Jurnal Ilmiah Matematika dan Pendidikan Matematika (JMP)

Vol. 10 No. 2, Desember 2018

ISSN (Cetak) : 2085-1456; ISSN (Online) : 2550-0422

JURUSAN MATEMATIKA

FMIPA Universitas Jenderal Soedirman

Dewan Redaksi

Dewan Editor

Daftar Isi

Vol. 10 No. 2 Desember 2018

Isah Aisah, M. Suyudi, dan Tika Kartika, *Representasi Kode Kernel pada DNA* Hal. 1-8.

Nugroho Arif Sudibyo dan Siti Komsatun, *Pelabelan Total tak Reguler pada Beberapa Graf* Hal. 9-16.

Rusli Hidayat, Firdaus Ubaidillah, dan Hadi Siswanto, *Optimasi Proses Pengeringan Kopi di Pabrik Kopi PTPN XII Gumitir dengan Menggunakan Mason Dryer* Hal. 17-30.

Dyah Retno Kusumawardani, *Kemampuan Penalaran Berdasarkan Keyakinan Matematika dalam Pembelajaran PBL melalui Pendekatan Interaksi Dyadic*. Hal. 31-42.

Sri Utami Zuliana, *Penentuan Model Terbaik Regresi Ridge dan Terapannya*. Hal. 43-48.

Abduh Riski, Heri Purwantoro, dan Ahmad Kamsyakawuni, *Penyembunyian Chipertext Algoritma Gost pada Citra ke dalam Audio dengan Medote Least Significant Bit*. Hal. 49-62.

Ahmad Kamsyakawuni, Ahmad Husnan Fanani, dan Abduh Riski, *Pengamanan Citra dengan Algoritma Diffie-Hellman dan Algoritma Simplified Encryption Standard* (S-DES). Hal. 63-80.

Aloysius Joakim Fernandez, *Pembangkit Grup Persamaan Schrodinger*.

Hal. 81-92.

Sri Rahayu, Nurul Gusriani, dan Iin Irianingsih, *Zero-inflated Poisson Regression to Determine the Factors that Influence the Maternal Mortality Rate in Bandung 2016*. Hal. 93-106.

Jurnal Ilmiah Matematika dan Pendidikan Matematika (JMP)
Vol. 10 No. 2, Desember 2018, hal. 63-80
ISSN (Cetak) : 2085-1456; ISSN (*Online*) : 2550-0422

PENGAMANAN CITRA DENGAN ALGORITMA *DIFFIE-HELLMAN* DAN ALGORITMA *SIMPLIFIED DATA ENCRYPTION STANDARD* (S-DES)

Ahmad Kamsyakawuni

Mathematical Optimization and Computation Research Group,
Jurusan Matematika, FMIPA, Universitas Jember
kamsyakawuni.fmipa@unej.ac.id

Ahmad Husnan Fanani

Jurusan Matematika, FMIPA, Universitas Jember

Abduh Riski

Mathematical Optimization and Computation Research Group,
Jurusan Matematika, FMIPA, Universitas Jember

ABSTRACT. *Simplified Data Encryption Standard (S-DES) is a cryptographic algorithm whose data-disguise process is simple and fast enough compared to other algorithms. Because of its simplicity, the S-DES algorithm is vulnerable to statistical attacks when applied to imagery, so this study tries to minimize S-DES weaknesses in image data by modifying S-DES keys with Diffie-Hellman. Diffie-Hellman is one of the key generating algorithms and key exchange. This research uses image data that is RGB image and grayscale image. A modified S-DES key with Diffie-Hellman is then used to encrypt the image. This study also analyzed the security level of S-DES algorithm that the key has been modified with Diffie-Hellman.*

Keywords*: S-DES, Diffie-Hellman, Image encryption.*

ABSTRAK. *Simplified Data Encryption Standard* (S-DES) adalah algoritma pada kriptografi yang proses penyamaran datanya sederhana dan cukup cepat dibandingkan algoritma yang lain. Kesederhanaan algoritma S-DES rentan terhadap serangan statistik apabila diterapkan pada citra, sehingga penelitian ini mencoba meminimalisir kelemahan S-DES pada data citra dengan memodifikasi kunci S-DES dengan *Diffie-Hellman*. *Diffie-Hellman* adalah salah satu algoritma pembangkit kunci dan pertukaran kunci. Penelitian ini menggunakan data citra RGB dan data citra *grayscale*. Kunci S-DES yang sudah dimodifikasi dengan *Diffie-Hellman* kemudian digunakan untuk mengenkripsi citra. Penelitian ini juga menganalisis tingkat keamanan algoritma S-DES yang kuncinya sudah dimodifikasi dengan *Diffie-Hellman*.

