PENGGUNAAN ALGORITMA LSB DAN *VIGENERE* UNTUK PENGAMANAN DATA MELALUI POLA CITRA DIGITAL

Lia Cintia Purba[1], Muhammad Zarlis[2], Indra Gunawan[3], Sumarno[4], Zulaini Masruro[5]
[1,3,4,5]Teknik Informatika STIKOM Tunas Bangsa Pematangsiantar
[2]Universitas Sumatera Utara
liacintipurba0704@gmail.com[1] m.zarlis@yahoo.com[2]
indra@amiktunasbangsa.ac.id[3] sumarno@amiktunasbangsa.ac.id[4]
zulaini@amiktunasbangsa.ac.id[5]

Abstrak

Abstrak- Penerapan prosedur keamanan dapat membantu mengamankan data dengan menggunakan algoritma steganografi LSB yang dikombinasikan dengan teknik enkripsi Vigenère Cipher. Dimana Metode LSB digunakan untuk menyembunyikan pesan dengan menggunakan cover media citra digital PNG, serta dikombinasikan dengan teknik enkripsi vigenère yang merupakan tipe abjad paling majemuk yang menggunakan metode substitusi dan termasuk dalam kategori kunci simetris dimana kunci yang digunakan untuk proses enkripsi adalah kunci yang digunakan untuk proses dekripsi, serta menghasilkan pesan tersembunyi dan telah di enkripsi, sehingga tidak menimbulkan kecurigaan.

Kata Kunci: ***Keamanan, Steganografi LSB, Vigenere Cipher***

Abstract

Abstract- The application of security procedures can help secure data by using the LSB steganography algorithm combined with the Vigenère Cipher encryption technique. Where the LSB method is used to hide messages using PNG digital image media cover, and is combined with the Vigenère encryption technique which is the most complex type of alphabet that uses the substitution method and is included in the symmetric key category where the key used for the encryption process is the key used for the encryption process. decryption, and generate hidden messages that have been encrypted, so as not to arouse suspicion.

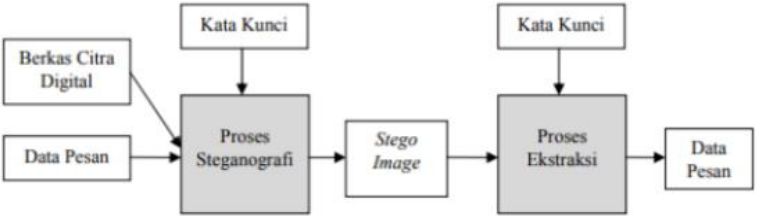
Keywords: Security, LSB Steganography, Vigenere Cipher

1. Pendahuluan

File mencerminkan hasil pekerjaan yang lama serta mahal. terkadang pekerjaan itu tidak bisa diulangi. Boleh dikatakan, informasi yang tersimpan serta dikirim dari *Computer* umumnya jauh lebih bernilai dari pada *computer* itu sendiri. File bisa jadi berharga serta tidak bisa ditukar, Sehingga diperlukan pengamanan. Pada Telkom Akses Kabanjahe keamanan data ialah salah satu aspek berarti bagi suatu sistem data, buat menggapai kerahasiaan, ketersediaan & integritas dalam sumber energi data. Pada sesuatu industri jadi strategi serta tata cara yang dipakai buat menghindari akses yg tidak legal, pergantian program, pencurian, ataupun kehancuran raga terhadap data. Akan tetapi, kasus keamanan ini kurang menerima perhatian menurut para pemilik & pengelola sistem informasi di Telkom Akses Kabanjahe, serta terhubungnya sistem informasinya dengan jaringan *internet* hal ini membuka akses secara *global*. (Maksud akses ini menjadi target & pula menjadi penyerang).

