikut merupakan notulen dari hasil wawancara:

1. Siapa nama dan jabatan anda di perusahaan PT. Indomobil Finance Indonesia?

Jawaban: Devin Williady, IT Core Business Application Staff

2. Apakah tujuan penggerjaan skripsi sesuai dengan kebutuhan perusahaan?

Jawaban: Ya Sesuai, dikarenakan untuk saat ini di perusahaan belum ada *staff* IT berkaitan dengan *cyber security*. Tujuan penggerjaan skripsi anda sesuai dengan kebutuhan untuk peningkatan keamanan aplikasi aplikasi yang sedang dibuat dan dikembangkan saat ini.

3. Apakah pengujian yang dilakukan sudah memenuhi kebijakan dari segi cakupan dan batasan-batasan dilakukannya pengujian?

Jawaban: Berdasarkan hasil yang dijabarkan dan bukti-bukti yang diberikan, pengujian yang dilakukan berkaitan dengan *security* sebuah aplikasi sudah dalam batasan dan cakupan pengujian yang cukup sesuai dengan kebijakan perusahaan.

4. Apakah hasil penulisan dari pengujian keamanan dapat dimengerti dengan baik?

Jawaban: Dari hasil yang kamu berikan kemarin dan yang sudah saya dan tim baca. untuk penjelasan mengenai apa yang ditest dan hasil rekomendasi yang dijabarkan setiap poinnya dapat dimengerti dengan baik oleh saya dan tim.

5. Apakah bukti yang diberikan relevan terhadap kondisi keamanan aplikasi saat ini?

Jawaban: Berdasarkan hasil yang diberikan dan penjelasan berkaitan dengan kelemahan pada aplikasi menggambarkan kondisi dari keamanan aplikasi saat ini.

6. Apakah rekomendasi atau saran yang diberikan memungkinkan untuk diterapkan pada aplikasi?

Jawaban: Terdapat beberapa rekomendasi yang bisa dikerjakan secara langsung secara *development* langsung diimplementasikan kembali. sebagai contoh untuk rekomendasi *device binding* untuk fitur absensi. Rekomendasi ini bisa memastikan karyawan untuk melakukan absen melalui *device*-nya masing-masing. namun, juga terdapat beberapa poin yang membutuhkan persetujuan dan pembicaraan yang lebih dengan tim it lainnya untuk diterapkan. seperti penggunaan *protocol handler* HTTPS.

7. Menurut bapak, apakah skor dari keamanan aplikasi sesuai dengan tingkat keamanan aplikasi saat ini?

Jawaban: Tingkat keamanan aplikasi saat ini kurang lebih memiliki tingkatan yang sesuai dengan hasil *scoring* yang kamu berikan. Jika dilihat skor tersebut sangat rendah ya, jadi setelah hasil diberikan, sudah beberapa rekomendasi langsung diaplikasikan (tidak hanya pada aplikasi ini saja) dan sebagai ilmu untuk kedepannya dalam pengembangan aplikasi lainnya.

8. Bisakah bapak menjelaskan bagaimana hasil penulisan ini membantu bapak dan perusahaan dalam meningkatkan keamanan aplikasi saat ini!

Jawaban: Hasil penulisan ini sangat membantu perusahaan untuk dapat mengetahui kelemahan-kelemahan apa yang ada pada aplikasi, selain itu saya sebagai salah satu pengembang aplikasi bisa mengetahui tingkat keamanan aplikasi yang saya dan tim buat dan menjadikan penulisan ini sebagai referensi bagi saya dan para pengembang lainnya untuk memperbaiki dan mencegah terjadinya masalah-masalah dari sisi keamanan pada aplikasi.

