PENERAPAN MODIFIKASI ALGORITMA *SIMPLIFIED DATA ENCRYPTION STANDARD* (S-DES) DENGAN *STREAM CIPHER* PADA CITRA *GRAYSCALE*

SKRIPSI

Oleh

Hairus Sholeh
NIM 131810101025

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017

PENERAPAN MODIFIKASI ALGORITMA *SIMPLIFIED DATA ENCRYPTION STANDARD* (S-DES) DENGAN *STREAM CIPHER* PADA CITRA *GRAYSCALE*

SKRIPSI

diajukan guna memenuhi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

Hairus Sholeh
NIM 131810101025

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2017

PERSEMBAHAN

Dengan menyebut nama Allah S.W.T yang maha pengasih lagi maha penyayang, serta sholawat atas Nabi Muhammad S.A.W, Saya persembahkan skripsi ini sebagai rasa syukur dan terima kasih saya kepada:

1. Allah S.W.T, karena hanya atas izin dan karunia-Nya skripsi ini dapat terselesaikan;
2. Orang tua saya, Bapak Abdul Muthallip dan Ibu Siti Maimunah, yang selalu memberi kepercayaan, dukungan moril maupun materiil serta do'a yang tiada henti untuk kesuksesan saya;
3. Kakak-kakakku tersayang;
4. Dosen Fakultas Matematika dan Ilmu Pengetahuan Alam;
5. Teman-teman angkatan 2013 (ATLAS), yang selalu berbagi semangat, masukan, dukungan, dan pengalaman berharga;
6. Almamater tercinta Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTTO

"*When it's hard to make up your mind, don't look too far ahead, just think about tomorrow, think about what you want to do tomorrow, and that will give you a different answer*".[1]

[1] Kim Won

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

nama : Hairus Sholeh

nim : 131810101025

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Penerapan Modifikasi Algoritma *Simplified-Data Encryption Standard* (S-DES) dengan *Stream Cipher* pada Citra *Grayscale*" adalah benar-benar hasil karya sendiri, kecuali kutipan yang telah disebutkan sumbernya, belum pernah diajukan di institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Juni 2017

Yang menyatakan,

Hairus Sholeh

NIM 131810101025

SKRIPSI

PENERAPAN MODIFIKASI ALGORITMA *SIMPLIFIED DATA ENCRYPTION STANDARD* (S-DES) DENGAN *STREAM CIPHER* PADA CITRA *GRAYSCALE*

Oleh

Hairus Sholeh
NIM 131810101025

Pembimbing:

Dosen Pembimbing Utama : Ahmad Kamsyakawuni, S.Si., M.Kom.

Dosen Pembimbing Anggota : Kusbudiono, S.Si., M.Si.

PENGESAHAN

Skripsi berjudul "Penerapan Modifikasi Algoritma *Simplified-Data Encryption Standard* (S-DES) dengan *Stream Cipher* pada Citra *Grayscale*" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Tim Penguji:

Ketua,	Anggota I,
Ahmad Kamsyakawuni, S.Si., M.Kom.	Kusbudiono, S.Si., M.Si.
NIP. 197211291998021001	NIP. 197704302005011001
Anggota II,	Anggota III,
Dr. Firdaus Ubaidillah, S.Si., M.Si.	Kosala Dwidja Purnomo, S.Si., M.Si.
NIP. 197006061998031003	NIP. 196908281998021001

Mengesahkan
Dekan,

Drs. Sujito, Ph.D.

NIP. 196102041987111001

RINGKASAN

Penerapan Modifikasi Algoritma *Simplified-Data Encryption Standard* (S-DES) dengan *Stream Cipher* pada Citra *Grayscale*; Hairus Sholeh, 131810101025; 2017; 54 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Globalisasi membawa dampak yang sangat besar, salah satunya yaitu mulai ditinggalkannya penyimpanan data secara cetak dan beralih ke penyimpanan data melalui komputer dan ponsel. Penyimpanan data melalui komputer dan ponsel ini berisiko karena kemajuan teknologi informasi mempermudah pelaku kejahatan menyalahgunakan data tersebut, sehingga terbentuklah metode yang disebut kriptografi.

Kriptografi adalah metode untuk melindungi data dengan cara mengubahnya menjadi bentuk yang tidak lagi dipahami maknanya. Algoritma kriptografi yang digunakan dalam penelitian ini adalah *Simplified-Data Encryption Standard* (S-DES). Algoritma S-DES ini memiliki beberapa keunggulan yaitu proses pengenkripsian dan pendekripsian datanya sederhana dan cepat karena kuncinya hanya berupa bilangan desimal berukuran 10-bit. Karena ukuran kunci dan prosesnya yang sederhana, S-DES memiliki kelemahan jika diterapkan pada data yang kompleks seperti citra karena rentan terhadap *statistical attack*. Untuk mengatasinya dilakukanlah modifikasi pada kunci S-DES dengan menggunakan metode *stream cipher*. Metode *stream cipher* adalah metode pengenkripsian yang dilakukan secara mengalir bit per bit.