2. Metode Penelitian

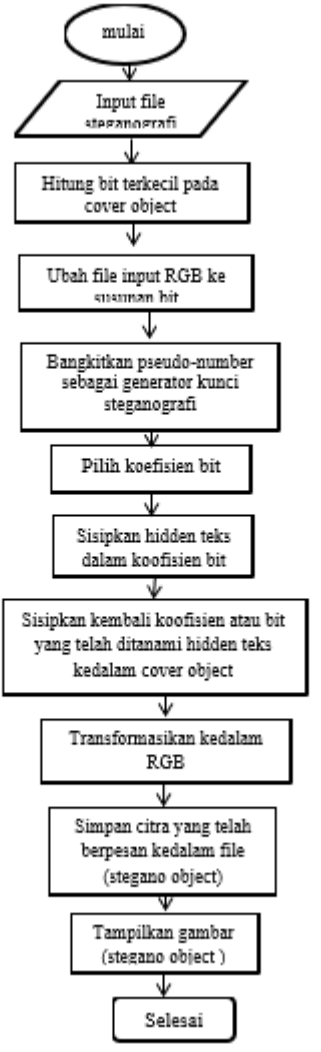
Pada tahap penerapan Algoritma Steganografi LSB data yang telah di peroleh selanjutnya akan dilakukan tahap mengenkripsi teks alfabet dengan menggunakan serangkaian sandi Caesar yang berbeda berdasarkan huruf dari kata kunci, sehingga data akan lebih mudah diolah dengan metode Steganografi LSB. hal ini dilakukan untuk mempermudahkan proses perhitungan algoritma Steganografi *Least significant Bit.* Adapaun model sistem dsteganografi dapat dilihat pada gambar 1 sebagai berikut :



Gambar 1. Model Sistem Steganografi

Agar menjadi lebih aman, data diacak terlebih dahulu menggunakan *Vigenere Cipher*, kemudian baru dilakukan proses

steganografi agar lebih maksimal dalam mengamankan dan menjaga kerahasiaan. Steganografi membutuhkan dua properti, yaitu data dan wadah penampung data. wadah penampung yang umumnya digunakan berupa teks, suara, gambar, atau video. Sedangkan data yang disembunyikan dapat berupa teks, gambar, atau data yang lainnya. Keuntungan menggunakan steganografi adalah memungkinkan pengiriman pesan secara rahasia tanpa diketahui bahwa pesan sedang dikirim karena pesan tersembunyi. Ini membuat pihak ketiga tidak menyadari keberadaan pesan. Sebaliknya, penggunaan *Vigenere Cipher* akan menarik kecurigaan pihak ketiga bahwa ada sesuatu yang disembunyikan dalam pesan yang sedang dikirim. Adapun *flowchart* Steganografi *Least significant* dalam melakukan penyisipan *Hidden* Teks dapat dilihat pada gambar 2 sebagai berikut:

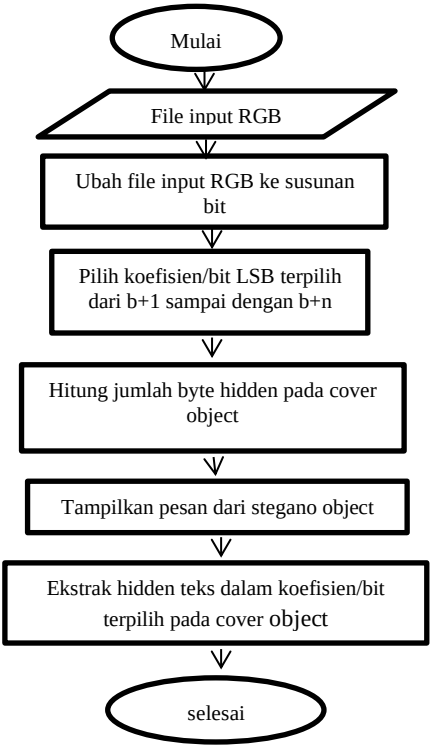


Gambar 2. *Flowchart* Proses Embeded Metode LSB

Pada gambar 2 di atas merupakan diagram alir dari proses embed atau penyisipan pesan ke dalam gambar atau media penampung menggunakan metode LSB. Langkah awal dimulai dengan input file. Kemudian file input RGB tersebut diubah ke dalam bentuk susunan bit. Selanjutnya menghitung bit terkecil (LSB) pada gambar cover. Proses selanjutnya adalah memilih koefisien bit LSB mulai dari b+1 sampai dengan b+n untuk disisipkan hidden text (pesan). Setelah itu

sisipkan hidden text ke dalam koefisien bit terpilih dan disisipkan kembali ke dalam gambar cover. Setelah itu transformasikan kembali ke dalam nilai RGB yang baru dan simpan gambar yang telah memiliki pesan di dalamnya sebagai gambar steganografi.

