IMPLEMENTASI METODE KRIPTOGRAFI *STREAM CIPHER* GIFFORD UNTUK ENKRIPSI INTENSITAS WARNA PIKSEL PADA CITRA DIGITAL RAHASIA

Leliana Harahap[1)], Jonas Franky R Panggabean[2)] dan Sutrisno Situmorang[3)]

Akademi Informatika dan Komputer Medicom Medan
Jl. Darat No. 74, Medan 20153, Sumatera Utara, Indonesia

[1)]leliharahap05@gmail.com
[2)]jonasfrankypanggabean@gmail.com
[3)]risnositumorang4@gmail.com

Abstrak

Perkembangan citra digital yang pesat membuat citra juga membutuhkan aspek keamanan. Citra digital rahasia ini dapat dienkripsi dengan menggunakan metode kriptografi. Setelah dienkripsi, citra menjadi teracak, sehingga walaupun berhasil didapatkan oleh pihak yang tidak berwenang, citra tidak memiliki makna. Algoritma kriptografi digunakan dalam penelitian ini adalah metode Gifford. Metode Gifford merupakan *stream cipher*, yaitu algoritma enkripsi simetri yang mentransformasikan data secara karakter per karakter. Gifford memiliki 8 buah *register* yang diisi dengan bit-bit kunci. Proses-proses yang dilakukan metode Gifford adalah proses *Output Function*, proses *Sticky Shift Right* 1 bit, proses *Left Shift* 1 bit, operasi XOR dan operasi geser *register* ke kanan. Proses dekripsi harus menggunakan kunci yang sama dengan proses enkripsi agar dapat diperoleh citra semula. Aplikasi dapat digunakan untuk mengamankan citra digital yang bersifat rahasia dengan melakukan proses enkripsi terhadap nilai-nilai intensitas warna piksel di dalam citra menggunakan metode *Stream Cipher* Gifford.

Kata kunci: enkripsi, warna piksel, Gifford

Abstract

The rapid development of digital images, making images also requires security aspects. This secret digital image can be encrypted using cryptographic methods. After being encrypted, the image becomes scrambled, so that even if it was obtained by an unauthorized party, the image has no meaning. The cryptographic algorithm used in this study is the Gifford method. Gifford method is a stream cipher, a symmetric encryption algorithm that transforms data character by character. Gifford has 8 registers filled with key bits. The processes carried out by the Gifford method are the Output Function process, the Sticky Shift Right 1 bit process, the 1 bit Left Shift process, the XOR operation and the shift register operation to the right. The decryption process must use the same key as the encryption process in order to obtain the original image. The application can be used to secure confidential digital images by encrypting the pixel color intensity values in the image using the Gifford Stream Cipher method.

***Keywords**: encryption, pixel color, Gifford*

1. Pendahuluan

Salah satu aspek keamanan yang ditawarkan kriptografi adalah kerahasiaan data (*data confidentiality*). Perkembangan citra digital yang begitu pesat, membuat data berbentuk citra juga membutuhkan aspek keamanan. Contoh citra digital yang bersifat rahasia adalah citra medis dan citra privat pasien dalam dunia kedokteran, citra desain proyek, gedung, perkantoran dan kompleks yang menjadi karya dalam dunia teknik sipil. Citra-citra digital rahasia ini dapat dienkripsi dengan menggunakan metode kriptografi. Setelah dienkripsi, citra menjadi teracak, sehingga walaupun berhasil didapatkan oleh pihak yang tidak berwenang, citra tidak memiliki makna. Untuk mengembalikan citra ke bentuk semula, maka lakukan proses dekripsi dengan menggunakan kunci tertentu. Dengan demikian, hanya pihak-pihak tertentu saja yang dapat mengakses citra tersebut.

Salah satu algoritma *stream cipher* yang dapat digunakan untuk mengenkripsi citra adalah metode Gifford. Metode ini dikembangkan oleh David

Gifford. Dalam ilmu kriptografi, metode Gifford merupakan *stream cipher*, yaitu algoritma enkripsi simetri yang mentransformasikan data secara karakter per karakter. *Stream cipher* dapat dibuat sangat cepat sekali, jauh lebih cepat dibandingkan dengan algoritma *block cipher* yang manapun. Metode Gifford memiliki 8 buah *register*. *Register-register* ini memiliki nilai yang diisi dengan bit-bit kunci sebelum diproses selanjutnya. Proses-proses yang dilakukan metode Gifford dalam proses enkripsi dan dekripsi adalah *Output Function*, proses *Sticky Shift Right* 1 bit, proses *Left Shift* 1 bit, operasi XOR dan operasi geser *register* ke kanan. Oleh karena Gifford merupakan kriptografi kunci simetris, maka proses dekripsi harus menggunakan kunci yang sama dengan proses enkripsi, agar dapat diperoleh citra semula.