Kata Kunci: S-DES, *Diffie-Hellman*, Enkripsi citra.

1. PENDAHULUAN

Perkembangan teknologi yang begitu pesat memungkinkan data dalam bentuk citra bersifat rahasia yang disimpan melalui komputer mudah untuk diretas

oleh pelaku kejahatan. Algoritma S-DES adalah salah satu algoritma untuk pengamanan data. Algoritma S-DES memiliki proses enkripsi yang sederhana dan cepat tetapi kurang aman apabila diterapkan pada data dalam bentuk citra. Berdasarkan penelitian Kumar (2014) algoritma S-DES rentan terhadap serangan statistik apabila digunakan pada data seperti citra dikarenakan kuncinya yang pendek dan seragam sehingga nilai piksel yang memiliki derajat keabuan yang sama apabila dienkripsi akan menghasilkan nilai yang sama (Kumar, 2014).

Algoritma *Diffie-Hellman* merupakan algoritma pertukaran kunci yang dapat digunakan untuk membangkitkan kunci yang akan digunakan oleh pengirim data tersandi dengan penerima data tersandi. Algoritma *Diffie-Hellman* tidak digunakan untuk melakukan poses enkripsi dan deskripsi, melainkan untuk mendapatkan kunci rahasia (*secret key*) pada proses enkripsi dan dekripsi. Kunci rahasia (*secret key*) dapat sebarkan secara bebas tanpa harus khawatir akan keamanannya. Kelebihan dari algoritma *Diffie-Hellman* yakni tidak terjadi proses pertukaran kunci enkripsi dan kunci dekripsi antara pengirim dan penerima pesan, sehingga seorang peretas akan kesulitan untuk mendapatkan kunci enkripsi maupun kunci dekripsi tersebut.

Penelitian awal terkait dengan algoritma S-DES yang diterapkan pada data berupa citra dilakukan oleh Hardjo (2016), yang melakukan penelitian mengenai enkripsi citra RGB (citra berwarna) menggunakan penggabungan algoritma S-DES dan algoritma *DNA-Vigenere Cipher*, serta beberapa perlakuan diantara pengenkripsian dua algoritma tersebut. Hasil yang didapat menyatakan penggabungan dua algoritma tersebut aman diterapkan pada citra RGB. Namun kelemahan dari penelitian tersebut, proses waktu enkripsi terlalu lama dikarenakan proses enkripsi dilakukan dua kali. *Plainimage* mula-mula dienkripsi menggunakan algoritma S-DES kemudian dienkripsi menggunakan algoritma *DNA-Vigenere Cipher*.

Penelitian selanjutnya, Sholeh (2017) mencoba memperbaiki kelemahan dari penelitian Hardjo (2016) yang membahas enkripsi citra *grayscale* dengan algoritma S-DES dan *Stream Cipher*. Hasil yang didapat menyatakan penggabungan dua algoritma tersebut aman diterapkan pada citra *grayscale*

dengan proses enkripsi yang lebih sederhana. Kunci yang digunakan untuk proses enkripsi dibangkitkan dari bilangan biner *random* 10 bit yang kemudian dienkripsi menggunakan algoritma *Stream Cipher*. Proses tersebut nantinya akan menghasilkan *key citra* yang akan dijadikan kunci untuk proses enkripsi menggunakan algoritma S-DES. Kelemahan dari penelitian tersebut terdapat pada proses dekripsi. Proses dekripsi menggunakan *key citra* dari proses enkripsi sebelumnya.

Penelitian ini dilakukan untuk meminimalisir kelemahan algoritma S-DES ketika diterapkan pada data yang berupa citra dengan memodifikasi kunci S-DES yang berupa citra menggunakan algoritma *Diffie-Hellman*. Kunci baru hasil modifikasi kemudian ditampilkan dalam bentuk citra baru yang disebut *secret key*. *Secret key* kemudian digunakan sebagai kunci S-DES baru untuk mengenkripsi citra.

2. METODE PENELITIAN

2.1. Algoritma *Diffie-Hellman*

Algoritma *Diffie-Hellman* diperkenalkan pertama kali oleh Whitfield Diffie dan Martin Hellman. Adapun langkah-langkah dalam melakukan pertukaran dan pembangkitan kunci menggunakan algoritma *Diffie-Hellman* adalah sebagai berikut.

a. Menentukan bilangan prima p, dan bilangan bulat tak nol yang kurang dari p yaitu g. Bilangan p dan g dapat diketahui oleh orang lain (bersifat publik).

b. Menentukan sebarang bilangan bulat positif x untuk pengirim pesan, dan y untuk penerima pesan. Bilangan x dan y tidak boleh diketahui oleh orang lain (*private key*).

c. Pengirim menghitung $A = g^x \bmod p$ dan penerima menghitung $B = g^y \bmod p$.

d. Bilangan A dan B merupakan *shared key* yang kemudian ditukarkan, A diberikan ke penerima dan B diberikan ke pengirim.

e. Pengirim menghitung $K_A = B^x \bmod p$ dan penerima menghitung $K_B = A^y \bmod p$.

