APLIKASI *HYBRID* STEGANOGRAFI EOF DAN ENKRIPSI AES-128 UNTUK KEAMANAN *FILE* PDF ANDROID

Dika Pramudia[1], Aswin Fitriansyah[2], Randi Ramliyana[3]
Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer,
Universitas Indraprasta PGRI
Jalan Raya Tengah No 80, Kelurahan Gedong, Pasar Rebo, Jakarta Timur
dikapramudia@outlook.co.id[1], aswin.fitriansyah@gmail.com[2], randi.ramliyana@gmail.com[3]

Abstrak

Keamanan informasi menjadi hal yang sangat diperhatikan, khususnya sekarang dimana semua aspek informasi mulai masuk era digitalisasi. Bahkan file dokumen yang bersifat rahasia atau penting mulai terintegrasi dengan digital. Sedangkan media pertukaran informasi sekarang lebih fleksibel dengan adanya perangkat telepon pintar dengan operasi sistem seperti android. Dari aspek tersebut diharuskan adanya bentuk keamanan yang diterapkan sekaligus mampu menjaga keaslian dari file dokumen. Banyak konsep keamanan yang disediakan oleh aplikasi, salah satu penerapannya dengan metode steganografi yang digabungkan dengan enkripsi dimana akan diterapkan pada perangkat dengan operasi sistem android serta difokuskan untuk file dokumen PDF yang bersifat sangat penting atau rahasia. Adapun rumusan masalah antara lain bagaimana cara pembuatan aplikasi hybrid steganografi EOF dan enkripsi AES-128 untuk keamanan file PDF pada Android dan bagaimana cara kerja steganografi EOF untuk menyisipkan pesan teks dan enkripsi AES untuk mengenkripsi pesan teks dalam file PDF. Sedangkan tujuan dari penelitian yakni memahami cara kerja steganografi EOF dalam menyisipkan pesan teks kedalam file dengan format PDF, memahami cara kerja AES dalam mengenkripsi pesan teks, dan implementasi bentuk penyisipan informasi dengan metode steganografi EOF dan metode enkripsi AES. Hasil penelitian mempengaruhi ukuran file PDF yang dimana ukuran yang asli daripada file lebih kecil dibandingkan dengan ukuran file yang terdapat stego text didalamnya, dan file yang didalamnya terdapat stego text berhasil diverifikasi keasliannya pada aplikasi. Sehingga membentuk sebuah penanda didalam *file* PDF yang tidak terlihat mata serta hanya pengguna saja yang mengetahuinya.

Kata Kunci : Steganografi EOF, Enkripsi AES-128, Android

Abstract

Information security is a matter of great concern, especially now that all aspects of information are entering the digitalization era. Even document files that are confidential or important begin to be integrated digitally. Meanwhile, the information exchange media is now more flexible with the existence of smart phone devices with operating systems such as Android. From this aspect, it is necessary to have a form of security that is applied while being able to maintain the authenticity of the document file. Many security concepts are provided by the application, one of which is the application of the steganographic method combined with encryption which will be applied to devices with the Android operating system and is focused on PDF document files that are very important or confidential. The problem formulations include how to create a hybrid application of EOF steganography and AES-128 encryption for PDF file security on Android and how EOF steganography works to insert text messages and AES encryption to encrypt text messages in PDF files. While the objectives of the research are to understand how EOF steganography works in inserting text messages into PDF format files, understand how AES works in encrypting text messages, and implement information embedding forms using the EOF steganography method and the AES encryption method. The results of the study affect the size of the PDF file where the original size of the file is smaller than the size of the file that has stego text in it, and the file that contains stego text has been verified for authenticity in the application. So that it forms a marker in the PDF file that is not visible to the eye and only the user knows it.

***Keyword** : Steganography EOF, Encryption AES-128, Android*

PENDAHULUAN

Era Digitalisasi seperti sekarang ini, dimana pertukaran informasi lebih fleksibel dengan adanya sistem operasi seperti android yang ada pada perangkat telepon pintar (*smartphone*). Berkaitan dengan hal tersebut keamanan informasi sangat dibutuhkan. Dalam instansi pemerintah atau swasta tentu memiliki banyak dokumen atau file yang bersifat penting, sehingga diperlukan keamanan agar tidak disalahgunakan oleh pihak – pihak yang tidak berkepentingan. Beberapa aspek keamanan informasi antara lain yaitu *confidentiality* (kerahasiaan), *authentication* (otentikasi), *integrity* (keutuhan data), *non repudiation* (nir-penyangkalan), *access control* (kontrol pengguna). Konsep keamanan informasi juga terbagi menjadi beberapa macam salah satunya adalah metode steganografi. Dalam penelitian ini metode steganografi digabungkan dengan enkripsi yang diterapkan pada operasi sistem android untuk file dokumen PDF yang memiliki sifat penting dan rahasia.