Data yang digunakan dalam penelitian ini adalah citra *grayscale* yang digunakan sebagai *plain image*. Sebelum mengenkripsi citra *grayscale* dengan algoritma S-DES, terlebih dahulu mengenkripsi kunci S-DES dengan menggunakan metode *stream cipher* dimana *keystream*-nya menggunakan bilangan *random* yang dibangkitkan sebanyak ukuran dari citra *grayscale* yang menjadi *plain image*. Kemudian citra *grayscale* dienkripsi menggunakan kunci

hasil modifikasi kunci S-DES. Enkripsi ini menghasilkan sebuah *cipher image* yang tidak lagi mengandung informasi dari *plain image* yang ada.

Analisis keamanan diperlukan kaitannya dengan perlindungan data pada sebuah gambar. Dengan menggunakan analisis histogram derajat keabuan, analisis diferensial, dan analisis sensitivitas kunci, algoritma yang diajukan menunjukkan bahwa algoritma aman dari *statistical attack* dan memiliki kunci yang sensitif.

PRAKATA

Puji syukur ke hadirat Allah S.W.T atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul ”Penerapan Modifikasi Algoritma *Simplified-Data Encryption Standard* (S-DES) dengan *Stream Cipher* pada Citra *Grayscale*”. Skripsi ini disusun untuk memenuhi salah satu syarat pada program pendidikan strata satu (S1) Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Pada kesempatan ini penulis mengucapkan terima kasih atas bantuan dan bimbingan dalam penyusunan tugas akhir ini, terutama kepada yang terhormat:

1. Drs. Sujito, Ph.D., selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
2. Kusbudiono, S.Si., M.Si., selaku Ketua Jurusan Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
3. Ahmad Kamsyakawuni, S.Si., M.Kom., selaku Dosen Pembimbing Utama, Kusbudiono, S.Si., M.Si., selaku Dosen Pembimbing Anggota, Dr. Firdaus Ubaidillah, S.Si., M.Si., selaku Dosen Penguji I, Kosala Dwidja Purnomo, S.Si., M.Si., selaku Dosen Penguji II yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
4. Dosen dan Karyawan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember;
5. Andrias Budi Hardjo, S.Si., dan semua pihak yang telah membantu terselesaikannya skripsi ini.

Semoga bantuan, bimbingan, dan dorongan beliau dicatat sebagai amal baik oleh Allah S.W.T dan mendapat balasan yang sesuai dari-Nya. Selain itu, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, Juni 2017

Penulis

DAFTAR ISI

Halaman

HALAMAN JUDUL i
HALAMAN PERSEMBAHAN ii
HALAMAN MOTTO iii
HALAMAN PERNYATAAN iv
HALAMAN PEMBIMBING v
HALAMAN PENGESAHAN vi
RINGKASAN vii
PRAKATA ix
DAFTAR ISI x
DAFTAR GAMBAR xii
DAFTAR TABEL xiii
DAFTAR LAMPIRAN xiv
BAB 1. PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan Penelitian 3
1.4 Manfaat Penelitian 3
BAB 2. TINJAUAN PUSTAKA 4
2.1 Kriptografi 4
2.1.1 Terminologi 4
2.1.2 Prinsip Kriptografi 5
2.2 Citra 6
2.2.1 Model Citra 6
2.2.2 Digitalisasi Citra 7
2.2.3 Citra *Grayscale* 8
2.3 Bilangan Biner 8
2.4 *Stream Cipher* (*Cipher* Aliran) 10
2.5 Algoritma S-DES 12

2.5.1 Pembangkitan Kunci pada Algoritma S-DES 13
2.5.2 Enkripsi S-DES 14
2.5.3 Dekripsi S-DES 17
2.6 Analisis Keamanan 17
2.6.1 Analisis dengan Histogram Derajat Keabuan 18
2.6.2 Analisis dengan Diferensial 18
2.6.3 Analisis Sensitivitas Kunci 19
BAB 3. METODE PENELITIAN 20
3.1 Data Penelitian 20
3.2 Langkah-langkah Penelitian 20
BAB 4. HASIL DAN PEMBAHASAN 25
4.1 Hasil 25
4.1.1 Memodifikasi Kunci S-DES dengan menggunakan Metode *Stream Cipher* 26
4.1.2 Membangkitkan sub kunci K_1 dan K_2 28
4.1.3 Enkripsi *Plain Image* dengan Algoritma S-DES 29
4.1.4 Dekripsi *Cipher Image* dengan Algoritma S-DES 31
4.1.5 Analisis Hasil 34
4.1.6 Program *Secret Image Encryption* 36
4.1.7 Simulasi Program 37
4.2 Pembahasan 39
4.2.1 Proses Enkripsi 40
4.2.2 Proses Dekripsi 40
4.2.3 Analisis Keamanan 40
BAB 5. PENUTUP 42
5.1 Kesimpulan 42
5.2 Saran 42
DAFTAR PUSTAKA 43
LAMPIRAN 45