Pengirim dan penerima mengetahui *secret key* yang nantinya akan digunakan sebagai kunci untuk proses enkripsi dan dekripsi pada algoritma simetri dalam kriptografi (Abbadi, dkk. 2016).

2.2. Algoritma *Simplified Data Encryption Standard* (S-DES)

Algoritma S-DES (*Simplified Data Encryption Standard*) dikenalkan pertama kali oleh Edward Schaefer dari Universitas Santa Clara. Algoritma S-DES merupakan penyederhanaan dari algoritma DES (*Data Encryption Standard*). Algoritma S-DES memiliki keunggulan jika dibandingkan dengan algoritma DES yaitu dari segi kecepatan proses enkripsi dan dekripsi serta kesederhanaannya karena algoritma S-DES hanya menggunakan 10-bit kunci untuk proses enkripsi dan dekripsinya.

Simplified Data Encryption Standard (S-DES) merupakan algoritma kriptografi yang pada proses enkripsi membutuhkan 8-bit *plaintext*, 10-bit kunci, dan menghasilkan 8-bit *ciphertext*, sedangkan untuk proses dekripsi membutuhkan 8-bit *ciphertext* dan 10-bit kunci dan menghasilkan 8-bit *plaintext*. Kunci S-DES 10-bit digunakan untuk membangkitkan subkunci 8-bit K_1 dan 8-bit K_2 (Garg, 2015).

2.3. Langkah-langkah Enkripsi dan Deskripsi

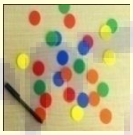
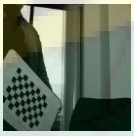
Pengenkripsian *plainimage* yang berupa citra *grayscale* dan citra RGB dengan algoritma S-DES, dilakukan dengan terlebih dahulu memodifikasi bagian kuncinya dengan algoritma *Diffie-Hellman*. Modifikasi dilakukan guna mengatasi kelemahan algoritma S-DES. Tabel 1 merupakan data yang digunakan dalam penelitian.

Berikut ini adalah langkah-langkah enkripsi dan deskripsi dari metode yang digunakan dalam penelitian ini.

a. Proses pembangkitan kunci

   Proses pembangkitan kunci dilakukan sesuai dengan alur pada Gambar 1 yang diuraikan sebagai berikut.

Tabel 1. Data Penelitian

No	*Plain image*	*Pubkey*1	*Pubkey*2	*PrivKey* (*Sender*)	*PrivKey* (*Receiver*)
1.	 (truck.png)	 (building.png)	 (holdingCup.png)	5	8
2.	 (*city*.png)	 (*image*05.png)	 (*onion*.png)	3	4
3.	 (*text*.png)	 (*paper*.png)	 (*peppers*.png)	2	6
4.	 (*textgray*.png)	 (*image*0150.png)	 (*left*05.png)	5	4
5.	 (*cameraman*.png)	 (*coins*.png)	 (*image*1.png)	6	7

1. Pembentukan nilai *p* dan *g* dari *public key*

   Nilai derajat keabuan pada setiap *pixel* dari *public key* akan dijadikan sebagai nilai *p* dan *g*. Ada beberapa perlakuan mengenai penentuan nilai *p* dan *g* dari dua buah citra tersebut. Berikut merupakan beberapa perlakuan dalam penentuan nilai *p* dan *g*. Misalkan terdapat dua buah citra yaitu citra A dan citra B yang akan dijadikan sebagai *public key*, maka

   a) Jika nilai derajat keabuan pada setiap *pixel* citra A atau citra B sama dengan nol maka nilai derajat keabuan *pixel* citra A atau citra B akan ditambah dua.

   b) Jika nilai derajat keabuan pada setiap *pixel* citra A sama dengan citra B maka nilai derajat keabuan *pixel*citra A akan ditambah dua.

c) Jika nilai derajat keabuan pada setiap *pixel* citra A lebih besar dari citra B maka nilai derajat keabuan *pixel* citra A akan dijadikan sebagai *p* dan nilai derajat keabuan *pixel* citra B akan dijadikan sebagai *g*, berlaku juga untuk sebaliknya.