Melalui dua metode tersebut pada penelitian ini akan diterapkan pada file berbentuk PDF dimana hasil yang didapatkan akan terlihat seperti *watermark* yang tidak kasat mata dan hanya pemilik saja yang mengetahui. Hasil dari file tersembunyi tersebut tidak dapat dibaca oleh pihak lain serta terjaga keamanan dan keasliannya. Berhubungan dengan adanya risiko dan dampak besar yang mungkin ditimbulkan dan didorong dengan terbatasnya sistem keamanan dan aplikasi keamanan yang tidak terintegrasi dengan baik. Dengan demikian dalam penelitian ini memberikan solusi mengenai cara pembuatan aplikasi *Hybrid Steganografi* EOF dan Enkripsi AES 128 untuk keamanan file PDF pada android. Dijelaskan pula cara kerja steganografi EOF dan Enkripsi AES 128. Dengan penelitian ini diharapkan dapat memahami cara kerja steganografi EOF dalam menyisipkan pesan teks ke dalam file pada format PDF serta cara kerja AES dalam mengenkripsi pesan teks. Fungsi dan kegunaan yang didapatkan dari segi individu maupun kelompok kerja yaitu mampu mengamankan file dokumen PDF yang ada pada sistem berbasis android. Penelitian ini dilengkapi pula dengan penerapan dan implementasinya pada perangkat android.

PENELITIAN RELEVAN

Menurut (Yahya, 2018) dan (Jannah et al., 2018) memaparkan bahwa steganografi berbeda dengan kriptografi, dimana kriptografi mengubah bentuk data untuk menjaga informasi agar tetap aman. Sedangkan steganografi adalah metode menyembunyikan data yang membuat pihak lain tidak dapat melihat atau mencuri informasi rahasia yang disisipkan. Berdasarkan pemaparan (Hariady et al., 2016) dan (Shih, 2020) maka dapat disimpulkan bahwa steganografi adalah bentuk sistem keamanan yang menerapkan konsep menyembunyikan suatu bentuk informasi kedalam *file* dengan tujuan agar informasi tersebut tidak bisa dibaca ataupun dicuri oleh pihak yang tidak diinginkan. Tiga tipe steganografi antara lain yaitu *Technical Steganography*, *Linguistic Steganography* dan *Digital Steganography.*