DAFTAR GAMBAR

Halaman

2.1 Proses enkripsi dan dekripsi 4
2.2 Menentukan koordinat titik pada citra 7
2.3 Alur proses pembangkitan kunci pada S-DES 14
3.1 Cameraman.tif 20
3.2 Proses enkripsi 21
3.3 Proses dekripsi 23
3.4 Skema langkah – langkah penelitian 24
4.1 Hasil proses enkripsi 25
4.2 *Key* citra 27
4.3 Analisis dengan Histogram Derajat Keabuan 34
4.4 Analisis dengan Diferensial 35
4.5 Program *Secret Image Encryption* 37
4.6 Tampilan Proses Enkripsi Program *Secret Image Encryption* 38
4.7 Tampilan Proses Dekripsi Program *Secret Image Encryption* 38
4.8 Proses Enkripsi Citra *RGB* 39
4.9 Proses Dekripsi Citra *RGB* 39

DAFTAR TABEL

Halaman

2.1 Operasi XOR pada bilangan biner ····· 10
2.2 Tabel permutasi *P*10 ····· 13
2.3 Tabel permutasi *P*8 ····· 13
2.4 Tabel permutasi *IP* ····· 15
2.5 Tabel ekspansi/permutasi ····· 15
2.6 S-Box *S*0 ····· 16
2.7 S-Box *S*1 ····· 16
2.8 Tabel permutasi *P*4 ····· 16
2.9 Tabel permutasi Akhir IP^{-1} ····· 17
4.1 Potongan derajat keabuan Cameraman.tif ····· 25
4.2 Potongan *keystream random* 10-bit ····· 26
4.3 *Keystream random* 10-bit dalam bentuk bilangan biner ····· 26
4.4 Proses Enkripsi kunci S-DES dengan metode *stream cipher* ····· 27
4.5 Hasil Enkripsi kunci S-DES dengan metode *stream cipher* ····· 27
4.6 Pembangkitan sub kunci K_1 dan K_2 ····· 28
4.7 Hasil pembangkitan sub kunci pada potongan derajat keabuan *key* citra ·· 28
4.8 Bilangan biner dari potongan *pixel* cameraman.tif ····· 29
4.9 Hasil enkripsi pada *pixel* pertama *plain image* ····· 29
4.10 Hasil enkripsi potongan *pixel* cameraman.tif ····· 31
4.11 Bilangan biner dari potongan *pixel cipher image* ····· 32
4.12 Hasil dekripsi pada *pixel* kedua *cipher image* ····· 32
4.13 Hasil dekripsi potongan *pixel cipher image* ····· 34
4.14 Analisis Sensitivitas Kunci 765 dengan kunci lainnya ····· 35

DAFTAR LAMPIRAN

Halaman

A. Matriks derajat keabuan dari Gambar 4.1 (a) 45
B. *Keystream random* 10-bit 46
C. Derajat keabuan *key* citra 47
D. Derajat keabuan hasil enkripsi 48
E. Derajat keabuan hasil dekripsi 49
F. Skrip Program Proses Enkripsi 50
G. Skrip Program Proses Dekripsi 52
H. Skrip Program Analisis dengan Diferensial 54

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Globalisasi membawa dampak yang sangat besar bagi perkembangan teknologi saat ini. Salah satu dampak yang sangat terlihat yaitu manusia sudah mulai meninggalkan penyimpanan data secara cetak dan beralih ke penyimpanan data melalui komputer dan ponsel. Penyimpanan data melalui komputer dan ponsel bukanlah tanpa risiko karena kemajuan teknologi informasi ini mempermudah pelaku kejahatan dalam bertindak. Tindakan ini sangatlah mengganggu privasi seseorang karena dengan kejahatan *cyber* ini, data tersebut bisa diketahui bahkan disalahgunakan oleh pihak – pihak yang tidak berwenang pada data tersebut. Untuk menghindari penyalahgunaan data dan pesan, perlu juga adanya perkembangan pengamanan kerahasiaan data atau pesan tersebut.