Adapun proses pengembalian gambar yang telah disisipkan data dapat dilihat pada gambar 3 di bawah ini :



Gambar 3. Proses Ekstaksi Citra Digital

Diagram alir di atas merupakan diagram alir dari proses ekstraksi menggunakan metode LSB. Langkah awal dimulai dari input file RGB. Selanjutnya mengubah file RGB ke dalam bentuk susunan bit (contoh: 11100100). Langkah selanjutnya memilih koefisien/bit LSB terpilih dari b+1 sampai dengan b+n. Selanjutnya menghitung jumlah byte hidden text pada gambar cover. Proses selanjutnya adalah melakukan

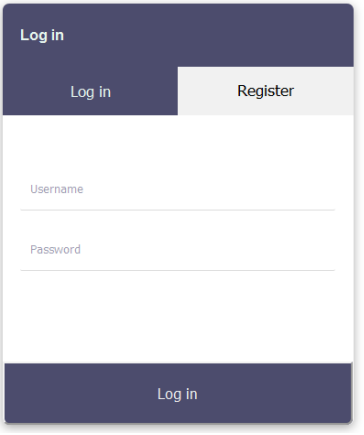
ekstrak hidden text bit terpilih dalam gambar object. Maka akan menghasilkan pesan yang sebelumnya disisipkan pada gambar cover.

3. Hasil dan Pembahasan

Berisikan tampilan eksekusi atau implementasi yang di buat. Berikut ini tampilan hasil dari implementasi.

1. *Form Login*

   Tampilan ini muncul pada saat pertama sekali program dijalankan. Tampilan *Login* dapat dilihat pada gambar 4

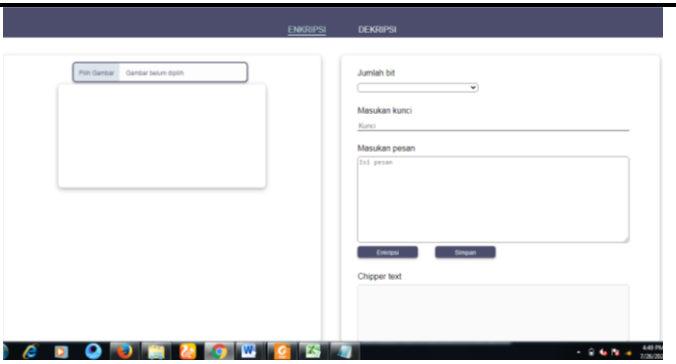


Gambar 4. *From Login*

Untuk bisa menjalankan aplikasi, terlebih dahulu pengguna harus masukan *Username* dan *Password* harus sesuai data yang ada di *database* sistem. Jika pengguna salah memasukan *Username* dan *Password,* maka akan muncul pesan kesalahan pada sistem.

2. *Form* Enkripsi

   Form ini ditunjukan untuk mengenkripsi pesan yang akan disisipkan. Tampilan form Enkripsi dapat dilihat seperti gambar 5 di bawah ini :



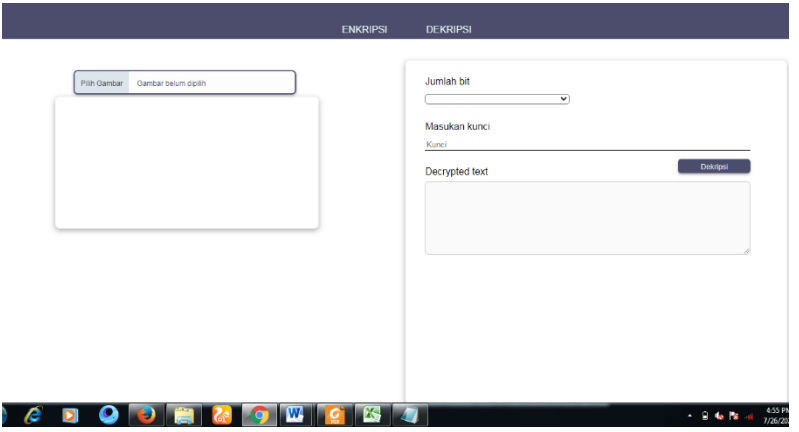
Gambar 5. Form Enkripsi

Gambar diatas adalah tampilan form enkripsi. Berikut ini penjelasan beberapa menu yang digunakan peneliti dari tampilan enkripsi :

