**IMPLEMENTASI TEKNIK STEGANOGRAFI END OF FILE, KOMPRESI LEMPEL ZIV WELCH DAN ALGORITMA TRIPLE DES UNTUK MENGAMANKAN DATA KEDALAM CITRA BERFORMAT BMP**

Oldi Syahputra Ramadhani, Muhamad Radzi Rathomi, Nurul Hayaty

150155201049@student.umrah.ac.id

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Maritim Raja Ali Haji

**Abstract**

*Many techniques and methods are used to maintain data security. Like Cryptographic Techniques, Compression and Steganography. Triple DES (Triple Data Encryption Standard) is an algorithm developed from the DES (Data Encryption Standard) algorithm. Basically the algorithm used is the same, only in Triple DES was developed by encrypting the DES algorithm implementation three times. LZW is the short word for Lempel Ziv Welch. The LZW algorithm compresses using a dictionary, where parts of the text are replaced with indexes obtained from a dictionary. The End of File (EOF) method is a technique that inserts data at the end of a file. EOF can be used to insert data that is the same size as the container file size, and will then output an insert that is larger than the original container file. After being implemented, it was concluded that the Triple DES, LZW and EOF Method were successfully implemented when implemented per method. However, when the methods are combined per 2 methods and 3 methods at the same time, the Triple DES, LZW and EOF methods unable to restore data in its entirety*

***Keywords : Triple DES, Lempel Ziv Welch, End Of File***

1. **Pendahuluan**

Masalah keamanan dan kerahasiaan data merupakan suatu hal yang sangat penting didalam dunia teknologi informasi. Dimana banyak data yang dikirimkan berisi pesan yang bersifat rahasia dan keamanannya harus terjaga. Berbagai macam ancaman seperti serangan cracker, sadap, spam maupun hacker meningkat untuk mencuri data-data yang bersifat rahasia. Ancaman-ancaman kejahatan tersebut dapat menyebabkan data disalahgunakan oleh pihak yang tidak bertanggung jawab. Untuk mengurangi ancaman pencurian data, diperlukan langkah-langkah agar dapat meningkatkan keamanan disaat melakukan pertukaran data informasi.

Kriptografi merupakan sebuah proses yang mengubah pesan awal menjadi sebuah pesan yang sulit dipecahkan dengan menggunakan suatu algoritma tertentu. Dimana data tersebut diacak menggunakan suatu kunci yang telah ditentukan sehingga tidak dapat dibaca oleh pihak yang berkepentingan. Proses kriptografi disebut dengan *encryption*, yaitu suatu proses yang mengkonversikan sebuah pesan *plaintext* menjadi sebuah *ciphertext* yang bisa dibalikan ke bentuk asli seperti semula, yang juga disebut sebagai proses *decryption* (Anggraini dan Sakti., 2014). Ilmu kriptografi pada awalnya dianggap terlarang untuk diajarkan sehingga tidak ada bahan bacaan untuk mempelajarinya. Setelah David Kahn membuat bukunya di tahun 1969, maka ilmu pengamanan data ini menjadi lebih terbuka untuk dipelajari. Saat ini sudah sangat banyak buku yang membahas mengenai hal ini, mulai dari yang umum (tidak teknis) sampai ke yang teknis (Rahardjo, 2017). Salah satu algoritma kriptografi adalah *Triple* DES. *Triple* DES (*Triple Data Encryption Standard)* merupakan Suatu algoritma pengembangan dari algoritma DES (*Data Encryption Standard*). Pada dasarnya algoritma yang digunakan sama, hanya pada *Triple* DES dikembangkan dengan melakukan enkripsi dengan implementasi algoritma DES sebanyak tiga kali.

Steganografi merupakan sebuah teknik atau ilmu untuk menyembunyikan pesan atau data rahasia. Jadi data atau pesan yang sudah dienkripsi akan disisipkan ke sebuah media agar tidak diketahui oleh orang yang tidak berkepentingan. Steganografi berguna untuk menyembunyikan suatu data rahasia sehingga sulit di deteksi, dan juga dapat melindungi hak cipta dari suatu produk (Jannah et al., 2018). Salah satu metode steganografi adalah EOF. EOF (*End Of File*) merupakan salah satu teknik yang menyisipkan data kedalam citra. Dalam teknik EOF, data yang disisipkan pada citra diberi tanda khusus sebagai pengenal dibagian akhir dari citra tersebut.