Dalam penulisan tugas akhir ini, akan dikembangkan sebuah aplikasi yang dapat melakukan enkripsi dan dekripsi terhadap citra dengan menggunakan metode Gifford. Dengan demikian, penulisan tugas akhir ini diberi judul "**Implementasi Metode Kriptografi *Stream Cipher* Gifford untuk Enkripsi Intensitas Warna Piksel pada Citra Digital Rahasia**".

2. Landasan Teori

2.1. Kriptografi

Kata kriptografi (*cryptography*) berasal dari bahasa Yunani, yaitu *kriptos* yang artinya *secret* (rahasia), dan *graphein*, yang artinya *writing* (tulisan). Jadi, kriptografi berarti *secret writing* (tulisan rahasia). Ada beberapa definisi kriptografi yang telah dikemukakan di dalam berbagai literatur. Definisi yang dipakai di dalam buku-buku yang lama (sebelum tahun 1980-an) menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Definisi ini mungkin cocok pada masa lalu di mana kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi di kalangan militer, diplomat dan mata-mata. Namun saat ini kriptografi lebih dari sekadar *privacy*, tetapi juga untuk tujuan *data integrity*, *authentication*, dan *non-repudiation*. [2]

Di dalam bukunya yang berjudul '*Handbook of Applied Cryptografphy*', Menezes Alfred, Paul van Oorschot dan Scott A. Vanstone mengartikan kriptografi sebagai ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasiaan, integritas data serta otentikasi. Sedangkan Bruce Schneier dalam bukunya '*Applied Cryptography*', mengartikan kriptografi sebagai ilmu dan seni untuk menjaga keamanan pesan. [2]

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Keamanan pesan diperoleh dengan menyandikannya menjadi pesan yang tidak mempuyai makna. Zaman sekarang, kerahasiaan informasi menjadi sesuatu yang penting. Informasi yang rahasia perlu disembunyikan agar tidak diketahui oleh orang yang tidak berhak. Seseorang tentu tidak ingin nomor PIN kartu kredit atau kartu ATM-nya diketahui orang. Atau, jika suatu pesan ditulis secara rahasia dan tidak ingin diketahui atau dibaca oleh orang lain. Kriptografi dapat digunakan untuk menyamarkan informasi yang bersifat rahasia dari orang atau pihak yang tidak berhak membacanya. [8]

Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu: [1]

1. Enkripsi: merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiaannya. Pesan asli disebut *plaintext*, yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan *cipher* atau kode atau sandi. Ketika seseorang tidak mengerti suatu kata maka orang tersebut akan mencarinya di dalam kamus atau daftar istilah. Beda halnya dengan enkripsi, untuk mengubah teks asli ke bentuk teks kode, digunakan algoritma yang dapat mengkodekan data yang diinginkan.
2. Dekripsi: merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (teks asli), disebut dengan dekripsi pesan.
3. Kunci: yang dimaksud di sini adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi.

Secara umum, operasi enkripsi dan dekripsi dapat dijelaskan secara matematis sebagai berikut:

$$EK(P) = C \quad \text{(Proses Enkripsi)}$$
$$DK(C) = P \quad \text{(Proses Dekripsi)}$$