Ilmu yang mempelajari tentang kerahasiaan data atau pesan adalah kriptografi. Menurut Kromodimoeljo (2010), kriptografi adalah ilmu mengenai teknik enkripsi (pengacakan data) dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi (pengembalian data). Saat proses enkripsi dan dekripsi data ini, banyak algoritma yang bisa diterapkan, salah satunya adalah algoritma *Simplified-Data Encryption Standard* (S-DES). Algoritma S-DES adalah penyederhanaan dari algoritma *Data Encryption Standard* (DES). Algoritma S-DES ini memiliki beberapa keunggulan jika dibandingkan dengan algoritma DES, salah satunya yaitu proses pengenkripsian dan pendekripsian datanya lebih sederhana dan lebih cepat karena kuncinya hanya berupa bilangan desimal berukuran 10-bit serta tidak membutuhkan memori yang besar. Karena ukuran kunci dan prosesnya yang sederhana, S-DES memiliki kelemahan jika diterapkan pada data yang kompleks seperti citra, yaitu *pixel-pixel* yang memiliki derajat keabuan yang sama menghasilkan enkripsi yang juga bernilai sama sehingga algoritma S-DES rentan terhadap *statistical attack*. Hal ini berdasarkan penelitian Kumar dan Srivastava pada tahun 2014, yang telah mengenkripsi sebuah citra dengan menggunakan

algoritma S-DES. Oleh karena itu, modifikasi pada algoritma S-DES diperlukan untuk menutupi kelemahan algoritma tersebut.

Selain menggunakan algoritma S-DES, juga akan menggunakan metode *stream cipher* (*cipher* aliran). Metode *stream cipher* adalah metode pengenkripsian data menjadi *cipher* bit per bit (1 bit setiap kali transformasi) dengan kunci *keystream* (Munir, 2004). Pemilihan metode ini dengan mempertimbangkan kelebihan yang ada pada *stream cipher* yaitu tidak dibatasi oleh panjang bit data, sehingga metode ini mudah diterapkan pada algoritma-algoritma kriptografi.

Pada penelitian sebelumnya, Hardjo (2016) membahas mengenai pengenkripsian citra RGB (citra berwarna) dengan menggunakan penggabungan dari algoritma S-DES, algoritma DNA-*Vigenere Cipher*, dan beberapa perlakuan diantara pengenkripsian 2 algoritma tersebut. Hasil yang didapat menyatakan penggabungan dua algoritma tersebut aman diterapkan pada citra RGB.

Pada penelitian ini, penulis akan mengajukan metode baru dengan proses yang lebih sederhana dibandingkan penelitian sebelumnya tetapi memiliki hasil yang kurang lebih sama. Metode yang dimaksud yaitu mengenkripsi data digital yang berupa citra dengan modifikasi algoritma S-DES dengan *stream cipher*. Modifikasi dilakukan untuk mengatasi kelemahan dari algoritma S-DES, modifikasi yang dimaksud yaitu dengan membangkitkan kunci baru dari kunci lama S-DES menggunakan metode *stream cipher*. Penerapan metode *stream cipher* pada modifikasi algoritma S-DES ini diharapkan mampu meningkatkan keamanan data citra yang disembunyikan karena memanfaatkan dua metode pengamanan data.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas, rumusan masalah yang dikemukakan adalah:

a. Bagaimana membangkitkan kunci baru dengan mengenkripsi kunci lama dengan menggunakan metode *stream cipher*, kemudian mengenkripsi citra *grayscale* dengan kunci baru hasil enkripsi pada algoritma S-DES.

b. Bagaimana mendekripsi citra *grayscale* yang telah dienkripsi dengan modifikasi algoritma S-DES.
c. Bagaimana analisis keamanan dari modifikasi algoritma S-DES tersebut.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

a. Membangkitkan kunci baru dengan mengenkripsi kunci lama dengan menggunakan metode *stream cipher*, kemudian mengenkripsi citra *grayscale* dengan kunci baru hasil enkripsi pada algoritma S-DES.
b. Mendekripsi citra *grayscale* yang telah dienkripsi dengan modifikasi algoritma S-DES.
c. Menganalisis keamanan dari modifikasi algoritma S-DES tersebut.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

a. Mengetahui cara mengenkripsi S-DES yang sudah dimodifikasi dengan menggunakan metode *stream cipher* serta mengetahui hasilnya.
b. Mengetahui cara mengembalikan citra *grayscale* hasil enkripsi ke bentuk citra yang sebenarnya.
c. Mampu menganalisis keamanan dari modifikasi algoritma S-DES tersebut.

BAB 2. TINJAUAN PUSTAKA

2.1 Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani yaitu "*crypto*" yang berarti "*secret*" (rahasia) dan "*graphy*" yang berarti "*writing*" (tulisan). Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga (Firmansyah, 2012).