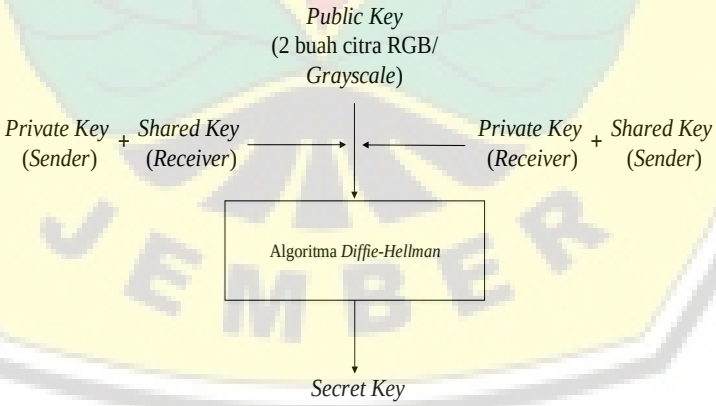
d) Nilai *p* haruslah prima sehingga apabila nilai derajat keabuan dari suatu *pixel* bukan prima maka akan dilakukan pembulatan ke atas ke prima terdekat.

2. Pembentukan *shared key*

Untuk mendapatkan *shared key*, setiap nilai *p* dan *g* dioperasikan dengan *private key* sesuai dengan algoritma *Diffie-Hellman*. Hasil dari proses pembentukan *shared key* berupa sebuah citra RGB atau citra *grayscale*, sesuai dengan citra *public key* yang diinputkan.

3. Pembentukan *secret key*

Setelah diperole nilai *p* dan *g* serta *shared key*, dilakukan proses pembangkitan kunci menggunakan algoritma *Diffie-Hellman*. Hasil dari proses pembangkitan kunci ini penulis sebut sebagai *secret key* berupa sebuah citra RGB atau citra *grayscale* (lihat Gambar 1).



Gambar 1. Proses Pembangkitan Kunci

b. Proses Enkripsi

Proses enkripsi dapat dilihat pada Gambar 2, sedangkan penjelasannya adalah sebagai berikut.

1. Konversi nilai *pixel* menjadi biner

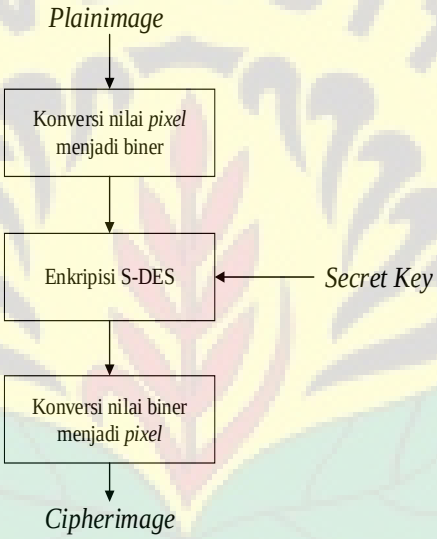
Nilai-nilai *pixel* pada *plainimage* diubah menjadi bilangan biner.

2. Enkripsi dengan algoritma S-DES

Nilai derajat keabuan pada setiap *pixel* dienkripsi dengan algoritma S-DES dengan kunci yang telah dibangkitkan sebelumnya menggunakan algoritma *Diffie-Hellman* (*secret key* pengirim).

3. Konversi nilai biner menjadi *pixel*

Setiap bilangan biner 8bit hasil dari proses enkripsi S-DES dikonversi menjadi bilangan desimal yang nantinya akan menjadi nilai derajat keabuan pada *pixel* dan merupakan hasil akhir dari proses enkripsi.



Gambar 2. Proses Enkripsi

c. Proses Dekripsi

Proses dekripsi dilakukan sesuai dengan proses yang ditunjukkan pada Gambar 3. Proses dekripsi dilakukan pada setiap *pixel cipherimage*.

1. Konversi nilai *pixel* menjadi biner,

Nilai-nilai *pixel* pada *cipherimage* diubah menjadi bilangan biner.