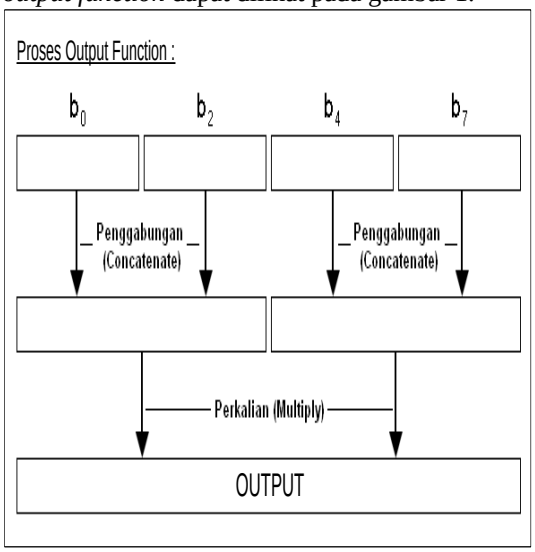
Pada saat proses enkripsi, pesan P (*plaintext*) disandikan dengan suatu kunci K (*key*), sehingga dihasilkan pesan C (*ciphertext*). Pesan C adalah pesan terenkripsi dan tidak dapat dibaca. Dalam hal ini, C berfungsi sebagai sandi rahasia yang hanya dapat dibaca oleh pihak-pihak yang berhak. Sedangkan pada proses dekripsi, pesan C tersebut disandikan kembali dengan menggunakan kunci K sehingga dihasilkan pesan P yang sama seperti pesan sebelumnya. Dengan demikian, keamanan suatu pesan tergantung pada kunci yang digunakan dan tidak tergantung pada algoritma yang digunakan. Tidak menjadi masalah apabila seseorang mengetahui algoritma yang kita gunakan. Selama ia tidak mengetahui kunci yang dipakai, ia tetap tidak dapat membaca pesan. [8]

2.2. Metode Gifford

David Gifford menemukan sebuah *stream cipher* dan menggunakannya untuk mengenkripsi surat kabel (*news wire report*) di Boston. Dalam kriptografi, Gifford merupakan *stream cipher*, yaitu algoritma enkripsi simetri yang mentransformasikan data secara karakter per karakter. Metode ini memiliki 8-*byte register*, yaitu: b_0, b_1, b_2, b_3, b_4, b_5, b_6 dan b_7. *Register-register* ini memiliki nilai atau kondisi awal sebelum diproses selanjutnya [12].

Proses ini harus dilakukan untuk membangkitkan bit-bit kunci yang akan digunakan pada proses enkripsi dan proses dekripsi. Proses-proses yang dilakukan metode Gifford dalam membangkitkan bit kunci adalah sebagai berikut,

1. Proses-1: Proses *Output Function*.
   Output dari proses ini adalah bilangan acak sepanjang 8 bit. Untuk menghasilkan sebuah bilangan acak, lakukan operasi penggabungan (*concatenate*) antara *register* b_0 dan *register* b_2 dan antara *register* b_4 dan *register* b_7. Kemudian, lakukan operasi perkalian antara kedua hasil penggabungan tersebut untuk menghasilkan bilangan 32 bit. *Byte* ketiga dari kiri merupakan *output* bilangan acak. Proses *output function* dapat dilihat pada gambar 1.



Gambar 1. Prosedur Kerja Algoritma Asimetris [12]

2. Proses ke-2: *Sticky Shift Right* 1 bit terhadap *register* b_1.
   Ambil isi *register* b_1 dan lakukan proses *sticky right shift* 1 bit terhadap b_1. Ini berarti menggeser bit (*shift*) ke kanan. Nilai *left most bit* dipertahankan dan bit kosong yang tergeser (bit yang berada di belakang *left most bit*) diganti dengan nilai *left most bit*. Hasilnya disimpan ke variabel A.
3. Proses ke-3: *Left Shift* 1 bit berhadap *register* b_7.
   Ambil isi *register* b_7 dan lakukan operasi *left shift* 1 bit. Ini berarti semua bit digeser (*shift*) ke kiri dan nilai *right most bit* diisi dengan nilai 0. Hasilnya disimpan ke variabel B.
4. Proses ke-4: Operasi XOR antara *register* b0, nilai variabel A dan variabel B
   Lakukan operasi XOR terhadap isi *register* b_0, nilai variabel A (hasil dari proses kedua) dan nilai variabel B (hasil dari proses ketiga). Ini berarti operasi akan menghasilkan nilai bit '0' (nol) untuk bit yang bernilai sama dan akan menghasilkan nilai bit '1' (satu) untuk bit yang berbeda .Hasilnya disimpan ke variabel C.
5. Proses ke-5: Geser blok *register* b ke kanan 1 blok
   Geser *register* b (b_0 ... b_7) 1 blok ke kanan. Ini berarti *register* b_7 dihilangkan (*discard*), *register* b_6 diisi oleh register b_5, *register* b_5 diisi oleh register b_4, *register* b_4 diisi oleh register b_3, *register* b_3 diisi oleh register b_2 dan *register* b_2 diisi oleh register b_1, sedangkan *register* b_0 yang kosong, diisi oleh nilai variabel C (hasil dari proses keempat).