a. *Button* pilih Gambar adalah sebuah tombol yang digunakan untuk memilih gambar yang akan disisipkan pesan rahasia.
b. *Combo Box* jumlah Bit merupakan kotak daftar yang berisikan Bit yang akan dipilih untuk menyisipkan pesan rahasia.
c. *Text Box* Kunci adalah sebuah kotak teks yang digunakan untuk memasukan kunci yang akan digunakan untuk mengenkripsi teks.
d. *Text Box* Pesan merupakan kotak teks yang digunakan untuk memasukan pesan yang akan di enkripsi.
e. *Button Enkripsi* adalah tombol yang berisikan perintah untuk mengenkripsi pesan.
f. *Button* Simpan merupakan sebuah tombol yang digunakan untuk menyimpan gambar yang berisikan pesan yang telah di enkripsi.
g. Box Chipper Text digunakan untuk menampilkan pesan yang telah di rahasiakan.

3. *Form* Dekripsi

Form ini ditunjukan untuk mendekripsikan pesan yang akan disisipkan. Tampilan form Dekripsi dapat dilihat seperti gambar 6 di bawah ini :



Gambar 6. Form Dekripsi

Gambar diatas adalah tampilan form enkripsi. Berikut ini penjelasan beberapa menu yang digunakan peneliti dari tampilan enkripsi :

a. *Button* pilih Gambar adalah tombol yang digunakan untuk memilih gambar yang telah disisipkan pesan rahasia.
b. *Combo Box* jumlah Bit merupakan kotak daftar yang berisikan Bit yang akan dipilih untuk mendeskripsikan pesan rahasia.
c. *Text Box* Kunci adalah sebuah kotak teks yang digunakan untuk memasukan kunci yang akan digunakan untuk mendekripsikan teks.
d. *Box Decrypted Text* digunakan untuk menampilkan pesan rahasia yang telah di ubah kedalam plaintext (teks asli).

Pengolahan Data

Misalkan pesan yang akan disisipkan adalah ” Candro##” Tambahkan ##### sebagai delimiter pada pesan. Alasan, agar tidak semua piksel harus dibaca, sedangkan kunci yang digunakan adalah ”key” maka Generate kunci adalah “keykeyke”, maka proses enkripsi pesan dengan menggunakan metode *Vigenere Cipher* :

1. Karakter Pesan ke-1 = 'C'
   P(1) = kode ascii dari 'C' = 67
   Kunci = 'k'
   K(1) = kode ascii dari ‘k’ = 107
   C(1) = (P(1) + K(1)) mod 128
   C(1) = (67 + 107) mod 128

C(1) = 46, diubah ke karakter = .

2. Karakter Pesan ke-2 = 'a'
   P(2) = kode ascii dari 'a' =97
   Kunci = 'e'
   K(2) = kode ascii dari ‘e’ = 101
   C(2) = (P(2) + K(2)) mod 128
   C(2) = (97 + 101) mod 128
   C(2) = 70, diubah ke karakter = F

3. Karakter Pesan ke-3 = 'n'
   P(3) = kode ascii dari 'n' = 110
   Kunci = 'y'
   K(3) = kode ascii dari ‘y’ = 121
   C(3) = (P(3) + K(3)) mod 128
   C(3) = (110 + 121) mod 128
   C(3) = 103, diubah ke karakter = g

4. Karakter Pesan ke-4 = 'd'
   P(4) = kode ascii dari 'd' = 100
   Kunci = 'k'
   K(4) = kode ascii dari ‘k’ = 107
   C(4) = (P(4) + K(4)) mod 128
   C(4) = (100 + 107) mod 128
   C(4) = 79, diubah ke karakter = O

5. Karakter Pesan ke-5 = 'r'
   P(5) = kode ascii dari 'r' =114
   Kunci = 'e'
   K(5) = kode ascii dari ‘e’ = 101
   C(5) = (P(5) + K(5)) mod 128
   C(5) = (114 + 101) mod 128
   C(5) = 87, diubah ke karakter = W

6. Karakter Pesan ke-6 = 'o'
   P(6) = kode ascii dari 'o' = 111
   Kunci = 'y'
   K(6) = kode ascii dari ‘y’ = 121

C(6) = (P(6) + K(6)) mod 128
C(6) = (111 + 121) mod 128
C(6) = 104, diubah ke karakter = h