BAB BERIKUTNYA HARUS PADA HALAMAN GANJIL

BAB 5

KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Berdasarkan hasil pengujian aplikasi IMFI One milik PT. Indomobil Finance Indonesia menggunakan *requirement* dari OWASP MASVS beserta *testing guide* yang ada pada OWASP MASTG, aplikasi memiliki skor keamanan yang kecil dari keseluruhan pengujian. Dari 8 kategori yang ada pada *requirement* OWASP MASVS, aplikasi memiliki nilai kepatuhan 32,68% dari total 84 poin pengujian. Berikut merupakan kesimpulan yang berisi ringkasan keseluruhan poin pada setiap kategori:

1. *Architecture, Design, and Threat Modelling* memiliki persentase pemenuhan sebanyak 45,45%. Pada kategori ini, aplikasi tidak memenuhi kriteria karena tidak memiliki struktur *high-level architecture*, dokumentasi, *threat model*, serta kebijakan yang sesuai. Selain itu, pada kategori ini juga ditemukan poin yang berdampak *critical* yaitu keamanan yang tidak diterapkan pada sisi *endpoint*. Pelaku penyerangan dalam kasus terburuk bisa melakukan drop pada *database* yang menyebabkan kerusakan, kehilangan pada data, serta ketersediaan akses pada sumber daya.
2. *Data Storage and Privacy* memiliki persentase pemenuhan sebanyak 40%. Pada kategori ini, aplikasi tidak memenuhi kriteria dalam penyimpanan data serta pengamanan data dengan enkripsi di *local storage*. Selain itu, aplikasi juga masih menampilkan log dan belum mempunyai kebijakan dalam pengaturan perangkatnya.
3. *Cryptography* memiliki persentase pemenuhan sebanyak 0%. Pada kategori ini pengujian pada setia poinnya tidak bisa dilakukan dan dianggap tidak memenuhi kriteria karena tidak ditemukan adanya penggunaan kriptografi.
4. *Authentication and Session Management* memiliki persentase pemenuhan sebanyak 28,57%. Pada kategori ini, aplikasi ditemukan masih memiliki kelemahan pada *remote endpoint* dari penerapan kebijakan *password* dan pengamanannya atas serangan *bruteforce*. Selain itu, aplikasi juga masih belum memenuhi kriteria dari segi otentikasi tambahan pada fitur yang membutuhkan.
5. *Network Communication* memiliki persentase pemenuhan sebanyak 0%. Hal ini terjadi karena tidak ditemukan adanya penggunaan *protocol handler* HTTPS

yang berguna untuk melakukan enkripsi pada data yang ditransmisikan. Selain itu, aplikasi juga masih menggunakan media yang lemah seperti email untuk melakukan *recovery account*.

6. *Platform Interaction* memiliki persentase sebanyak 50%. Pada aplikasi ditemukan kelemahan yang cukup *critical* yaitu tidak ditemukannya sanitasi input baik dari inputan berupa text maupun inputan pada fitur *upload*. Hal ini berarti aplikasi memiliki kelemahan dengan kasus terburuk yang sama dengan poin 1. Selain itu, aplikasi juga tidak melakukan *rendering javascript* dengan baik dan penyimpanan data seperti cache dan sumber daya lainnya yang tidak aman pada *webview*.
7. *Code Quality and Build Settings* memiliki persentase sebanyak 66.67% yang berarti hampir sebagian poin telah memenuhi kriteria pemenuhan. Pada kategori ini, aplikasi masih memiliki masalah dalam pengurusan *exception* dari respons API dan tampilan data pada log dan debug *code* yang masih dapat terlihat.
8. *Resilience Against Reverse Engineering* memiliki persentase sebanyak 30.77%. Aplikasi tidak melakukan enkripsi pada file untuk mencegah *reverse engineering* dan aplikasi tidak memiliki mekanisme pendekripsi pada penggunaan root, emulator dan *tool reverse engineering*. Selain itu aplikasi juga tidak memiliki mekanisme pencegahan saat terjadi *tampering* pada file dan *code* dalam aplikasi yang berjalan.