Kompresi merupakan salah satu metode untuk mengecilkan data sehingga diperoleh data yang ukurannya lebih kecil daripada data aslinya (Satyapratama et al., 2016). Salah satu metode kompresi adalah LZW. LZW merupakan kependekan kata dari Lempel-Ziv-Welch. Abraham Lempel, Jacob Ziv, dan Terry Welch adalah penciptanya. Dengan dikembangkan sebuah sistem dengan mengimplementasikan sistem keamanan data menggunakan steganografi dengan metode *End Of File* (EOF) dan Algoritma *Data Encryption Standard* (DES) berbasis Java *Programming* diharapkan dapat melindungi data rahasia perusahaan agar tidak mudah terbaca oleh orang yang tidak berkepentingan (Anggraini dan Sakti, 2014), namun Namun, metode DES rawan terkena serangan *bruteforce* karena panjang key nya hanya 56 bit saja (Garg et al., 2017). Dengan metode LSB (*Least Significant Bit*) dan *Triple* DES, *image* yang disisipkan pesan atau dokumen tidak terlalu banyak terlihat perbedaan dari citra warna terkecuali disisipkan pesan atau dokumen dengan ukuran besar (Harjo et al., 2016) Oleh karena itu, akan diimplementasikan kriptografi dan steganografi ditambah kompresi LZW untuk memampatkan data dan mengurangi ukuran dari data tersebut. untuk Metode yang akan dipakai untuk mengimplementasikan adalah metode *Triple Data Encryption Standard*, *Lempel Ziv Welch* dan *End Of File*. format BMP (*Bitmap Picture*) dipilih untuk format citra ketika disisipkan karena Nilai PSNR (*Peak Signal to Noise Ratio*) setelah disisipkan lebih besar dan lebih baik daripada format JPEG, PNG dan TIFF (Jannah et al., 2018) sehingga mengurangi kecurigaan kalau citra telah disisipkan data.Tujuan penelitian ini adalah mengimplementasikan metode dengan mengenkripsi, kompresi dan menyembunyikan data kedalam citra menggunakan metode *Triple* DES*, Lempel Ziv Welch* dan *End Of file*. Manfaat dari penelitian ini adalah membuktikan apakah metode *Triple* DES, Kompresi *Lempel Ziv Welch* dan *End Of File* apakah bisa diimplementasikan untuk keamanan data, dapat digunakan untuk mengamankan data dari pihak yang tidak bertanggung jawab dan menambah pengetahuan tentang bagaimana penerapan metode *Triple* DES, *Lempel Ziv Welch* dan *End Of File* untuk mengamankan data.

1. **Metode Penelitian**
	1. **Triple DES**

*Triple* DES (*Triple Data Encryption Standard*) merupakan suatu algoritma pengembangan dari algoritma DES (*Data Encryption Standard*). Pada dasarnya algoritma yang digunakan sama, hanya pada *Triple* DES dikembangkan dengan melakukan enkripsi dengan implementasi algoritma DES sebanyak tiga kali. Pada algoritma *Triple* DES dibagi menjadi tiga tahap, setiap tahapnya merupakan implementasi dari algoritma DES. Pada *Triple* DES, kuncinya sepanjang 56 bit. tiga kunci yang digunakan bisa bersifat saling bebas (K1 ≠ K2 ≠ K3), hanya dua buah kunci yang saling bebas dan satu kunci lainnya sama dengan kunci pertama (K1 ≠ K2 dan K3 = K1) ataupun hanya menggunakan satu buah kunci yang sama (Gunawan et al., 2014).

Langkah-langkah algoritma DES menurut Susanto et al (2017) adalah :