7. Karakter Pesan ke-7 = '#'
   P(7) = kode ascii dari '#' = 35
   Kunci = 'k'
   K(7) = kode ascii dari ‘k’ = 107
   C(7) = (P(7) + K(7)) mod 128
   C(7) = (35 + 107) mod 128
   C(5) = 14, diubah ke karakter =

8. Karakter Pesan ke-8 = '#'
   P(8) = kode ascii dari '#' = 35
   Kunci = 'e'
   K(8) = kode ascii dari ‘e’ = 101
   C(8) = (P(8) + K(8)) mod 128
   C(8) = (35 + 101) mod 128
   C(8) = 8, diubah ke karakter =

Dengan demikian, hasil enkripsi dari pesan ”Candro##” dengan menggunakan kunci berupa ”key” adalah ” .,F,g,O,W,h, ,”. Selanjutnya, ubah setiap karakter dari pesan *cipher* ke biner, dan sisipkan bit-bit ke citra digital dengan menggunakan teknik LSB, sebagai berikut:

1. Ubah *ciphertext* ke biner.
   Karakter ke-1 = kode ascii = 46, diubah ke biner = 00101110
   Karakter ke-2 = kode ascii = 70, diubah ke biner = 01000110
   Karakter ke-3 = kode ascii = 103, diubah ke biner = 01100111
   Karakter ke-4 = kode ascii = 79, diubah ke biner = 01001111
   Karakter ke-5 = kode ascii = 87, diubah ke biner = 01010111
   Karakter ke-6 = kode ascii = 104, diubah ke biner = 00101001
   Karakter ke-7 = kode ascii = 14, diubah ke biner = 00001110
   Karakter ke-8 = kode ascii = 8, diubah ke biner = 00001000

2. Pilih citra yang akan disisipkan bit-bit biner dari ciphertext
   Citra ukuran 9x9 :

Tabel 1. Tabel Nilai Desimal Citra 9x9

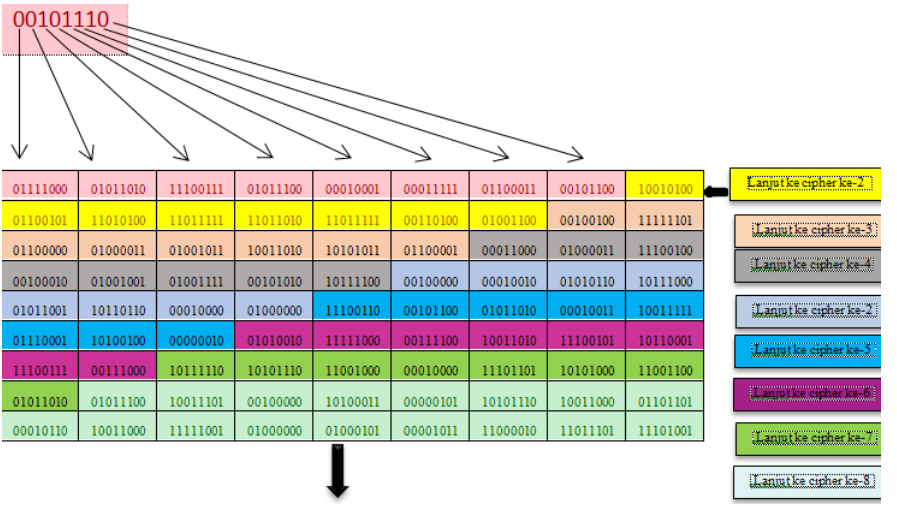
120	90	230	92	16	31	99	44	149
100	212	222	218	222	52	77	37	253
97	66	75	155	170	96	25	67	229
35	72	79	42	189	33	19	87	184
88	182	16	65	231	45	90	19	159
112	164	3	83	249	61	154	229	177
230	56	190	174	200	17	237	169	205
90	92	157	32	163	5	174	152	109
22	152	249	64	69	11	194	221	233