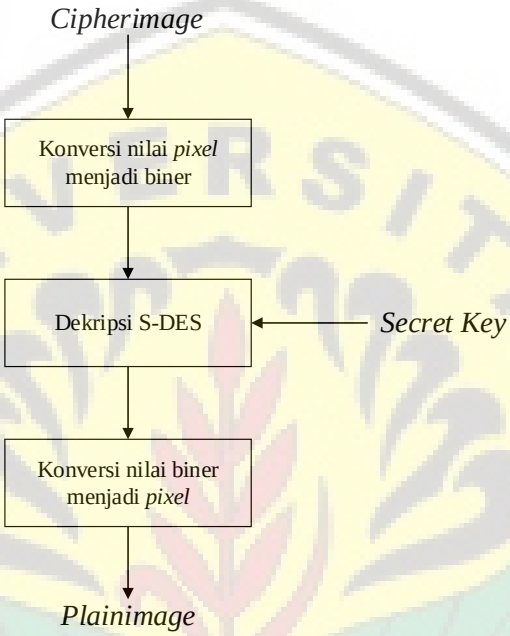
2. Dekripsi dengan algoritma S-DES

Nilai derajat keabuan pada setiap *pixel* didekripsi dengan algoritma S-DES dengan kunci yang telah dibangkitkan sebelumnya menggunakan algoritma *Diffie-Hellman* (*secret key* penerima)

.

3. Konversi nilai biner menjadi *pixel*

   Setiap bilangan biner 8 bit hasil dari proses dekripsi S-DES dikonversi menjadi bilangan desimal yang nantinya akan menjadi nilai derajat keabuan pada *pixel* dan merupakan hasil akhir dari proses dekripsi.



Gambar 3. Proses Dekripsi

3. HASIL DAN PEMBAHASAN

Pada bagian hasil dan pembahasan dijelaskan contoh proses perhitungan terhadap data kelima dimana citra yang digunakan merupakan citra *grayscale* yang dapat dilihat pada Tabel 1. Tabel 2 menunjukkan potongan *pixel*, tiga baris pertama dan tiga kolom pertama derajat keabuan pada *plainimage* "cameraman.png" yang digunakan.

Sesuai Gambar 2 dan Gambar 3, kunci yang digunakan untuk proses enkripsi dan dekripsi adalah *secret key*. *Secret key* dibangkitkan dari beberapa kunci input yakni *public key*, *private key*, dan *shared key* yang selanjutnya dioperasikan sesuai dengan algoritma *Diffie-Hellman*. Tabel 3 dan Tabel 4 menunjukkan potongan tiga baris pertama dan tiga kolom pertama derajat keabuanpada *public key* 1 dan *public key* 2 yang digunakan.

Tabel 2. Potongan derajat keabuan c*ameraman*

Pixel	1	2	3
1	158	158	157
2	158	157	157
3	158	157	157

Tabel 3. Potongan nilai derajat keabuan *Public key* 1

Pixel	1	2	3
1	49	49	49
2	47	48	49
3	49	49	48

Tabel 4. Potongan nilai derajat keabuan *Public key* 2

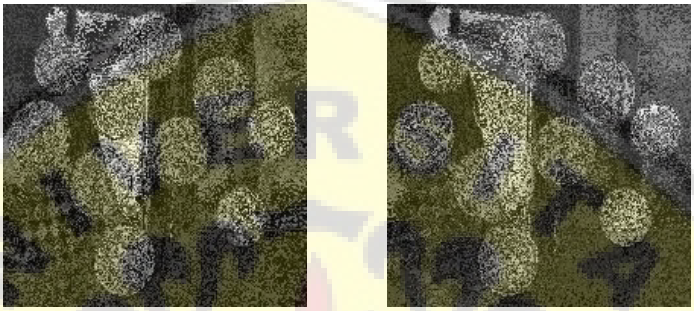
Pixel	1	2	3
1	24	23	18
2	33	23	18
3	46	22	13

Nilai *pixel* dari dua buah *public key* tersebut dijadikan sebagai nilai *p* dan *g* dalam algoritma *Diffie-Hellman* dengan mengikuti beberapa perlakuan yang telah dijelaskan sebelumnya. Tabel 5 merupakan hasil potongan nilai *pixel* dari proses penentuan nilai *p* dan *g*.

Private key 1 = 6 dan *private key* 2 = 7 adalah *private key* yang digunakan pada data kelima. *Private key* 1 ditentukan dari pengirim (*sender*) pesan tersandi dan *private key* 2 ditentukan dari penerima (*receiver*) pesan tersandi. Masing-masing dari pengirim dan penerima pesan tersandi, mengoperasikan setiap nilai *p* dan *g* dengan masing-masing *private key* sesuai algoritma *Diffie-Hellman* sehingga menghasilkan dua *shared key*, yaitu *shared key* pengirim dan *shared key* penerima pesan tersandi. Hasil dari seluruh perhitungan *shared key* kemudian ditampilkan dalam bentuk citra seperti pada Gambar 4. Tabel 6 dan Tabel 7 berturut-turut menunjukkan potongan nilai derajat keabuan tiga baris dan tiga kolom awal dari Gambar 4.