Agar lebih jelas, perhatikan contoh berikut. Sebagai contoh, misalkan panjang *register* = 4 bit dan nilai awal dari masing-masing *register* adalah sebagai berikut:

1. $b_0 = 1001$.
2. $b_1 = 0011$.
3. $b_2 = 0001$.
4. $b_3 = 0110$.
5. $b_4 = 0000$.
6. $b_5 = 1000$.
7. $b_6 = 1110$.
8. $b_7 = 1111$.

Proses kerja yang dilakukan metode Gifford adalah sebagai berikut:

1. Untuk menghasilkan *output*-1, lakukan proses *output function* berikut.
   a. Gabungkan isi *register* antara b_0 dan b_2 dan antara b_4 dan b_7.
      Hasil penggabungan *register* b_0 dan b_2 = 10010011.
      Hasil penggabungan *register* b_4 dan b_7 = 00010110.
   b. Kalikan hasil penggabungan *register*.
      10010011 x 00010110
      = 0000100001111111.
   c. Didapat *Output* bit kunci ke-1 adalah byte *ketiga* dari kiri, yaitu: 00001000
2. *Sticky Right Shift* 1 bit pada *register* b_1.
   A = SSR(b_1)
   = SSR(0011)
   A = 0001
3. *Left Shift* 1 bit pada register b_7.
   B = LS(b_7)

= LS(1111)
B = 1110

4. Lakukan operasi XOR antara register b_0, nilai A dan nilai B.
   C = *register* b_0 XOR A XOR B
   = 1001 XOR 0001 XOR 1110
   C = 0110
5. Geser *register* ke kanan dengan melakukan langkah-langkah berikut,
   a. Buang isi *register* b_7.
   b. Geser isi *register* ke kanan.
      $b_7 = b_6 = 1110$.
      $b_6 = b_5 = 1000$.
      $b_5 = b_4 = 0000$.
      $b_4 = b_3 = 0110$.
      $b_3 = b_2 = 0001$.
      $b_2 = b_1 = 0011$.
      $b_1 = b_0 = 1001$.
   c. Isi *register* b_0 dengan nilai C.
      $b_0 = 0110$.

Untuk bit kunci pada proses berikutnya, lakukan proses yang sama. Pada proses pertama, didapatkan bit kunci ke-1 = 00001000

Proses enkripsi pada metode Gifford akan menggunakan bit kunci untuk mengubah *plaintext* menjadi *ciphertext*, dan sebaliknya pada proses dekripsi. Proses enkripsi dan dekripsi adalah berupa operasi XOR dari *plaintext* dan kunci untuk menghasilkan *ciphertext* atau operasi XOR *ciphertext* dan kunci untuk menghasilkan *plaintext* [12]

$$P = C \oplus K$$
$$C = P \oplus K$$

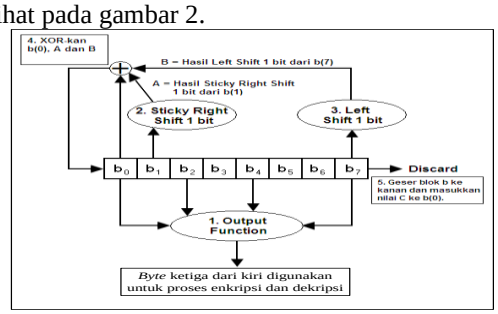
dengan :

P = *Plaintext*
K = *Key*
C = *Ciphertext*

3. Hasil dan Analisa

3.1. Pembentukan Bit Kunci

Pada tahapan ini, bit-bit kunci dihasilkan dari kunci *user* dan digunakan pada proses enkripsi atau dekripsi. Skema pembentukan bit kunci dengan menggunakan metode *stream cipher* Gifford dapat dilihat pada gambar 2.