1. Ubah Plainteks dan kunci kedalam Biner.
2. Lakukan pengurutan plainteks dari bilangan biner karakter awal sampai akhir menjadi sebanyak 64 bit.
3. Lakukan operasi *Initial Permutation* pada bit plainteks menggunakan Tabel IP.
4. Setelah itu *generate* biner kunci menggunakan Tabel permutasi kompresi (PC1) untuk mendapatkan kunci internal. Pada langkah ini, membuang masing-masing 1 blok bit paling belakang dari 64 bit menjadi 56 bit.
5. Hasil dari kompresi dipecah jadi 2 bagian, kiri dan kanan yang masing-masing panjangnya 28 bit per bagian. Dan setelah itu bagian kiri dan kanan bit disebut C0 dan D0
6. Lakukan pergeseran kiri (*left shift*) pada 2 bagian bit yang dipecah sebanyak 1 atau 2 kali selama 16 putaran, Jumlah pergeseran tergantung dari tabel *left shift*.
7. Setiap hasil putaran digabungkan kembali dan dimasukkan ke tabel permutasi kompresi (PC2) dan terjadi kompresi data kembali dari 56 bit menjadi 48 bit sehingga didapatkan hasilnya yaitu 16 kunci internal.
8. Masuk ke bagian *enciphering* setelah selesai melakukan permutasi awal. setiap blok plainteks mengalami 16 kali putaran, Dan blok plainteks terbagi menjadi dua yaitu bagian kiri (L) dan kanan (R). kedua bagian ini akan masuk kedalam proses 16 putaran DES. Pada setiap putaran, bagian R merupakan masukan untuk fungsi transformasi yang disebut f. di fungsi f, blok R di XOR dengan kunci internal K. keluaran dari fungsi f setelah di XOR, akan di XOR kan dengan bagian L untuk mendapatkan bagian R yang baru, Sedangkan bagian L yang baru nantinya diambil dari bagian R sebelumnya. secara matematis, satu putaran DES dinyatakan sebagai berikut :

|  |  |
| --- | --- |
| Li = Ri-1 | (1) |
| Ri= Li-1 ⊕ f(R1, Ki) | (2) |

1. Lebih jelasnya, *enciphering* diawali dari proses ekspansi R0 dari 32 bit menjadi 48 bit dengan menggunakan tabel ekspansi.
2. lakukan operasi XOR pada R0 dan kunci.
3. Selanjutnya, lakukan operasi *substitution box*, dimana setiap blok terdiri dari 6 bit.
4. Hasil dari proses *S-Box* akan dimasukkan kedalam tabel operasi *P-box*. Nah dari sini akan didapatkan nilai R1. Setelah itu lakukan langkah *enciphering* lagi sampai 16 kali dengan memakai cara yang sama.
5. Terakhir adalah menggabungkan R16 dan L16 kemudian dipermutasikan untuk terakhir kalinya dengan *invers initial permutation*. Output yang dihasilkan adalah Cipherteks didalam biner. Chiperteks yang didapat merupakan operasi pertama. Lakukan langkah-langkah algoritma DES sebanyak 3 kali untuk mendapatkan cipherteks akhir dari Triple DES.

**2.2 End Of File**

Metode *End of File* (EOF) merupakan salah satu teknik yang menyisipkan data kedalam citra. Teknik ini dapat digunakan untuk menyisipkan data yang ukurannya sama dengan ukuran citra atau menyisipkan data yang ukurannya lebih besar dari citra. Dalam teknik EOF, data yang disisipkan pada citra diberi tanda khusus sebagai pengenal dibagian akhir dari citra tersebut. Metode *End Of File* (EOF) dengan *Least Significant Bit* (LSB) tidak begitu banyak perbedaan dalam alur algoritmanya, namun terdapat perbedaan yaitu pada data yang disisipi dan *output*. Pada metode LSB, data yang disisipi ukurannya harus lebih kecil dari citra yang akan disisipi, tetapi lain halnya pada metode EOF ukuran data yang akan disisipi bisa lebih besar dari ukuran citranya. Pada metode LSB citra yang telah disisipi data tidak terlalu mempengaruhi ukuran citranya, tetapi akan mempengaruhi kualitas citranya. Sedangkan pada metode EOF, kualitas citra setelah disisipi data tidak berubah, tetapi akan mengubah ukuran citranya (Pandapotan dan Zebua, 2016).

**2.3 Lempel Ziv Welch**

LZW merupakan kependekan kata dari Lempel-Ziv-Welch. Abraham Lempel, Jacob Ziv, dan Terry Welch adalah penciptanya. Algoritma ini melakukan kompresi dengan menggunakan *dictionary*. 0-255 nomor *dictionary* pertama berasal dari kode ASCII sedangkan nomor *dictionary* selanjutnya berasal dari karakter pertama dan karakter berikutnya dari sebuah masukan (Kamble, 2015) Ini sangat efektif karena banyak karakter dapat dikodekan dengan mengacu pada string yang telah muncul sebelumnya dalam teks (Namburi, 2020). Adapun beberapa hal yang biasanya digunakan dalam menguji metode LZW ini menurut Hidayat et al. (2015) adalah Rasio Kompresi, Faktor Kompresi dan Persentase penghematan. Rasio Kompresi adalah membandingkan ukuran data awal sebelum dikompresi dan data setelah dikompresi. Semakin kecil nilai rasio kompresi dan nilainya dibawah 1, maka hasil kompresi semakin memuaskan. Ketika nilai rasio diatas 1 maka data terekspansi. Ketika sama dengan 1 maka data tidak terkompresi sama sekali. Rumus untuk menghitung rasio kompresi adalah :