5.2 SARAN

Berdasarkan hasil kesimpulan dari pengujian aplikasi IMFI One milik PT. Indomobil Finance Indonesia dengan panduan OWASP MASTG dan MASVS, terdapat beberapa saran bagi PT. Indomobil Finance Indonesia untuk meningkatkan kualitas perusahaan dalam segi penggunaan aplikasi bagi *user*. Berikut merupakan beberapa saran yang diberikan:

- Menerapkan pencegahan dan tindakan remediasi pada poin yang memiliki kategori *"Medium"* dan *"Critical"* terlebih dahulu sebagai prioritas utama untuk mencegah terjadinya kejadian yang tidak diinginkan terhadap aplikasi IMFI One maupun aplikasi milik perusahaan lainnya.
- Saran yang diberikan pada setiap poin pada bab 4 merupakan salah satu metode dari sekian banyak metode lainnya yang dapat digunakan bagi perusahaan dalam mencegah serta mengatasi masalah keamanan pada aplikasi. Perusahaan dapat menerapkan tindakan pencegahan lainnya berdasarkan kebijakan dari perusahaan yang sudah

disesuaikan dengan pertimbangan-pertimbangan lainnya dari sisi bisnis perusahaan.

- Menerapkan tindakan pencegahan paling umum dan sederhana seperti memperkerjakan *staff* dalam segi keamanan aplikasi serta menerapkan *code review* dalam *software development life cycle*.
- Penerapan keamanan harus dilakukan tidak hanya pada sisi pembuatan aplikasi *mobile*, namun juga diperlukan pada sisi *remote endpoint* atau server pada penerapan API.
- Terdapat saran pada poin-poin di bab 4 yang tidak memiliki *score*, hal ini berarti *score* dari kerentanan tersebut adalah 0. Hal ini bukan berarti saran untuk pencegahan yang diberikan tidak penting untuk dilakukan, namun saran tersebut dapat digunakan untuk mengecilkan kemungkinan terjadinya dampak yang tidak diinginkan dan dapat memperkuat keamanan aplikasi secara keseluruhan.

BAB BERIKUTNYA PADA HALAMAN GANJIL

REFERENSI

- Alabduljabbar, R. (2021). Development of a System to Manage Letters of Recommendation. *International Journal of Advanced Computer Science and Applications*, 12(1).
- Alvarez, S. (2006). *radare2*. Diambil kembali dari radare2: <https://rada.re/n/>
- Chebbi, C. (2018). *Advanced Infrastructure Penetration Testing: Defend your systems from methodized and proficient attackers*. Packt Publishing Ltd.
- Coronel, C., & Morris, S. (2016). *Database Systems: Design, Implementation, & Management*. Cengage Learning.
- First, O. (2015). *CVSS v3.0 User Guide*. Diambil kembali dari first.org: <https://www.first.org/cvss/v3.0/user-guide>
- Florackis, C., Louca, C., Michaelly, R., & Weber, M. (2020). CYBERSECURITY RISK. *NBER WORKING PAPER SERIES*.
- Gani, A. G. (2018). Pengenalan Teknologi Internet Serta Dampaknya. *JSI (Jurnal Sistem Informasi) Universitas Suryadama*, 2.
- Gargenta, M. (2011). *Learning Android*. O'Reilly Media, Inc.
- Gilski, P., & Stefanski, J. (2015). Android os: A review. *Tem Journal*, 116.
- Ginta, P. w., Kusum, G. P., & Negara, E. K. (2013). IMPLEMENTASI TOOLS NETWORK MAPPER PADA LOKAL AREA NETWORK (LAN). *JURNAL MEDIA INFOTAMA*.
- Goel, J. N., & Mehtre, B. (2015). Vulnerability Assessment & Penetration Testing as a Cyber Defence Technology. *Procedia Computer Science*.
- ibotpeaches. (2010). *apktool*. Diambil kembali dari apktool: <https://ibotpeaches.github.io/Apktool/>
- ISO. (2022, 10 25). *ISO/IEC 27001 and related standards*. Diambil kembali dari ISO: <https://www.iso.org/isoiec-27001-information-security.html>
- MobSF. (2015). *MobSF*. Diambil kembali dari MobSF Github: <https://github.com/MobSF/Mobile-Security-Framework-MobSF>
- Obotivere, B. A., & Nwaezeigwe, A. O. (2020). Cyber Security Threats on the Internet. *International Journal of Advanced Research in Computer and Communication Engineering*.
- OWASP. (2001). *owasp*. Diambil kembali dari owasp: <https://owasp.org/>
- Pan, Y. (2019). Interactive Application Security Testing. *International Conference on Smart Grid and Electrical Automation (ICSGEA)*.

patrickfav. (2016). *uber-apk-signer*. Diambil kembali dari Uber Signer Github:
<https://github.com/patrickfav/uber-apk-signer>

Pranata, B. A., Hijriani, A., & Junaidi, A. (2018). PERANCANGAN APPLICATION PROGRAMMING INTERFACE (API) BERBASI WEB MENGGUNAKAN GAYA ARSITEKTUR REPRESENTATIONAL STATE TRANSFER (REST) UNTUK PENGEMBANGAN SISTEM INFORMASI ADMINISTRASI PASIEN KLINIK.