Secara umum, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan (data atau informasi) dengan menyamarkan menjadi data yang tidak lagi bermakna. Pesan yang akan dirahasiakan dinamakan *plaintext* atau dinamakan *plain image* jika pesan dalam bentuk citra, sedangkan pesan hasil penyamaran dinamakan *ciphertext* atau dinamakan *cipher image* jika pesan dalam bentuk citra. Proses penyamaran dari *plaintext* ke *ciphertext* disebut dengan enkripsi (*encryption*) dan proses kebalikan dari *ciphertext* ke *plaintext* disebut dengan dekripsi (*decryption*). Proses enkripsi dan dekripsi ini membutuhkan kunci dan algoritma matematika. Algoritma tersebut merupakan suatu fungsi matematika yang digunakan pada proses enkripsi dan dekripsi. Gambar 2.1 menunjukkan proses enkripsi dan dekripsi.

Gambar 2.1 Proses enkripsi dan dekripsi

2.1.1 Terminologi

Dalam kriptografi ada beberapa terminologi atau istilah yang harus diketahui sebagai berikut (Munir, 2004):

a. Pengirim dan Penerima pesan

   Seorang pengirim pesan (*sender*) ingin mengirim pesan kepada seorang penerima pesan (*receiver*). Pengirim pesan menginginkan pesan dapat dikirim secara aman yaitu pihak selain penerima pesan tidak dapat membaca isi pesan.

b. Pesan, *Plaintext*, dan *Ciphertext*

   Pesan adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah *plaintext*, pesan yang berbentuk citra dinamakan *plain image*. Pesan ini nantinya akan disandikan, tujuannya agar pesan tidak bisa dibaca oleh pihak lain. Hasil penyandian dari pesan tersebut dinamakan *ciphertext* atau *cipher image* jika data dalam bentuk citra. *Ciphertext* ini nantinya harus bisa ditransformasi kembali menjadi *plaintext*.

c. Enkripsi dan Dekripsi

   Proses menyandikan *plaintext* menjadi *ciphertext* disebut enkripsi (*encryption*), Sedangkan proses mengembalikan *ciphertext* menjadi *plaintext* disebut dekripsi (*decryption*).

d. Algoritma kriptografi dan Kunci

   Algoritma kriptografi adalah aturan untuk *encryption* dan *decryption*. Aturan ini dalam bentuk fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk enkripsi dan dekripsi. Sedangkan kunci adalah parameter yang berupa suatu deretan *bit* yang diperlukan untuk mengontrol jalannya algoritma. Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya.

2.1.2 Prinsip Kriptografi

Menurut Firmansyah (2012), prinsip yang mendasari adanya kriptografi adalah sebagai berikut:

a. *Confidentiality* (kerahasiaan) yaitu layanan agar pesan yang dikirimkan kerahasiaannya terjaga dari pihak lain kecuali pengirim, penerima dan pihak-pihak yang diperbolehkan membaca pesan. Hal ini dilakukan dengan

menggunakan sebuah algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.

b. Data *integrity* (keutuhan data) yaitu layanan yang mampu mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan) data yang tidak sah oleh pihak lain.
c. *Authentication* (otentik) yaitu layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian sebuah data.
d. *Non-repudiation* (anti-penyangkalan) yaitu layanan untuk mencegah suatu pihak menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dari dirinya).

2.2 Citra

Citra (*image*) adalah gambar pada bidang dua dimensi (2D). Ditinjau dari sudut pandang matematis, citra merupakan fungsi berkelanjutan (*continue*) dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Kemudian pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang tidak lain adalah citra tersebut akan terekam (Murni, 1992).

Citra dibagi menjadi 2 jenis yaitu citra diam (*still image*) dan citra bergerak (*moving picture*). Citra diam (*still image*) adalah citra tunggal yang tidak bergerak sedangkan citra bergerak (*moving picture*) adalah gabungan dari beberapa citra diam (*still image*) yang ditampilkan secara berurutan sehingga memberi kesan pada optik mata sebagai citra yang bergerak. Ilusi optik sangat berperan penting pada pembentukan citra bergerak (*moving picture*). Setiap citra diam (*still image*) dalam rangkaian citra bergerak (*moving picture*) itu biasa disebut dengan *frame.*