|  |  |
| --- | --- |
| $$Rasio Kompresi=\frac{Ukuran setelah}{Ukuran sebelum}$$ | (3) |

Faktor kompresi menunjukkan seberapa kali kehandalan ketika berhasil memproses satu kompresi. Bedanya dengan rasio kompresi yaitu semakin besar nilai faktor kompresi dan nilainya lebih besar dari 1, maka hasil kompresi semakin memuaskan. Rumus yang digunakan sama seperti rumus rasio kompresi. Sedangkan persentase penghematan adalah persentase perbandingan ukuran data sebelum dikompresi dan sesudah dikompresi dengan ukuran data sebelum dikompresi. Persentase ini menunjukkan seberapa hematnya suatu kompresi. Semakin besar persentase maka ukuran data akan semakin kecil. Rumus untuk menghitung Rasio penghematan adalah :

|  |  |
| --- | --- |
| $$Persentase Penghematan=\frac{Uk.sblm-Uk.stlh}{Uk.sblm} x 100\%$$ | (4) |

**2.4 Alur Kerja Aplikasi**

Alur Kerja Aplikasi merupakan alur untuk menjalankan aplikasi/sistem sesuai dengan kebutuhan. Alur kerja aplikasi terbagi menjadi beberapa bagian, yaitu:

1. Alur kerja Metode *Triple* DES

Alur kerja metode *Triple* DES merupakan alur proses enkripsi dan dekripsi menggunakan metode *Triple* DES. Alur dimulai dengan memasukkan data plainteks yang berada didalam *file* bertipe txt beserta memasukkan kuncinya. Setelah dimasukkan, data akan dienkripsi dengan metode *Triple* DES. Ketika berhasil, hasil dari proses enkripsi akan menghasilkan chiperteks dan disimpan kedalam *file* bertipe txt. Proses alur dekripsi pun hampir sama seperti enkripsi. Yang membedakan hanya data yang dimasukkan di proses alur dekripsi adalah data chiperteks bertipe txt dan hasil dari proses dekripsi akan menghasilkan plainteks dan disimpan kedalam data yang ada didalam *file* bertipe txt.



Gambar 1. *Flowchart* Alur kerja metode *Triple* DES

1. Alur kerja Metode LZW

Alur kerja metode LZW merupakan alur kompresi dan dekompresi data. Alur dimulai dengan memasukkan data yang berada didalam *file* bertipe txt. Kemudian data dikompresi sehingga didapatkan data hasil proses kompresi dan menyimpannya kedalam *file* txt. Proses alur dekompresi juga hampir sama dengan kompresi, yang berbeda hanya data yang dipakai untuk proses dekompresi adalah data yang telah dikompresi dan hasil dari dekompresi adalah data awal sebelum dikompresi.



Gambar 2. *flowchart* alur kerja metode LZW

1. Alur kerja Metode EOF

Alur kerja metode EOF merupakan alur menyisipkan dan mengeluarkan data (mengekstrak) dari citra. Alur dimulai dengan memasukkan citra bmp yang akan menjadi penampung data terlebih dahulu, kemudian masukkan data yang berada didalam *file* bertipe txt untuk disisipkan kedalam citra. Selanjutnya, data akan diproses dengan metode EOF sehingga hasil dari proses metode EOF menghasilkan citra yang telah disisipkan data (*stegoimage*). Proses ekstrak hampir sama, bedanya hanya disaat mengekstrak, yang dimasukkan adalah citra yang telah disisipkan data, lalu diproses dengan metode EOF untuk mengeluarkan data dari citra.