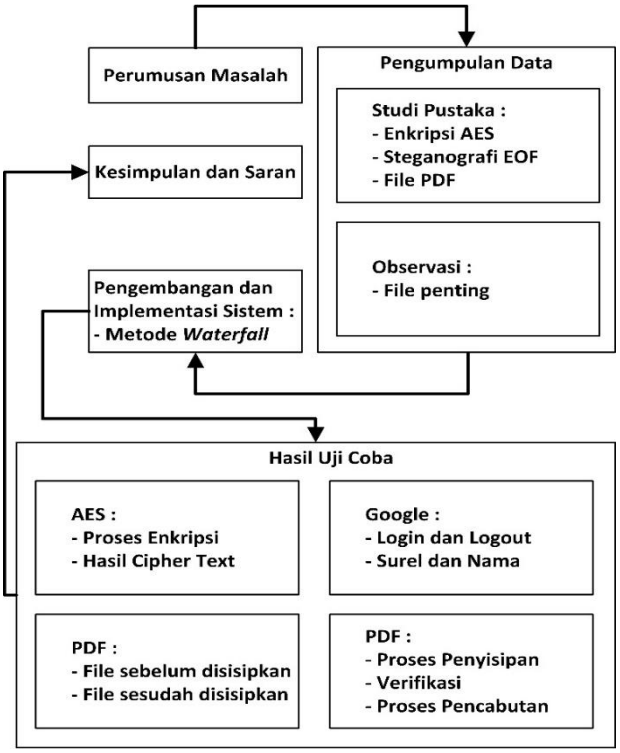
(Haunts, 2019) berpendapat bahwa kriptografi, atau enkripsi berkaitan dengan pembuatan catatan tertulis dan penerapan proses matematika didalamnya untuk membuat pesan tidak dapat dibaca. Saat menggunakan kriptografi, teks asli yang ingin dienkripsi disebut sebagai *plaintext*. Sejalan dengan pendapat (Aumasson, 2018) dan (Tripathi & Agrawal, 2014) bahwa enkripsi adalah aplikasi utama dari kriptografi. Dalam enkripsi menggunakan algoritma yang disebut sandi dan nilai rahasia yang disebut kunci. File asli yang belum terenkripsi disebut *plaintext* dan data yang sudah dienkripsi disebut *ciphertext.* Dapat disimpulkan bahwa enkripsi mampu mengamankan bentuk informasi yang bersifat rahasia, dengan metode penyandian yang rumit serta penerapan kunci untuk mengakses informasi yang dienkripsi atau disebut dekripsi. Kriptografi berdasarkan kunci yang digunakan terbagi menjadi dua metode yaitu enkripsi simetris dan enkripsi asimetris. Pendapat (Aumasson, 2018) mengenai Algoritma *Adavance Encryption Standard* (AES) memproses menggunakan *secret key* yang memiliki panjang kunci 128, 192 dan 256 *bit* yang digunakan untuk bentuk aplikasi yang ingin dikembangkan dengan tingkat keamanan yang berbeda - beda. Tahapan yang terjadi pada AES meliputi *AddRoundKey, SubBytes, ShiftRow* dan *MixColumn,* tahap tersebut akan dilakukan pada putaran untuk proses enkripsi dan pada putaran terakhir tidak akan dilakukan *MixColumn.*

Hasil penelitian terdahulu yang dapat digunakan sebagai acuan ditunjukkan pada Tabel 1 di bawah.

Tabel 1. Hasil Penelitian Terdahulu

No	Peneliti, tahun	Judul Penelitian	Hasil
1	(Mu'Mi, 2017)	Steganografi Citra Menggunakan Kriptografi *Hybrid Playfair Cipher* dan *Caesar Cipher*.	Mengkaji permasalahan tentang kriptografi dan steganografi secara matematis khususnya penyisipan pesan pada citra menggunakan *hybrid playfair cipher* dan *caesar ciphers* dengan bantuan program MATLAB.
2	(Disimbar, 2017)	Penyembunyian Pesan pada *Image* Berformat JPEG dengan Metode LSB dan *Vigenere Chiper*.	Hasil yang didapatkan dapat memberikan keamanan untuk menyembunyikan, menjaga kerahasiaan data, serta mengintegrasikan data sehingga hanya orang tertentu yang dapat mengakses data.
3	(Hariady et al., 2016)	Keamanan dan Penyisipan Pesan Rahasia pada Gambar Dengan Enkripsi *Bowfish* dan Steganografi *End Of File*.	Hasil yang diperoleh *user* dapat melakukan pengamanan data yang akan dikirim dengan menggunakan aplikasi pengiriman pesan rahasia yang telah ter-enkripsi dan disisipkan di dalam sebuah media berupa gambar.