Tabel 5. Hasil proses penentuan nilai *p* dan *g*

Pixel	1		2		3	
	p	*g*	*p*	*g*	*p*	*g*
1	53	24	53	23	53	18
2	47	33	53	23	53	18
3	53	46	53	22	53	13



(a) *Shared key* pengirim (b) *Shared key* penerima

Gambar 4. *Shared key* data kelima

Tabel 6. Potongan derajat keabuan s*hared key* pengirim

Pixel	1	2	3
1	28	52	4
2	42	52	4
3	42	25	46

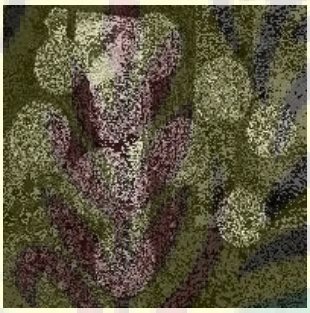
Tabel 7. Potongan derajat keabuan s*hared key* penerima

Pixel	1	2	3
1	36	30	19
2	23	30	19
3	24	20	15

Selanjutnya, dua buah citra *shared key* tersebut saling ditukarkan yakni citra *shared key* pengirim diberikan ke penerima pesan tersandi dan citra *shared key* penerima diberikan ke pengirim pesan tersandi.

Setelah pengirim dan penerima pesan tersandi melakukan pertukaran citra *shared key*, pengirim pesan melakukan proses pembentukan kunci enkripsi S-DES dan penerima pesan melakukan proses pembentukan kunci dekripsi S-DES, dimana kunci yang dimaksud adalah *secret key*. Proses pembentukan *secret key*

hampir sama dengan proses pembentukan *shared key*, yang membedakan adalah nilai *g* pada saat pembentukan *shared key* diganti dengan nilai derajat keabuan dari *shared key*. Sehingga untuk pembentukan kunci enkripsi S-DES nilai *g* diganti dengan nilai derajat keabuan *shared key* penerima, sedangkan untuk pembentukan kunci dekripsi S-DES nilai *g* diganti dengan nilai derajat keabuan *shared key* pengirim. Berdasarkan algoritma *Diffie-Hellman*, nilai *secret key* untuk pengirim pesan sama dengan nilai *secret key* pada penerima pesan. Hasil dari seluruh perhitungan pembentukan *secret key* kemudian ditampilkan dalam bentuk citra seperti pada Gambar 5. Tabel 8 merupakan potongan nilai derajat keabuan dari *secret key*.



Gambar 5. *Secret key* data kelima

Tabel 8. Potongan nilai derajat keabuan *secret key*

Pixel	1	2	3
1	44	52	7
2	36	52	7
3	28	9	24

Tabel 9. Pembangkitan sub kunci K_1 dan K_2

Pixel(1,1) (desimal)	44	
Pixel(1,1) (biner)	00001	01100
*P*10	01010	00010
LS-1	10100	00100
*P*8 (K_1)	**01001000**	
LS-2	10010	10000
*P*8 (K_2)	**10010000**	

Selanjutnya dilakukan proses enkripsi pada *plainimage* dengan algoritma S-DES. Pertama-tama dibangkitkan subkunci K_1 dan K_2 dari *secret key*. Tabel 9 menunjukkan proses pembangkitan subkunci K_1 dan K_2 dari *pixel*(1,1) pada Tabel 8.

Setelah didapatkan sub kunci K_1dan K_2 dilakukan proses enkripsi pada *plainimage*. Tabel 10 menunjukkan proses enkripsi S-DES pada *pixel*(1,1) dari *plainimage* cameraman.png. Tabel 11 adalah hasil dari proses enkripsi pada Tabel 2 dengan menggunakan kunci dari data Tabel 8.

Tabel 10. Proses dan hasil enkripsi pada *pixel* (1,1)

Plainimage	10011110 (158)
IP	01011011
f_1	10101011
SW	10111010
f_2	01111010
IP^{-1}	10111100
Cipherimage	10111100 (188)

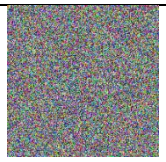
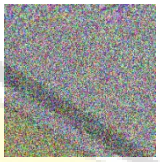
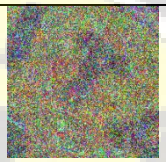
Tabel 11. Hasil enkripsi potongan *pixel cameraman*

Pixel	*Plainimage*			*Cipherimage*		
	1	2	3	1	2	3
1	158	158	157	188	17	62
2	158	157	157	188	47	62
3	158	157	157	145	168	100

Dengan menggunakan cara yang sama, didapatkan hasil enkripsi untuk masing-masing data penilitian. Tabel 12 adalah hasil enkripsi dari kelima data penelitian.