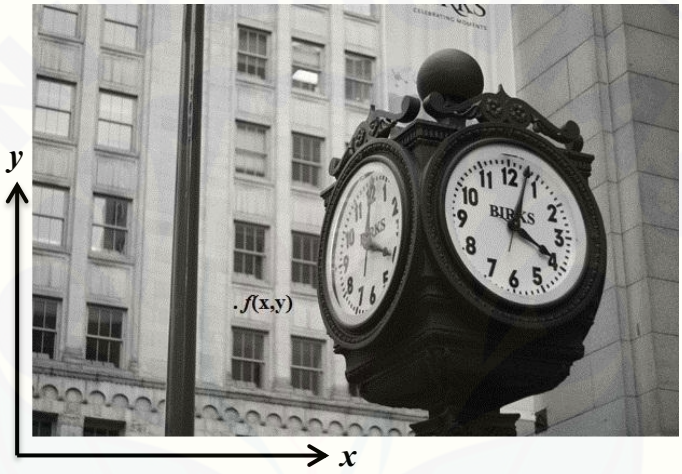
2.2.1 Model Citra

Secara matematis fungsi intensitas cahaya pada bidang dua dimensi disimbolkan dengan $f(x, y)$, yang dalam hal ini:

(x,y) : koordinat pada bidang dua dimensi

f(x,y) : intensitas cahaya (*brightness*) pada titik (x,y)

Koordinat pada sebuah citra diilustrasikan pada Gambar 2.2. Intensitas f dari gambar hitam putih pada titik (x,y) disebut sebagai derajat keabuan (*gray level*), sedangkan citranya disebut sebagai citra hitam-putih (*grayscale image*) atau citra monokrom (*monochrome image*). Derajat keabuan memiliki rentang nilai dari l_{min} sampai l_{max} atau secara matematis ditulis $l_{min} < f < l_{max}$. Selang (l_{min}, l_{max}) disebut sebagai skala keabuan (Gonzalez, 1977).



Gambar 2.2 Menentukan koordinat titik pada citra

(Sumber: Kaplan)

2.2.2 Digitalisasi Citra

Suatu citra harus direpresentasikan secara numerik dengan nilai-nilai diskrit agar suatu citra dapat diolah dengan komputer digital. Representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut sebagai proses digitalisasi. Citra hasil representasi inilah yang disebut sebagai citra digital (*digital image*). Pada umumnya, citra digital berbentuk persegi panjang, dan dimensi ukurannya dinyatakan dengan tinggi x lebar atau lebar x panjang.

Citra digital berukuran N x M lazim dinyatakan dengan matriks yang berukuran N baris dan M kolom sebagai berikut :

$$f(x,y) = \begin{bmatrix} f(1,1) & \cdots & f(1,M) \\ \vdots & \ddots & \vdots \\ f(N,1) & \cdots & f(N,M) \end{bmatrix}$$

Indeks baris (*i*) dan indeks kolom (*j*) menyatakan suatu koordinat titik pada citra, dan *f*(*i*,*j*) merupakan intensitas (derajat keabuan) pada titik (*i*,*j*).

Masing-masing elemen pada citra digital (elemen matriks) disebut dengan *pixel*. Jadi citra yang berukuran *N* x *M* mempunyai *NM* buah pixel (Dulimarta, 1997).

2.2.3 Citra *Grayscale*

Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, artinya nilai dari *Red* = *Green* = *Blue*. Citra *grayscale* berbeda dengan citra ”hitam-putih”, dimana pada konteks komputer citra ”hitam-putih” hanya terdiri atas 2 warna yaitu hitam dan putih saja. Pada citra *grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra *grayscale* mempunyai nilai minimum dan nilai maksimum. Banyaknya kemungkinan nilai minimum dan maksimum bergantung pada jumlah bit yang digunakan. Contohnya untuk skala keabuan 4 bit, maka jumlah kemungkinan nilainya adalah $2^4 = 16$, dan nilai maksimumnya adalah $2^4-1 = 15$, sedangkan untuk skala keabuan 8 bit, maka jumlah kemungkinan nilainya adalah $2^8 = 256$, dan nilai maksimumnya adalah $2^8 – 1 = 255$.

Secara digital suatu *grayscale image* dapat direpresentasikan dalam bentuk array dua dimensi. Tiap elemen dalam array menunjukkan intensitas (*graylevel*) dari *image* pada posisi koordinat yang bersesuaian. Apabila suatu citra direpresentasikan dalam 8 bit maka berarti pada citra terdapat 2^8 atau 256 level *grayscale* (biasanya bernilai 0 – 255), dimana 0 menunjukkan level intensitas paling gelap dan 255 menunjukkan intensitas paling terang. Format citra ini disebut skala keabuan karena pada umumnya warna yang dipakai adalah antara hitam sebagai warna minimal dan warna putih sebagai warna maksimal sehingga warna antaranya adalah abu-abu.

2.3 Bilangan Biner

Sistem bilangan biner adalah susunan bilangan yang mempunyai basis 2 sebab sistem bilangan ini menggunakan dua nilai koefisien yang mungkin yaitu 0

dan 1. Berbeda dengan bilangan desimal yang memiliki susunan basis 10 yaitu 0,1,2,3,4,5,6,7,8 dan 9. Sederhananya diberikan contoh di bawah ini.