Gambar 3. *flowchart* Alur kerja metode EOF

1. Alur kerja metode *Triple* DES, LZW dan EOF

Alur kerja metode *Triple* DES, LZW dan EOF merupakan alur kerja gabungan dari 3 metode yang digunakan dalam penelitian ini. Alur dimulai dengan memasukkan data plainteks bertipe txt beserta kuncinya. Data yang sebelumnya dimasukkan kemudian dienkripsi dengan metode *Triple* DES. Ketika berhasil, hasil dari proses enkripsi akan menghasilkan chiperteks dan disimpan kedalam file bertipe txt. Selanjutnya, data chiperteks akan dikompresi dengan menggunakan metode LZW. hasil dari kompresi dengan metode LZW juga akan disimpan kedalam data bertipe txt. Selanjutnya masukkan citra bmp yang akan menjadi penampung data dan sisipkan data hasil kompresi kedalam citra bmp dengan metode EOF. Hasil dari proses metode EOF akan menghasilkan citra yang telah disisipkan data (*stegoimage*)*.*



Gambar 4. *Flowchart* Alur kerja gabungan 3 metode

1. **Hasil dan Pembahasan**

**3.1 Persiapan dan Hasil Pengujian**

Data yang akan digunakan pada penelitian adalah beberapa data berjenis teks bertipe txt dan citra bertipe bmp sebagai media penampung. Data akan diuji per metode dan hasilnya seperti terlihat pada tabel (1) (2) dan (3) dan terakhir diuji per 3 metode dan hasilnya terlihat seperti tabel (4).

Tabel **1.** Pengujian metode *Triple* DES

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Ukuran data****Teks** | **Ukuran data setelah enkripsi** | **Berhasil Dekripsi ?** |
| **1** | 8 bytes | 12 bytes | Ya |
| **2** | 16 bytes | 23 bytes | Ya |
| **3** | 64 bytes | 89 bytes | Tidak semuanya |
| **4** | 7.12 kb | 11 kb | Tidak semuanya |

Berdasarkan Tabel 1, Data pertama dan kedua yang digunakan untuk pengujian berhasil dienkripsi lalu didekripsi dan data berhasil kembali seperti semula sedangkan data lainnya tidak semuanya berhasil kembali seperti semula karena data berukuran lebih besar dan tidak dapat terbaca setelah didekripsi.

Tabel **2.** Pengujian metode LZW

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Ukuran data****Teks** | **Ukuran data****setelah kompresi** | **Rasio****Kompresi** | **Faktor****Kompresi** | **Persentase****Penghematan** |
| **1** | 56 Bytes | 34 Bytes | 0.618 | 0.618 | 38.181 |
| **2** | 1.10 kb | 1.05 kb | 0.995 | 0.995 | 0.495 |
| **3** | 7.12 kb | 4.12 kb | 0.578 | 0.578 | 42.134 |
| **4** | 87.6 kb | 29 kb | 0.331 | 0.331 | 66.894 |

Berdasarkan Tabel 2, Data berhasil dikompresi dengan metode LZW. Ukuran data semuanya menjadi kecil setelah dikompresi dengan Rasio Kompresi paling bagus adalah 0.331, Faktor Kompresi paling bagus adalah 0.995 dan Persentase Penghematan paling bagus adalah 66.894

Tabel **3.** Pengujian metode EOF

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Ukuran data****Teks** | **Ukuran data****Citra penampung** | **Ukuran****Stego** |
| **1** | 16 bytes | 2.25 mb(2.359.434 bytes) | 2.25 mb (2.359.462 bytes) |
| **2** | 56 bytes | 2.25 mb(2.359.434 bytes) | 2.25 mb (2.359.462 bytes) |
| **3** | 64 bytes | 2.25 mb(2.359.434 bytes) | 2.25 mb (2.359.495 bytes) |
| **4** | 1.10 kb | 2.25 mb(2.359.434 bytes) | 2.25 mb (2.359.510 bytes) |

Berdasarkan Tabel 3, Data berhasil disisipkan kedalam citra dan menghasilkan stego yang ukurannya lebih besar dari citra. Ukuran citra stego yang besar dapat mengurangi kecurigaan bahwa citra stego telah disisipkan data.