Gambar 2. Pembentukan Bit Kunci Gifford

Sebagai contoh, apabila kunci yang digunakan adalah 'Kunci123' dan hendak dibangkitkan 10 buah sub kunci, maka proses pembentukan kunci dari sub kunci-1 hingga sub kunci ke-10 adalah sebagai berikut:

Nilai awal *register* b(0) sampai b(7):

Input kunci = 'Kunci123'
b(0) = biner (*ascii* dari huruf 'K') = biner(75) = 01001011
b(1) = biner (*ascii* dari huruf 'u') = biner(117) = 01110101
b(2) = biner (*ascii* dari huruf 'n') = biner(110) = 01101110
b(3) = biner (*ascii* dari huruf 'c') = biner(99) = 01100011
b(4) = biner (*ascii* dari huruf 'i') = biner(105) = 01101001
b(5) = biner (*ascii* dari huruf '1') = biner(49) = 00110001
b(6) = biner (*ascii* dari huruf '2') = biner(50) = 00110010
b(7) = biner (*ascii* dari huruf '3') = biner(51) = 00110011

Sub kunci ke-1

1. *Output Function*
   a. Gabungkan isi b(0) dengan b(2), dan b(4) dengan b(7).
      b(0) dan b(2) = 0100101101101110
      b(4) dan b(7) = 0110100100110011
   b. Kalikan hasil penggabungan *register*:
      0100101101101110 x 0110100100110011
      = 00011110111111110010010011101010
   c. Hasil sub kunci ke-1 = 00100100 (*byte* ketiga dari kiri)
      = 24 (heksa)
2. *Sticky shift right* 1 bit pada b(1)
   A = SSR(b(1))
   = SSR(01110101)
   = 00111010
3. *Left shift* 1 bit pada b(7)
   B = LS(b(7))
   = LS(00110011)
   = 01100110
4. Operasi XOR b(0), A dan B
   C = b(0) XOR A XOR B
   = 01001011 XOR 00111010 XOR 01100110
   C = 00010111
5. *Update register* b dengan menggeser blok b ke kanan.
   a. Buang isi *register* b(7).
   b. Geser ke kanan blok b
      b(7) = b(6) = 00110010
      b(6) = b(5) = 00110001
      b(5) = b(4) = 01101001
      b(4) = b(3) = 01100011
      b(3) = b(2) = 01101110
      b(2) = b(1) = 01110101

b(1) = b(0) = 01001011

c. Isi b(0) dengan nilai C.
b(0) = 00010111

Sub kunci ke-2

1. *Output Function*
   a. Gabungkan isi b(0) dengan b(2), dan b(4) dengan b(7).
      b(0) dan b(2) = 0001011101110101
      b(4) dan b(7) = 0110001100110010
   b. Kalikan hasil penggabungan *register*:
      0001011101110101 x 0110001100110010
      = 00001001000101101101001111011010
   c. Hasil sub kunci ke-2 = 11010011 (*byte* ketiga dari kiri)
      = D3 (heksa)
2. *Sticky shift right* 1 bit pada b(1)
   A = SSR(b(1))
   = SSR(01001011)
   = 00100101
3. *Left shift* 1 bit pada b(7)
   B = LS(b(7))
   = LS(00110010)
   = 01100100
4. Operasi XOR b(0), A dan B
   C = b(0) XOR A XOR B
   = 00010111 XOR 00100101 XOR 01100100
   C = 01010110
5. *Update register* b dengan menggeser blok b ke kanan.
   a. Buang isi *register* b(7).
   b. Geser ke kanan blok b
      b(7) = b(6) = 00110001
      b(6) = b(5) = 01101001
      b(5) = b(4) = 01100011
      b(4) = b(3) = 01101110
      b(3) = b(2) = 01110101
      b(2) = b(1) = 01001011
      b(1) = b(0) = 00010111
   c. Isi b(0) dengan nilai C.
      b(0) = 01010110

Lakukan perhitungan yang sama untuk sub kunci lainnya.

3.2. Proses Enkripsi Citra

Proses enkripsi dari *stream cipher* Gifford akan menggunakan bit kunci untuk mengubah citra asli (*plain-image*) menjadi citra terenkripsi (*cipher-image*). Proses enkripsi adalah berupa operasi XOR dari *plain-image* dan sub kunci untuk menghasilkan *cipher-image*.

$$P = C \oplus K$$

dengan :

P = *Plain-image*
K = *Key*
C = *Cipher-image*

Citra terdiri dari piksel, dan masing-masing piksel terdiri dari 3 unsur warna, yaitu *Red* (merah), *Green* (hijau) dan *Blue* (biru). Masing-masing komponen warna memiliki intensitas nilai dari 0 sampai 255, atau 8 bit biner. Pada proses enkripsi citra, setiap 8 bit komponen warna dari piksel akan di-XOR dengan 8 bit sub kunci, diambil dari sub kunci yang paling atas.