Untuk Desimal:

$14_{(10)} = (1*10^1) + (4*10^0) = 10 + 4 = 14$

Untuk Biner:

$1110_{(2)} = (1*2^3) + (1*2^2) + (1*2^1) + (0*2^0) = 8 + 4 + 2 + 0 = 14$

a. Konversi bilangan desimal ke bilangan biner

Untuk mengubah angka desimal menjadi angka biner digunakan metode pembagian dengan angka 2 sambil memperhatikan sisanya. Sisa yang pertama akan menjadi *least significant bit* (*LSB*) dan sisa yang terakhir menjadi *most significant bit* (*MSB*).

Contoh:

Mengubah bilangan $30_{(10)}$ (bilangan desimal) menjadi bilangan biner

30 : 2 = 30 sisa 0 (*LSB*)

15 : 2 = 7 sisa 1

7 : 2 = 3 sisa 1

3 : 2 = 1 sisa 1

2 : 2 = 1 (*MSB*)

Maka bilangan biner dari 30 adalah 11110

b. Konversi bilangan biner ke bilangan desimal

Masing-masing digit dalam sistem biner disebut bit (*binary digit*) dan hanya mempunyai dua harga, 0 dan 1. Konversi dilakukan dengan menggunakan persamaan (2.1).

$$D_r = \sum_i^{n-1}(d_i \cdot r^i) \qquad (2.1)$$

dimana:

r = basis bilangan biner yaitu 2

i = posisi nilai biner, dimulai dari 0

d = nilai biner

n = banyaknya angka biner

Contoh:

$11100_{(2)} = 0*2^0 + 0*2^1 + 1*2^2 + 1*2^3 + 1*2^4 = 0 + 0 + 4 + 8 + 16 = 28$

Bilangan biner memiliki beberapa operasi bilangan, salah satunya adalah *exclusive-OR* (XOR). Tabel 2.1 menunjukkan hasil dari operasi XOR pada bilangan biner.

Tabel 2.1 Operasi XOR pada bilangan biner

Bilangan 1	Bilangan 2	Bilangan 1 ⊕ Bilangan 2
1	1	0
1	0	1
0	1	1
0	0	0

2.4 *Stream Cipher* (*Cipher* Aliran)

Algoritma kriptografi simetris adalah salah satu jenis kriptografi yang pada proses enkripsi dan dekripsinya menggunakan kunci yang sama. Ada dua jenis algoritma kriptografi simetris yaitu *block cipher* dan *stream cipher*. Pada algoritma kriptografi *block cipher*, metode enkripsi dan dekripsi dilakukan dengan membagi *plaintext* menjadi blok-blok, dan masing-masing blok dilakukan enkripsi menggunakan kunci yang sama. Contoh *block cipher* yang dikenal luas saat ini adalah DES dan AES. Sedangkan pada algoritma kriptografi *stream cipher*, *plaintext* tidak dipotong menjadi blok-blok, akan tetapi enkripsi dilakukan secara mengalir menggunakan enkripsi dengan kunci yang mengalir juga sehingga bisa dikatakan mengenkripsinya bit per bit. Algoritma kriptografi *stream cipher* sering juga disebut dengan sandi aliran. *Stream cipher* banyak digunakan secara luas di internet dan di telepon seluler. Hal ini terjadi karena algoritma pada *stream cipher* memiliki proses enkripsi dan dekripsi yang relatif lebih cepat dibandingkan *block cipher*. Salah satu keuntungan dari *stream cipher* adalah tidak dibatasi oleh panjang *plaintext*, sehingga *stream cipher* cocok untuk digunakan pada enkripsi suatu komunikasi yang berlangsung secara berkelanjutan, seperti komunikasi

melalui telepon (Lestari dan Riyanto, 2012). Proses enkripsi dari *stream cipher* dapat dituliskan sebagai berikut.

$$C_i = P_i \oplus K \tag{2.2}$$

Sehingga proses dekripsi dari *stream cipher* adalah

$$P_i = C_i \oplus K$$

dimana P_i merupakan *plaintext* ke-*i*, C_i merupakan *ciphertext* ke-*i*, dan K merupakan *keystream*.

Kunci pada *stream cipher* disebut dengan *keystream* dan kunci tersebut dibangkitkan oleh prosedur yang disebut dengan *Keystream Generator*. Pembangkit harus menghasilkan bit-bit kunci yang kuat secara kriptografi karena keamanan sistem *cipher* aliran bergantung seluruhnya pada *keystream generator*.