Tabel **4**. Pengujian metode Triple DES, LZW dan EOF

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Ukuran data****Teks** | **Ukuran Citra penampung** | **Ukuran****Enkripsi** | **Ukuran****Kompresi** | **Ukuran****Stego** | **Berhasil dikembalikan** |
| **1** | 16 bytes | 2.359.434 bytes | 22 bytes | 32 bytes | 2.359.490 bytes | Tidak |
| **2** | 56 bytes | 2.359.434 bytes | 82 bytes | 104 bytes | 2.359.600 bytes | Tidak |
| **3** | 64 bytes | 2.359.434 bytes | 96 bytes | 168 bytes | 2.359.657 bytes | Tidak |
| **4** | 53 bytes | 2.359.434 bytes | 87 bytes | 104 bytes | 2.359.608 bytes | Tidak |

Berdasarkan Tabel 4, Ketika ketiga metode digabungkan, data tidak semuanya berhasil dikembalikan karena ketika melewati proses enkripsi, kompresi dan menyisipkan ke citra, lalu mengeluarkannya dari citra, dekompres dan didekripsi kembali, masing-masing dari setiap proses tidak dapat mengembalikan data secara utuh karena setelah melewati setiap proses, data menjadi berukuran lebih besar sehingga tidak semuanya data berhasil dikembalikan.

1. **Kesimpulan**

Dari hasil pembahasan, dapat disimpulkan bahwa Metode *Triple* DES, LZW dan EOF Berhasil diimplementasikan dengan baik ketika diimplementasikan per metode. Namun ketika metode digabungkan, Metode *Triple* DES, LZW dan EOF tidak berhasil diimplementasikan karena data tidak dapat dikembalikan secara utuh. Diantara ketiga metode yang digunakan, metode terbaik untuk mengamankan data adalah metode EOF karena dapat menggunakan data yang lebih banyak dan besar ditambah ukuran citra sisip yang juga besar sehingga mengurangi kecurigaan bahwa citra telah disisipkan data.

1. **Daftar Pustaka**

Anggraini, Y., dan Sakti, D. V. S. Y. 2014. Penerapan Steganografi *Metode End of File* ( Eof ) Dan Enkripsi Metode *Data Encryption Standard* ( Des ) Pada Aplikasi Pengamanan Data Gambar Berbasis Java. Konferensi Nasional Sistem Informasi, STMIK Dipanegara Makassar, September 2016, 1743–1753.

Garg, S., Garg, T., dan Mallick, B. 2017. *Secure Message Transfer using Triple DES. International Journal of Computer Applications, 165(8),* 1–4. https://doi.org/10.5120/ijca2017913937

Gunawan, H. A., Arifin, Z., dan Astuti, I. F. 2014. Keamanan Login Web Menggunakan Metode 3Des Berbasis Teknologi *Quick Response Code*. Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer (JIM), 2014(Juni), 18–23. <https://doi.org/10.1016/j.bjane.2013.07.009>

Harjo, T. B., Kapriati, M., dan Susanto, D. A. 2016. Aplikasi Steganografi Menggunakan LSB (*Least Significant Bit*) dan Enkripsi *Triple* Des Menggunakan Bahasa Pemrograman C #. Sifotek Global, 6(1), 13–17.

Hidayat, H., Pamungkas, T., dan Zarman, W. 2015. Implementasi Algoritma Kompresi Lzw Pada *Database Server*. Komputa : Jurnal Ilmiah Komputer Dan Informatika, 2(1). <https://doi.org/10.34010/komputa.v2i1.74>

Jannah, L. M., Santoso, I., dan Cristyono, Y. 2014. Kinerja Steganografi Metode *End Of File* Pada Data Citra Digital.

Kamble, S. 2015. FPGA *based Data Compression using Dictionary based “ LZW ” Algorithm*. 5(4), 104–109.

Namburi, S. R. 2020. *A New Approach To Increase Lzw*. 4(10), 141–144.

Pandapotan, S. T., dan Zebua, T. 2016. Analisa Perbandingan *Least Significant Bit* dan *End Of File* Untuk Steganografi Citra Digital Menggunakan Matlab. Seminar Nasional Inovasi Dan Teknologi (SNITI), 3, 604–608. <http://sniti.info/>

Rahardjo, B. 2017. Keamanan informasi.

Satyapratama, A., Yunus, M., Widjianto. 2016. Kompresi File Gambar Bmp Dan Png. 6(2), 69–81.

Susanto, J., Ilhamsyah., Rismawan, T., 2017. Aplikasi Enkripsi Dan Dekripsi Untuk Keamanan Dokumen Menggunakan *Triple* DES Dengan Memanfaatkan USB *Flash Drive*. Jurnal Coding Sistem Komputer Untan Jurnal Coding Sistem Komputer Untan ISSN : 2338-493X. 05(1), 68–79.