3. Konversi nilai setiap piksel citra ke dalam bentuk bilangan biner

Tabel 2. Tabel Nilai Biner Citra 9x9

01111000	01011010	11100110	01011100	00010000	00011111	01100011	00101100	10010101
01100100	11010100	11011110	11011010	11011110	00110100	01001101	00100101	11111101
01100001	01000010	01001011	10011011	10101010	01100000	00011001	01000011	11100101
00100011	01001000	01001111	00101010	10111101	00100001	00010011	01010111	10111000
01011000	10110110	00010000	01000001	11100111	00101101	01011010	00010011	10011111
01110000	10100100	00000011	01010011	11111001	00111101	10011010	11100101	10110001
11100110	00111000	10111110	10101110	11001000	00010001	11101101	10101001	11001101
01011010	01011100	10011101	00100000	10100011	00000101	10101110	10011000	01101101
00010110	10011000	11111001	01000000	01000101	00001011	11000010	11011101	11101001

4. Sisipkan bit-bit biner dari *ciphertext* secara berurutan pada bit paling terakhir (*Least Significant Bit* / LSB) dari citra, dimulai dari Karakter pertama dari cipher :



120	90	231	92	17	31	99	44	148
101	212	223	218	223	52	76	36	253
96	67	75	154	171	97	24	67	228
34	73	79	42	188	32	18	86	184
89	182	16	64	230	44	90	19	159
113	164	2	82	248	60	154	229	177
231	56	190	174	200	16	237	168	204
90	92	157	32	163	5	174	152	109
22	152	249	64	69	11	194	221	233

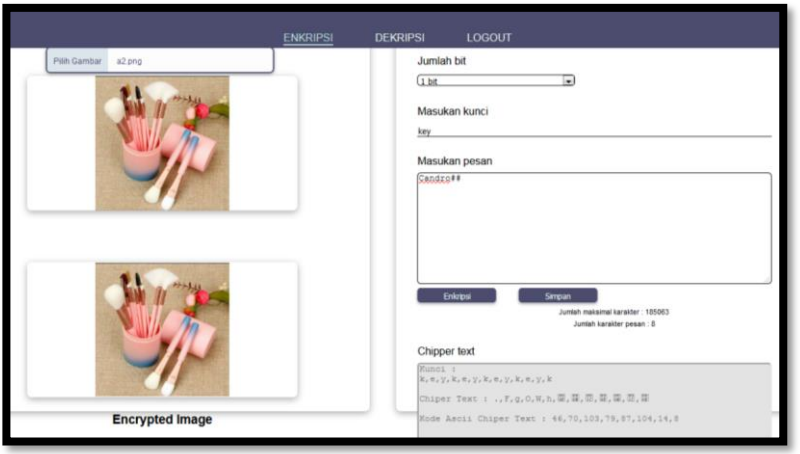
Gambar 7. Citra Output Metode LSB

Hasil Percobaan

Dalam pembahasan ini menjelaskan tentang penggunaan pengujian yang akan dilakukan penulis, baik itu menggunakan *software* atau aplikasi yang digunakan atau bentuk rangkaian-rangkaian serta

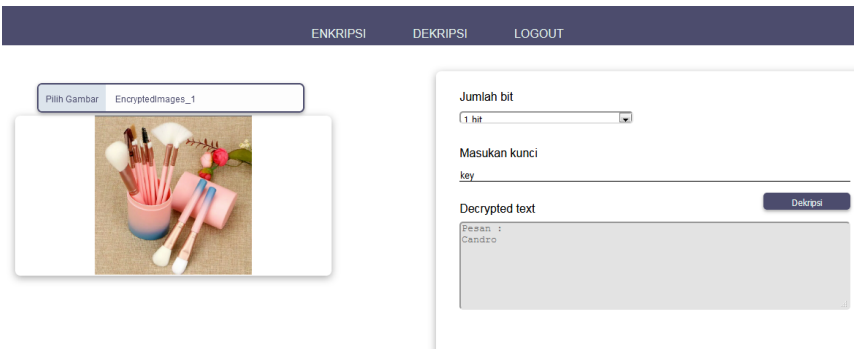
beberapa peralatan yang disesuaikan dengan bahan atau aplikasi yang digunakan dalam pengujian dari hasil penelitian tersebut.

Pengujian software



Gambar 8. Proses Enkripsi

Dalam tampilan form enkripsi di atas, kemampuan untuk memasukkan data ke dalam sebuah gambar pada citra penampung. Proses embedding dilakukan ketika data dipecah menjadi bit-bit citra dan ditransformasikan ke dalam citra yang menjadi wadah menggunakan metode least significant bit (LSB).