Rahalkar, S. (2021). *A Complete Guide to Burp Suite*. Apress.

Ravnas, O. A. (2012). *Frida*. Diambil kembali dari Frida: <https://frida.re/>

Reflutter. (2021). *Reflutter*. Diambil kembali dari Reflutter Github:
<https://github.com/ptswarm/reFlutter>

Regupathy, R. (2014). *Android Debug Bridge (adb) In Unboxing Android USB*. Springer.

Richet, J.-L. (2013). From Young Hackers to Crackers. *International Journal of Technology and Human Interaction*.

S, O. (2022, February 23). *Number of smartphone subscriptions worldwide from 2016 to 2021, with forecasts from 2022 to 2027*. Diambil kembali dari statista:
<https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>

Shah, A., Farris, K. A., & Jajodia, S. (2019). Vulnerability Selection for Remediation: An Empirical Analysis. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*.

Shao, Y., Luo, X., & Qian, C. (2014). RootGuard: Protecting Rooted Android Phones.

Syafrizal, M. (2020). *Pengantar Jaringan Komputer*. Andi.

T.Campbell, J. (2014). *Discovering The Internet*. Cengage Learning.

Tashildar, A., Shah, N., Gala, R., Giri, T., & Chavhan, P. (2020). APPLICATION DEVELOPMENT USING FLUTTER. *International Research Journal of Modernization in Engineering, Technology and Science*.

Whyte, J. (2019). Smartphone.

Zhigalov, K., & Ivanov, V. (2019). Reverse Engineering of Mobile Applications. *IOP Conference Series: Materials Science and Engineering*.

RIWAYAT HIDUP

Riwayat Hidup / Curriculum Vitae



PERSONAL INFORMATION

Binusian ID	: 2301926245
Full Name	: Dhia Hauzan Muafa
E-Mail	: hauzanm8@gmail.com / dhia.muafa@binus.ac.id
Address	: Current Perum Permata Taman Wanasari Indah Blok B No 9, Cibitung, Bekasi, 17520
	Permanent Perum Permata Taman Wanasari Indah Blok B No 9, Cibitung, Bekasi, 17520
Phone Numbers	: 0895389539960
Gender	: Male
Birth Place/Date	: Bekasi / 13 Juni 2001
Nationality	: Indonesian
Marital Status	: Single
Religion	: Islam

FORMAL EDUCATION

Elementary School	2007 – 2013	SDIT Daarussalam
Junior High School	2013 – 2016	SMP Al-Muhadjirin
Senior High School	2016 - 2018	SMAN 2 Tambun Selatan
Bachelor's degree	2019 – 2022 (Ongoing)	Universitas Bina Nusantara Cyber Security GPA: 3.65 (Current)

INFORMAL EDUCATION

2018 - 2019	Global English Course
-------------	-----------------------

PERSONAL CERTIFICATION

-

ORGANIZATION EXPERIENCE

2019 - 2021	CSC (Cyber Security Community)
2021	Proctor Final Contest ICPC
2022	Freshmen Leader

WORKING EXPERIENCE

2022 - 2023	Internship at PT. Indomobil Finance Indonesia Position: Full Stack Developer
-------------	--

Job Description:

- Membangun / mengembangkan *software* dengan menggunakan bahan pemrograman yang sudah ditentukan.
- Aplikasi yang dibuat berupa *web base application* dan *mobile application*
- Mengimplementasikan *requirement* dan *design* proses bisnis ke komputer dengan menggunakan algoritma/ logika dan bahasa pemrograman.
- Melakukan Testing terhadap *software* dan *security*-nya bila diperlukan.