METODE PENELITIAN



Gambar 1. Tahapan Penelitian

Desain Penelitian

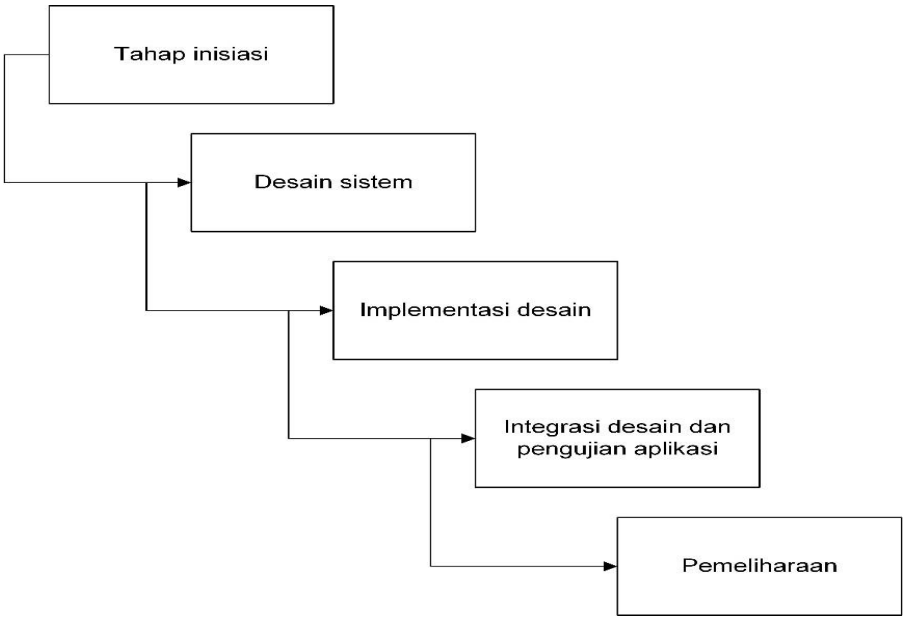
Didasarkan pada penelitian (STOICA et al., 2013) desain penelitian yang digunakan dalam pengembangan perangkat lunak adalah *waterfall* atau sering disebut siklus hidup klasik (*Classic Life Cycle*). Model *waterfall* mudah dimengerti dan diterapkan dalam pengembangan perangkat lunak. Model *waterfall* direkomendasikan untuk kasus – kasus antara lain persyaratan dipahami dengan baik, jelas dan final, definisi produk stabil, teknologi dapat dipahami, tidak ada persyaratan

yang ambigu, sumber daya yang melibatkan keahlian tersedia secara bebas dan termasuk proyek dalam waktu singkat.

Teknik Pengumpulan Data

Teknik yang digunakan dalam pengumpulan data yaitu observasi secara langsung dan dengan studi pustaka. Observasi langsung dengan cara mengamati objek penelitian dari jumlah individu dalam jangka waktu yang bersamaan. Dengan pengamatan proses sistem dan identifikasi dokumen penting sehingga dapat ditentukan metode keamanan yang sesuai untuk diterapkan. Sedangkan pada studi pustaka diperoleh dari berbagai sumber karya tulis ilmiah yang dapat dijadikan referensi dan mendukung penelitian yang dilakukan.

Langkah Pengembangan Sistem



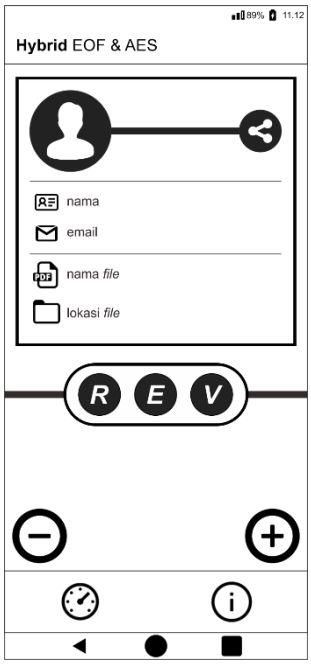
Gambar 3. Model *Waterfall*
Sumber: (Seema Suresh Kute & Surabhi Deependra Thorat, 2017)

Tahap pertama adalah tahap inisiasi yang akan dilakukan evaluasi mengenai kebutuhan sistem keamanan yang mampu menjaga keaslian *file* PDF yang penting atau rahasia. Setelah itu tahap selanjutnya desain sistem yaitu dilakukan penggambaran awal mengenai bagaimana aplikasi akan dibuat serta proses pengerjaan aplikasi dengan metode steganografi dan enkripsi. Tahap berikutnya, implementasi desain dilakukan perancangan tampilan antarmuka pengguna (*user interface*) aplikasi serta penulisan kode program. Kemudian tahap Integrasi desain dan pengujian aplikasi, dimana akan diuji tiap prosesnya pada contoh *file* PDF yang digunakan berhasil atau tidak. Tahap terakhir adalah pemeliharaan aplikasi, dimana akan dilakukan peninjauan kembali dari aplikasi yang telah digunakan dalam jangka waktu tertentu dan tinjauan aplikasi telah memenuhi kebutuhan instansi.