Sebagai contoh, misalkan piksel-1 yang akan dienkripsi memiliki nilai warna sebagai berikut:

Piksel-1: R = 228 = E4 (heksa)
G = 54 = 36 (heksa)
B = 106 = 6A (heksa)

Dengan menggunakan sub kunci-1, 2 dan 3 yang bernilai 24 D3 BE (heksa), maka proses enkripsi dari piksel pertama adalah sebagai berikut:

Hasil enkripsi piksel-1 :

R (baru) = E4 (heksa) xor 24 (heksa)
= C0 (heksa)
= 192 (desimal)
G (baru) = 36 (heksa) xor D3 (heksa)
= E5 (heksa)
= 229 (desimal)
B (baru) = 6A (heksa) xor BE (heksa)
= D4 (heksa)
= 212 (desimal)

Misalkan piksel ke-2 memiliki nilai warna sebagai berikut:

Piksel-2: R = 203 = CB (heksa)
G = 35 = 23 (heksa)
B = 178 = B2 (heksa)

Dengan menggunakan sisa sub kunci-4, 5 dan 6 yang bernilai 63 E5 16 (heksa), maka proses enkripsi dari piksel-2 adalah:

Hasil enkripsi piksel-2 :

R (baru) = CB (heksa) xor 63 (heksa)
= A8 (heksa)
= 168 (desimal)
G (baru) = 23 (heksa) xor E5 (heksa)
= C6 (heksa)
= 198 (desimal)
B (baru) = B2 (heksa) xor 16 (heksa)
= A4 (heksa)
= 164 (desimal)

Lakukan pross perhitungan yang sama untuk semua piksel sehingga nilai warna citra berubah.

3.3. Proses Dekripsi Citra

Proses dekripsi citra sama seperti proses enkripsi citra, yaitu dengan menggunakan fungsi XOR dari *cipher-image* dengan sub kunci untuk menghasilkan *plain-image*. Fungsi dari proses dekripsi dapat dinyatakan sebagai berikut:

$$C = P \oplus K$$

\dengan :

C = *Cipher-image*
K = *Key*
P = *Plain-image*

Sebagai contoh, misalkan piksel-1 yang akan didekripsi memiliki nilai warna sebagai berikut:

Piksel-1: R = 192 = C0 (heksa)
G = 229 = E5 (heksa)
B = 212 = D4 (heksa)

Dengan menggunakan sub kunci-1, 2 dan 3 yang bernilai 24 D3 BE (heksa), maka proses dekripsi dari piksel pertama adalah sebagai berikut:

Hasil dekripsi piksel-1 :

R (baru) = C0 (heksa) xor 24 (heksa)
= E4 (heksa) atau 228 (desimal)
G (baru) = E5 (heksa) xor D3 (heksa)
= 36 (heksa) atau 54 (desimal)
B (baru) = D4 (heksa) xor BE (heksa)
= 6A (heksa) atau 106 (desimal)

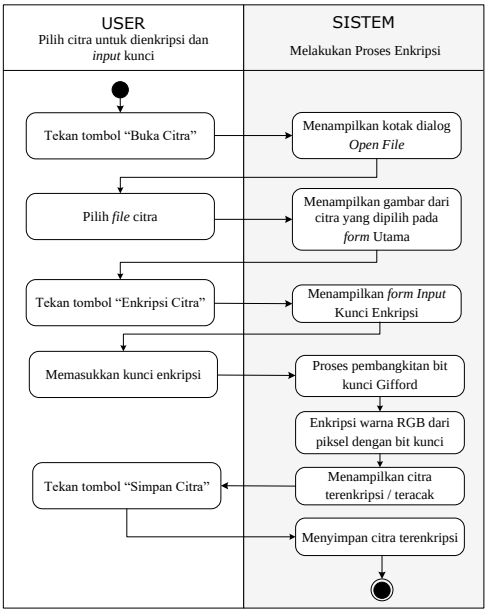
Misalkan piksel ke-2 memiliki nilai warna sebagai berikut:

Piksel-2: R = 168 = A8 (heksa)
G = 198 = C6 (heksa)
B = 164 = A4 (heksa)

Lakukan pross perhitungan yang sama untuk semua piksel sehingga nilai warna citra berubah.