Proses dekripsi S-DES hampir sama dengan saat proses enkripsi, yang membedakan yakni fungsi f_k yang pertama menggunakan sub kunci K_2 dan fungsi f_k kedua menggunakn sub kunci K_1. Tabel 14 merupakan proses dan hasil dekripsi *cipherimage* data ke lima pada *pixel*(1,2) pada Tabel13 dengan menggunakan kunci dari data Tabel 8.

Tabel 12. Hasil enkripsi seluruh data penelitian

No.	*Plainimage*	*Cipherimage*
1.		
2.		
3.		
4.		
5.		

Tabel 13. Hasil konversi potongan *pixel cipherimage*

Pixel	Bilangan desimal			Bilangan biner		
	1	2	3	1	2	3
1	188	17	62	10111100	00010001	00111110
2	188	47	62	10111100	00101111	00111110
3	145	168	100	10010001	10101000	01100100

Tabel 14. Proses dan hasil dekripsi pada *pixel* (1,2)

Cipherimage	00010001 (17)
IP	00001100
f_2	10111100
SW	11001011
f_1	01011011
IP^{-1}	10111100
Plainimage	10111100 (158)

Tabel 15. Hasil dekripsi potongan *pixel cipherimage*

Pixel	*Cipherimage*			*Plainimage*		
	1	2	3	1	2	3
1	188	17	62	158	158	157
2	188	47	62	158	157	157
3	145	168	100	158	157	157

Setiap potongan *pixel* nilai derajat keabuan pada Tabel 13 dilakukan proses dekripsi S-DES menggunakan cara yang sama dengan Tabel 14 dimana kunci yang digunakan merupakan nilai derajat keabuan pada Tabel 8 untuk masing-masing *pixel*. Tabel 15 adalah potongan *pixel* nilai derajat keabuan hasil dekripsi S-DES *cipherimage* data kelima. Dari hasil proses dekripsi, dapat terlihat bahwa *cipherimage* dapat didekripsi dan menghasilkan citra awal tanpa adanya informasi yang hilang.

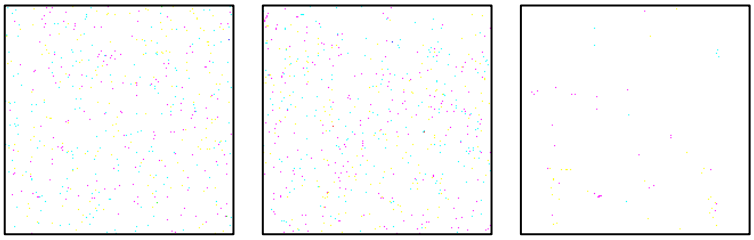
Analisis dengan diferensial

Nilai NPCR antara *plainimage* dengan *cipherimage* dari data-data hasil penelitian pada Tabel 12 yakni, untuk data ke-1 yaitu 99,5675%, untuk data ke-2 yaitu 99,5142%, untuk data ke-3 yaitu 99,9467%, untuk data ke-4 yaitu 99,98%, dan untuk data ke-5 yaitu 99,605%. Sedangkan Gambar 6 merupakan citra *NPCR* antara *plainimage* dengan *cipherimage* dari data hasil penelitian pada Tabel 12.

Nilai *NPCR* besar menunjukkan algoritma yang dipakai aman untuk diterapkan terhadap proses enkripsi *plainimage*. *NPCR* dengan nilai diatas 90% akan menyulitkan kriptanalisis dalam mencari hubungan statistik antara citra asli dengan citra terenkripsi (Akhavan, dkk. 2011). Terbukti bahwa nilai *NPCR* seluruh data penelitian bernilai diatas 99%, sehingga bisa dikatakan algoritma yang diajukan pada penelitian ini adalah aman untuk diterapkan.

Analisis Sensitivitas Kunci

Analisis sensitivitas kunci dengan kunci yang sedikit dirubah ditunjukkan pada Tabel 16. Pengujian analisis sensitivitas kunci dilakukan pada data penelitian ke lima.