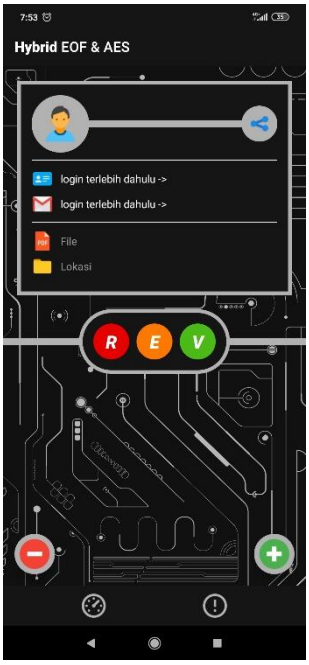
 e-ISSN : 2715-8756

HASIL DAN PEMBAHASAN

Hasil Analisis Rancangan Layar dan Implementasi Layar



Gambar 4. Rancngan Menu Utama



Gambar 5. Implemntasi Menu Utama

Gambar 6. Rancngan Menu Info

Gambar 7. Implementasi Menu Info

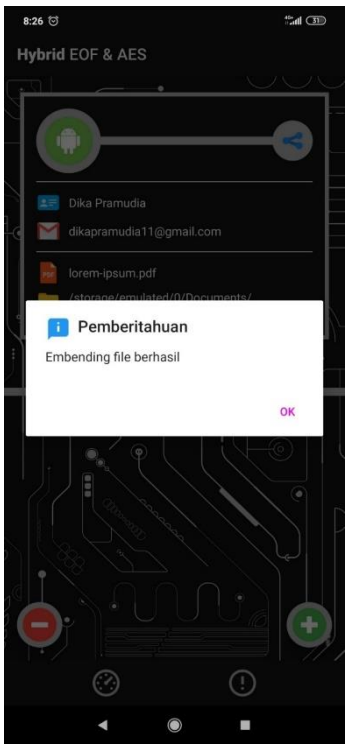
Pada gambar 4 merupakan rancangan menu utama dimana *file* PDF akan diproses seperti penyisipan *cipher text* kedalam *file*, pencabutan *stego text* yang terdapat didalam *file* serta verifikasi *stego text* yang ada dalam *file.* Sedangkan pada gambar 5 merupakan informasi mengenai penggunaan aplikai secara singkat, dan tempat *login* serta *logout* akun Google pengguna dari aplikasi.

Penggunaan Program (Aplikasi)

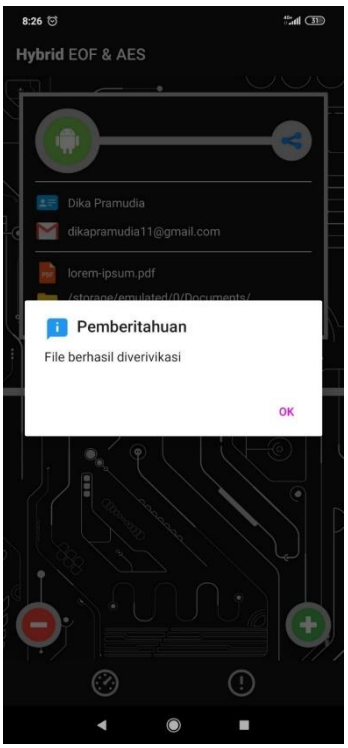
Untuk menyisipkan teks pada file. Tekan tombol plus pada posisi kanan bawah untuk membuka anatarmuka pemilihan file yang igin disisipkan. Pilih file yang ingin disisipkan dengan teks pada penyimpanan perangkat. Ketika file sudah terpilih, antarmuka pemilihan file akan tertutup dan user akan masuk kembali ke menu utama. Tekan tombol E untuk memulai penyisipan. Tunggu sampai proses penyisipan teks selesai. Dan terakhir akan muncul pemberitahuan yang menunjukkan bahwa proses penyisipan selesai dilakukan dan berhasil tidaknya teks disisipkan.

Untuk mencabut teks yang sudah disisipkan pada file. Tekan tombol plus pada posisi kanan bawah untuk membuka anatarmuka pemilihan file. Pilih file yang ingin diinginkan untuk mencabut teks yang tersisipkan pada penyimpanan perangkat. Ketika file sudah terpilih, antarmuka pemilihan file akan tertutup dan user akan masuk kembali ke menu utama. Tekan tombol revoke untuk memulai mencabut teks yang disisipkan. Tunggu sampai proses pencabutan teks selesai. Dan terakhir akan muncul pemberitahuan yang menunjukkan bahwa proses pencabutan selesai dilakukan. Dan berhasil tidaknya teks dicabut.