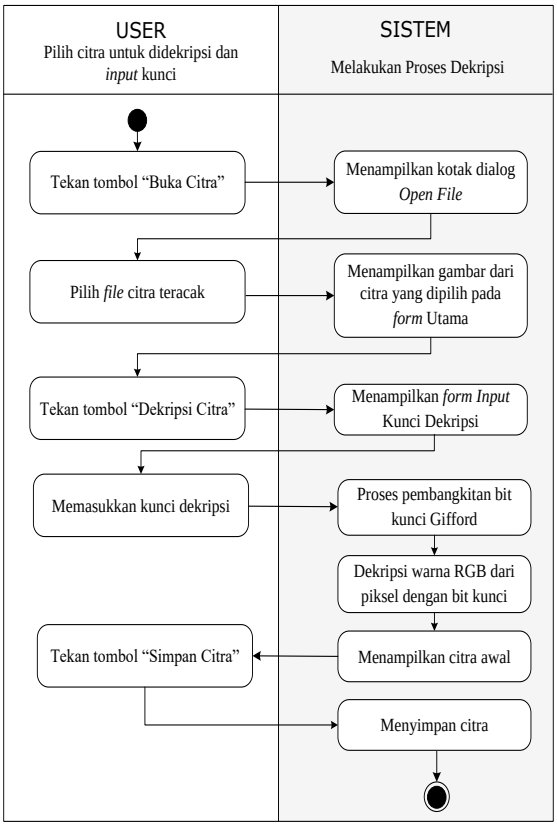
3.4 Pemodelan Sistem

Pemodelan sistem dapat dilakukan dengan menggunakan *Unified Modeling Language* (*UML*). Diagram *UML* menjelaskan mengenai tingkah laku sistem dan bukan bagaimana sistem bekerja. *Activity diagram* adalah salah satu diagram *UML*. Proses enkripsi citra yang terjadi di dalam aplikasi dapat digambarkan dengan *activity diagram* seperti terlihat pada gambar 3.



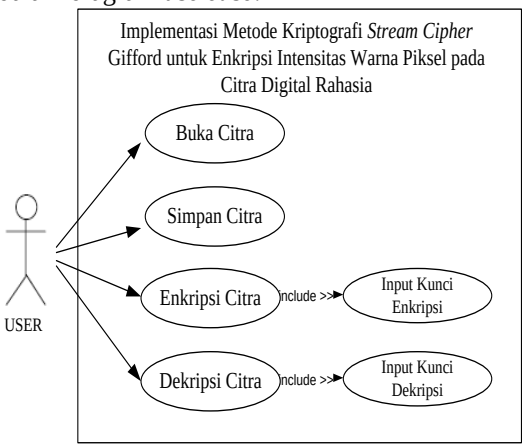
Gambar 3. *Activity Diagram* dari Proses Enkripsi

Dekripsi adalah kebalikan dari proses enkripsi. Proses dekripsi citra yang terjadi di dalam aplikasi dapat digambarkan dengan *activity diagram* seperti terlihat pada gambar 4.



Gambar 4. *Activity Diagram* dari Proses Dekripsi

Use case adalah salah satu diagram *Unified Modeling Language* (*UML*) yang dapat digunakan untuk menganalisis dan memodelkan sistem. Gambar 5 menunjukkan interaksi antara pengguna dan sistem di dalam diagram *use case*.



Gambar 5. Diagram *Use Case* dari Aplikasi

Dalam notasi *Use Case*, hubungan "*include*" antar *use case*, berarti *use case* X menggunakan *use case* Y sepenuhnya.