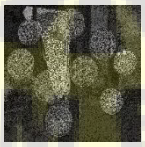
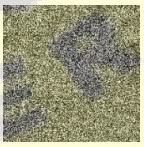
(a) (b) (c)

(d) (e)

(a) Citra NPCR data ke-1; (b) Citra NPCR data ke-2;
(c) Citra NPCR data ke-3; (d) Citra NPCR data ke-4;
(e) Citra NPCR data ke-5

Gambar 6. Analisis differensial seluruh data penelitian

Tabel 16. Analisis sensitivitas kunci

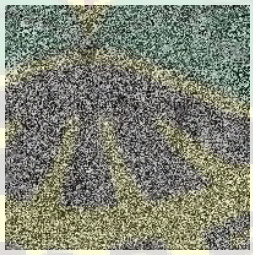
No.	*Private Key*	*Shared Key*	*Cipherimage*	NPCR (%)
1	*Privkey*1 = 6			95,925
2	*Privkey*1 = 5			97,805

Tabel 16 pada baris pertama menunjukkan nilai NPCR antara *cipherimage* data ke lima dengan *cipherimage* yang menggunakan kunci *private key* 1 = 6 dan *shared key* penerima seperti pada Gambar 5, dimana *shared key* penerima pada Gambar 7 adalah *shared key* yang dibentuk dari dua buah *public key* data ke lima dengan *private key* 2 = 8, sedangkan *shared key* penerima pada Gambar 4(b) dibentuk dari dua buah *public key*data ke limadengan *private key* 2 = 7. Baris kedua pada Tabel 16 menunjukkan nilai NPCR antara *cipherimage* data ke lima

dengan *cipherimage* yang menggunakan kunci *private key* 1 = 5 dan *shared key* penerima seperti pada Gambar 4(b). Gambar 8 merupakan citra hasil dekripsi *cipherimage* data ke lima dengan menggunakan kunci *private key* 1 = 4 dan *shared key* penerima pada Gambar 4(b).

Kunci dikatakan sensitif jika kunci yang sedikit berbeda diterapkan pada enkripsi *plainimage* menghasilkan *cipherimage* yang sangat berbeda. berdasarkan perhitungan yang dilakukan didapat nilai *NPCR* sebesar 95,925% untuk uji coba pertama dan 97,805% untuk uji coba kedua, sehingga bisa dikatakan algoritma yang diajukan memiliki kunci yang sensitif sehingga bisa dikatakan algoritma yang diajukan pada penelitian ini adalah aman untuk diterapkan.

Gambar 7. *Shared key* penerima (*private key* 2 = 8)



Gamabr 8. Citra hasil dekripsi dengan kunci berbeda

4. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, maka diperoleh beberapa kesimpulan sebagai berikut.

a. Algoritma *Diffie-Hellman* mampu digunakan untuk membangkitkan kunci dari beberapa kunci masukkan yang bersifat rahasia (*secret key*), yang dapat disebarkan tanpa harus khawatir akan keamanannya.

b. Proses enkripsi citra mampu memberikan keamanan karena modifikasi kunci yang berupa citra memungkinkan hasil enkripsi sebuah citra lebih bervariasi.
c. Proses dekripsi sudah sesuai yang diharapkan, dibuktikan dengan *cipherimage* yang dapat dikembalikan menjadi *plainimage* tanpa menghilangkan sedikit pun informasi awal.

Saran yang dapat diberikan untuk penelitian selanjutnya yaitu menerapkan metode penulis pada data yang lain seperti *text* dan juga pada *barcode*. Selain itu menggabungkan algoritma *Diffie-Hellman* dengan algoritma yang lain seperti AES, *Chaos Map*, dan *Mars*.

UCAPAN TERIMAKASIH

Penulis menyampaikan ucapan terima kasih kepada Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LP2M) Universitas Jember yang telah membiaya penelitian ini melalui Hibah Kelompok Riset (KeRis) Tahun 2018, dengan SK No.5462/UN25/LT/2018, tanggal 8 Mei 2018.

DAFTAR PUSTAKA

Abbadi, N. K. E., Abaas, S. T, dan Alaziz, A. A., *New Image Encryption Algorithm Based on Diffie-Hellman and Singular Value Decomposition*, Interna-tional Journal of Advanced Research in Computer and Communication Engineering **5**(1) (2016), 197-201.

Akhavan, A., Samsudin, A., dan Akhsani, A., *A Symmetric Image Encryption Scheme Based On Combination of Nonlinear Chaotic Maps*, Journal of the Franklin Institute, **348**(8) (2011), 1797-1813.

Garg, P., Varshney, S., dan Bhardwaj, M., *Crypt-analisis of Simplified Data Encryption Standard Us-ing Genetic Algorithm*, American Journal of Net-works and Communications, **4**(3) (2015), 32-36.

Kumar, S., dan Srivastava, S., *Image Encryption using Simplified Data Encryption Standard (S-DES)*, International Journal of Computer Application, **104**(2) (2014), 38-42.