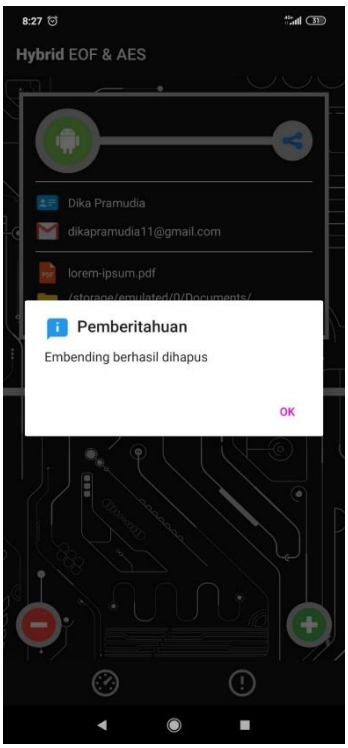
Untuk melakukan verifikasi pada file. Tekan tombol plus pada posisi kanan bawah untuk membuka anatarmuka pemilihan file. Pilih file yang ingin diinginkan untuk diverifikasi pada penyimpanan perangkat. Ketika file sudah terpilih, antarmuka pemilihan file akan tertutup dan user akan masuk kembali ke menu utama. Tekan tombol *verify* untuk memulai verifikasi pada file. Tunggu sampai proses verifikasi selasi dilakukan. Dan terakhir akan muncul pemberitahuan yang menunjukkan bahwa file terverifikasi atau tidak.



Gambar 8. Penyisipan *Ciphertext*



Gambar 9. Verifikasi *Ciphertext*

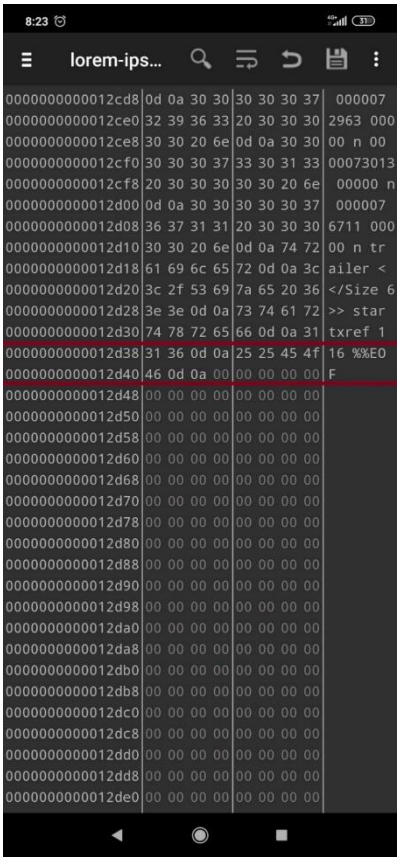


Gambar 10. Pencabutan *Ciphertext*

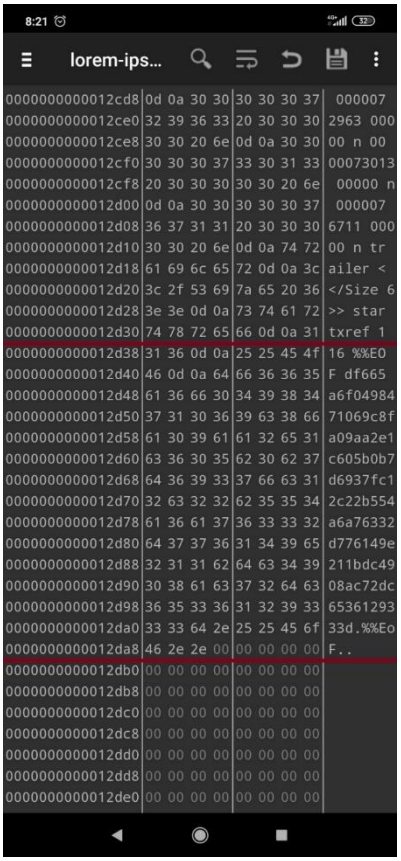
Uji Coba Program

Untuk teks yang disisipkan berupa gabungan dari nama dan *email* akun Google ketika *login* pada aplikasi yang terubah menjadi *cipher text*. Kunci enkripsi AES merupakan hasil *hash* dari gabungan nama dan *email* menggunakan algoritma MD5. Pada akhir *cipher text* akan ditambahkan penanda untuk mengindikasi *stego text*. Dengan aplikasi *Hex Editor* pada gambar 8 dan 9 akan ditampilkan perbedaan *byte file* PDF dimana pada gambar 8 belum tersisipkan *stego text* sedangkan pada gambar 9 sudah tersisipkan *stego text*. Ukuran pada *file* juga berubah karena adanya

penambahan *byte* dimana *file* asli memiliki ukuran yang lebih kecil daripada *file* yang sudah tersisipkan. Ukuran *file* juga bergantung pada berapa karakter *stego text* yang disispkan kedalam *file*.