4. Kesimpulan

Setelah menyelesaikan perancangan aplikasi implementasi metode kriptografi s*tream cipher* Gifford untuk enkripsi intensitas warna piksel pada citra digital rahasia, beberapa hal yang dapat disimpulkan adalah sebagai berikut:

1. Aplikasi dapat digunakan untuk mengamankan citra digital yang bersifat rahasia dengan melakukan proses enkripsi terhadap nilai-nilai intensitas warna piksel di dalam citra menggunakan metode *Stream Cipher* Gifford.
2. Aplikasi tidak dapat mendekripsi citra kembali ke citra awal apabila kunci yang digunakan berbeda dengan kunci enkripsi.
3. Apabila citra hasil enkripsi didistorsi atau dirusak, maka bagian citra yang tidak terdistorsi akan tetap kembali ke bentuk citra awal.
4. Semakin besar ukuran citra atau semakin besar jumlah piksel yang dimiliki oleh suatu citra, maka semakin lama proses enkripsi dan dekripsi terhadap citra tersebut.

5. Ucapan Terima Kasih

Selama penyusunan Skripsi dengan judul "**IMPLEMENTASI METODE KRIPTOGRAFI *STREAM CIPHER* GIFFORD UNTUK ENKRIPSI INTENSITAS WARNA PIKSEL PADA CITRA DIGITAL RAHASIA**", penulis banyak mendapatkan bantuan dan bimbingan yang bermanfaat, maka pada kesempatan ini penulis mengucapkan banyak terimakasih kepada semua pihak, khususnya kepada:

1. Bapak Kamson Sirait, S.T. M.Kom selaku Direktur AMIK Medicom, yang telah dengan sangat sabar dan tulus memberikan movitasi, saran dan arahan dalam penyelesaian penelitian ini.
2. Ketua LPPM AMIK Medicom beserta seluruh jajarannya.
3. Bapak Jeremia Siregar, S.Kom. M.Kom selaku Dekan Fakultas Teknologi Industri di Institut Sains dan Teknologi TD. Pardede Sebagai mitra kami untuk menyelesaikan penelitian ini.
4. Seluruh Dosen dan Staff AMIK Medicom Medan.
5. Yang paling istimewa kepada kedua orang tua dan keluarga yang telah memberikan semangat, doa, material, dukungan kepada penulis selama mengerjakan penelitian ini.
6. Semua Mahasiswa/i di AMIK Medicom yang telah memberikan semangat dan dorongan.

Penulis menyadari ada banyak kekurangan baik dalam penyampaian bahasa maupun dalam hal penyajian sehingga saran dan kritik sangat diperlukan dalam mengembangkan isi penelitian ini. Semoga penelitian ini dapat bermanfaat bagi para pembaca. Atas perhatian dan kerja sama yang baik ini, penulis mengucapkan terima kasih.

Daftar Pustaka

[1] Ariyus, D., 2008, Pengantar Ilmu Kriptografi, Andi, Yogyakarta.

[2] Basri, 2016, Kriptografi Simetris dan Asimetris dalam Perspektif Keamanan Data dan Kompleksitas Komputasi, Universitas Al Asyariah Mandar, Sulawesi Barat.

[3] Chandra, 2016, Keamanan Data dengan Metode Kriptografi Kunci Publik, USU, Medan.

[4] Basuki, dkk, 2015, Pengolahan Citra Digital Menggunakan *Visual Basic*, Graha Ilmu, Yogyakarta.

[5] Kadir, A. dan Susanto, A., 2013, Teori dan Aplikasi Pengolahan Citra, Andi, Yogyakarta.

[6] Kristina, M., 2012, Penerapan Metode Primavista Bagi Mahasiswa Praktek Instrumen Mayor (PIM) VI Piano di Jurusan Pendidikan Seni Musik, UN Yogyakarta, Yogyakarta.

[7] Kurniawan, 2014, Kriptografi, Keamanan Internet dan Jaringan Komunikasi, Informatika, Bandung.

[8] Munir, R., 2012, Matematika Diskrit, Revisi Kelima, Penerbit Informatika, Bandung.

[9] Munir, R., 2016, Kriptografi, Penerbit Informatika, Bandung.
[10] Omar, A., 2014, Penerapan Model Pembelajaran Kooperatif Tipe *Think Talk Write* untuk Meningkatkan Hasil Belajar Siswa, UIN Suska Riau, Pekanbaru.
[11] Putra, D., 2010, Pengolahan Citra Digital, Penerbit Andi, Yogyakarta.
[12] Schneier, B., 2016, *Applied Crytography*, John Willey and Sons Inc.
[13] Sutoyo, T., dkk, 2009, Teori Pengolahan Citra Digital, Penerbit Andi, Yogyakarta.