Gambar 9. Proses Dekripsi

Adapun untuk proses pengembalian gambar yang telah disisipi data untuk diambil data dapat dilihat pada gambar di atas yang

merupakan proses dekripsi dari citra yang telah dijadikan wadah dari data setelah steganografi. Adapun proses stegano atau penyisipan data ke dalam gambar dengan menggunakan perubahan data ke dalam bilangan biner.

4. Kesimpulan

Berdasarkan pembahasan sebelumnya dapat disimpulkan bahwa:

1. Algoritma Vigenere Cipher dan algoritma steganografi LSB dapat dijadikan sebagai metode untuk keamanan pesan rahasia yang disisipkan ke dalam image, dan dapat dekripsikan kembali sama persis dengan bentuk asli dan tidak hadapi kehancuran sedikitpun.
2. Penyisipan pesan tersembunyi berbentuk informasi bisa dicoba ke dalam wadah citra digital berformat PNG serta format citra digital yang lain, setelah itu bisa mengekstraksi kembali informasi tersembunyi tersebut dari dalam citra digital. Terjalin pergantian pada dimensi citra digital tetapi secara kasat mata perbandingan antara foto saat sebelum serta setelah disisipkan pesan tidak nampak. Tidak hanya itu waktu yang diperlukan buat proses enkripsi serta dekripsi dipengaruhi oleh kecepatan pc yang digunakan serta dimensi citra.

Daftar Pustaka

Abdurrazzaq Adrinta Muhammad. (2018). *Analisis Algoritma Modified Least Significant Bit Fungsi Polinomial untuk Pengamanan Citra Digital*. 1–91.

Adam Saputra, S. S. (2019). *Buku Sakti HTML, CSS & Javascript: Pemrograman Web Itu Gampang*. Anak Hebat Indonesia.

Gunawan, I. (2018). Penggunaan Algoritma Kriptografi Steganografi Least Significant Bit Untuk Pengamanan Pesan Teks dan Data Video. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, *2*(1), 57. https://doi.org/10.30645/j-sakti.v2i1.48

Hafiz, A. (2019). Steganografi Berbasis Citra Digital Untuk Menyembunyikan Data Menggunakan Metode Least Significant Bit (Lsb). *Jurnal Cendikia*, *17*(1), 194–198.

Hidayatullah, P. (2015). Pemrograman Web dengan HTML/CSS/JavaScript/XAMPP/PHP. *Penerbit: Informatika, Bandung*.

Janner, S., Sriadhi, & Robbi, R. (2019). *kriptografi* (M. Kika (ed.); 1st ed.). CV. ANDI OFFSET.

Keifer, G., & Effenberger, F. (2021). Inovasi Teknologi dan Produk Penelitian Pengabdian Masyarakat Berbasis Revolusi Industri 4.0 di Era New Normal. *Angewandte Chemie International Edition*, *6*(11), 951–952.

Komputer, W. (n.d.). *Panduan Praktis Menguasai Pemrograman Web dengan JavaScript 2009*. Penerbit Andi.

Nasution, S. D., Ginting, G. L., Syahrizal, M., & Rahim, R. (2017). Data Security Using Vigenere Cipher and Goldbach Codes Algorithm. *International Journal of Engineering Research & Technology (IJERT)*, *6*(01), 360–363.

Pardede, A. M. H. (2017). *Algoritma Vigenere Cipher Dan Hill Cipher Dalam Aplikasi Keamanan Data Pada File Dokumen*. *1*(1), 26–33. https://doi.org/10.31227/osf.io/7h36y

Saputra, H., Hadi, M. Z. S., & Syahroni, N. (2017). Implementasi Algoritma Steganografi Embedding Dengan Metode Least Significant Bit (Lsb) Insertion Dan Huffman Coding Pada Pengiriman Pesan Menggunakan Media Mms Berbasis J2Me. *Media*, 1–6.

Sianipar, R. H. (2015). *Pemrograman Javascript: Teori Dan Implementasi*. Penerbit INFORMATIKA.

Solichin, A. (n.d.). *Pemrograman Web dengan PHP dan MySQL*. Achmad Solichin.

Ummy Gusti Salamah, S. S. T. M. I. T., & Indonesia, M. S. (2021). *Tutorial Cascading Style Sheets (CSS)*. Media Sains Indonesia.