Gambar 11. *Byte File* Sebelum Disisipkan *Ciphertext*



Gambar 12. *Byte File* Sesudah Disisipkan *Ciphertext*

Tabel 2. Proses dan hasil teks

No	Variabel	Data
1	*Plain Text*	Dika Pramudia : dikapramudia11@gmail.com
2	*Key* (MD5)	9456ea40d0db1dc5ccb49f89a36158e2
3	Ciphertext (AES-128)	df665a6f0498471069c8fa09aa2e1c605b0b7d6937fc12c22 b554a6a76332d776149e211bdc4908ac72dc6536129333d

SIMPULAN

Simpulan yang dapat diperoleh diantaranya:

1. Aplikasi dikerjakan menggunakan bahasa program Java dimana berjalan pada sistem operasi Android. Sedangkan lingkungan pengembang terintegrasi menggunakan Android Studio yang didesain khusus dalam pengembangan aplikasi berbasis Android.
2. Proses penyisipan menggunakan steganografi EOF mengharuskan teridentifikasinya byte terkahir pada file. Tujuan identifikasi tersebut memberikan posisi yang tepat dimana ciphertext akan disisipkan. Sedangkan metode enkripsi menggunaka algoritma AES menjadikan informasi tidak bisa dikenali atau dibaca. Dimana AES-128 bekerja dengan panjang kunci 128 bit.

DAFTAR PUSTAKA

Aumasson, J.-P. (2018). *Serious cryptography : a practical introduction to modern encryption* (B. D. Reed (ed.); 1st ed.). No Starch Press, Inc.

Disimbar, N. D. S. (2017). *Penyembunyian Pesan Pada Image Berformat JPEG Dengan Metode LSB dan Viginere Chiper*. Universitas Muhammadiyah Ponorogo.

Hariady, M. M., Suyatno, A., & Astuti, I. F. (2016). Keamanan Dan Penyisipan Pesan Rahasia Pada Gambar Dengan Enkripsi Blowfish Dan Steganografi End Of File. *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, *11*(2), 1. https://doi.org/10.30872/jim.v11i2.207

Haunts, S. (2019). Applied Cryptography in .NET and Azure Key Vault. In *Applied Cryptography in .NET and Azure Key Vault*. https://doi.org/10.1007/978-1-4842-4375-6

Jannah, L. M., Santoso, I., & Christyono, Y. (2018). Kinerja Steganografi Metode End of File Pada Data Citra Digital. *Transient*, *7*(1), 34. https://doi.org/10.14710/transient.7.1.34-39

Mu'Mi, N. F. A. (2017). *Steganografi Citra Menggunakan Kriptografi Hybrid Playfair Cipher dan Caesar Cipher*. Universitas Negeri Makassar.

Seema Suresh Kute, P., & Surabhi Deependra Thorat, P. (2017). A Review on Various Software Development Life Cycle (SDLC) Models. *International Journal of Research in Computer and Communication Technology*, *3*(7), 776–781.

Shih, F. Y. (2020). *Digital Watermarking and Steganography: Fundamentals and Techniques (Second Edition)* (2nd ed.). CRC Press. https://doi.org/https://doi.org/10.1201/9781315121109

STOICA, M., MIRCEA, M., & GHILIC-MICU, B. (2013). Software Development: Agile vs. Traditional. *Informatica Economica*, *17*(4/2013), 64–76. https://doi.org/10.12948/issn14531305/17.4.2013.06

Tripathi, R., & Agrawal, S. (2014). Comparative Study of Symmetric and Asymmetric Cryptography Techniques. *International Journal of Advance Foundation and Research in Computer (IJAFRC)*, *1*(6), 68–76.

Yahya, A. (2018). Steganography techniques for digital images. In *Steganography Techniques for Digital Images*. https://doi.org/10.1007/978-3-319-78597-4