Terdapat tiga kasus yang dihasilkan oleh *keystream generator*:

a. *Keystream* seluruhnya 0

   Jika pembangkit mengeluarkan aliran-bit-kunci yang seluruhnya nol maka *ciphertext* sama dengan *plaintext*, karena

   $$C_i = P_i \oplus 0 = P_i$$

   dan proses enkripsi menjadi tak-berarti.

b. *Keystream* berulang secara perodik

   Jika pembangkit mengeluarkan *keystream* yang berulang secara periodik, maka algoritma enkripsinya sama dengan algoritma dekripsi dengan XOR sederhana yang memiliki tingkat keamanan yang rendah.

c. *Keystream* benar-benar acak

   Jika pembangkit mengeluarkan *keystream* benar-benar acak (*truly random*), maka algoritma enkripsinya sama dengan *one-time pad* dengan tingkat keamanan yang sempurna. *One-time pad* ini juga disebut *cipher Vernam* karena diciptakan oleh Vernam pada tahun 1917. *One-time pad* adalah satu-satunya algoritma enkripsi yang tetap aman meskipun daya komputasi dan pengetahuan musuh handal (Wu, 2008).

2.5 Algoritma S-DES

Simplified Data Encryption Standard (S-DES) adalah algoritma yang diadaptasi dari algoritma DES. Algoritma ini jauh lebih sederhana jika dibandingkan dengan DES meskipun sifat dan strukturnya sama dengan DES. Algoritma S-DES memiliki beberapa keuntungan jika dibandingkan dengan algoritma DES, yaitu S-DES eksekusinya jauh lebih cepat dan ukuran kunci dari S-DES lebih kecil daripada DES sehingga lebih mudah untuk digunakan. Untuk proses enkripsi, S-DES membutuhkan 8-bit *plaintext* dan 10-bit kunci. Hasil dari enkripsi ini akan menghasilkan 8-bit *ciphertext*. Sedangkan untuk proses dekripsi, S-DES membutuhkan 8-bit *ciphertext* dan 10-bit kunci. Hasil dari dekripsi ini akan menghasilkan 8-bit *plaintext* (Sharma dan Gupta, 2013).

Ada 5 fungsi yang akan digunakan saat proses enkripsi maupun dekripsi sebagai berikut:

a. Permutasi awal (*IP*)
b. Fungsi kompleks f_K, yang melibatkan operasi permutasi dan substitusi. Fungsi f_K bergantung pada kunci
c. Permutasi sederhana (*SW*) yang menukar posisi dari setengah bagian data
d. Fungsi kompleks f_K
e. Fungsi permutasi yang merupakan invers dari permutasi awal (IP^{-1}).

Secara singkat, proses enkripsi dapat ditulis sebagai berikut:

$$Ciphertext = IP^{-1}\,(f_{K_2}\,(SW\,(f_{K_1}\,(IP\,(plaintext)))))\qquad(2.3)$$

dimana

$$K_1 = P8\,(Shift\,(P10\,(key)))$$
$$K_2 = P8\,(Shift\,(Shift\,(Shift\,(P10\,(key)))))$$

sehingga proses dekripsi dapat ditulis sebagai berikut:

$$Plaintext = IP^{-1}\,(f_{K_1}\,(SW\,(f_{K_2}\,(IP\,(ciphertext)))))\qquad(2.4)$$

pada persamaan (2.3) yang bertindak sebagai *domain* adalah *plaintext* yang direlasikan ke *kodomain* yaitu *ciphertext* dengan menggunakan 5 fungsi yang sudah disebutkan di atas. Perlakuan fungsinya secara berurutan dari *IP*, f_{K1}, *SW*, f_{K2}, dan terakhir IP^{-1}, sedangkan pada persamaan (2.4) berkebalikan dengan persamaan (2.3) (Hardjo, 2016).

2.5.1 Pembangkitan Kunci pada algoritma S-DES

S-DES membutuhkan 10-bit kunci masukan. Dari kunci masukan tersebut akan diperoleh 8-bit sub kunci yang akan digunakan pada proses enkripsi dan dekripsi. Alur proses pembangkitan sub kunci dari kunci utama dapat dilihat pada Gambar 2.3.

Langkah-langkah pembangkitan kunci pada Gambar 2.3 dijelaskan sebagai berikut:

a. Kunci *plaintext* yang terdiri dari 10-bit yaitu (1, 2, 3, 4, 5, 6, 7, 8, 9, 10) dilakukan permutasi *P*10 sehingga menjadi (3, 5, 2, 7, 4, 10, 1, 9, 8, 6), *P*10 bertujuan untuk membuat kunci *plaintext* menjadi acak. Tabel permutasi *P*10 dapat dilihat pada Tabel 2.2.

Tabel 2.2 Tabel permutasi *P*10

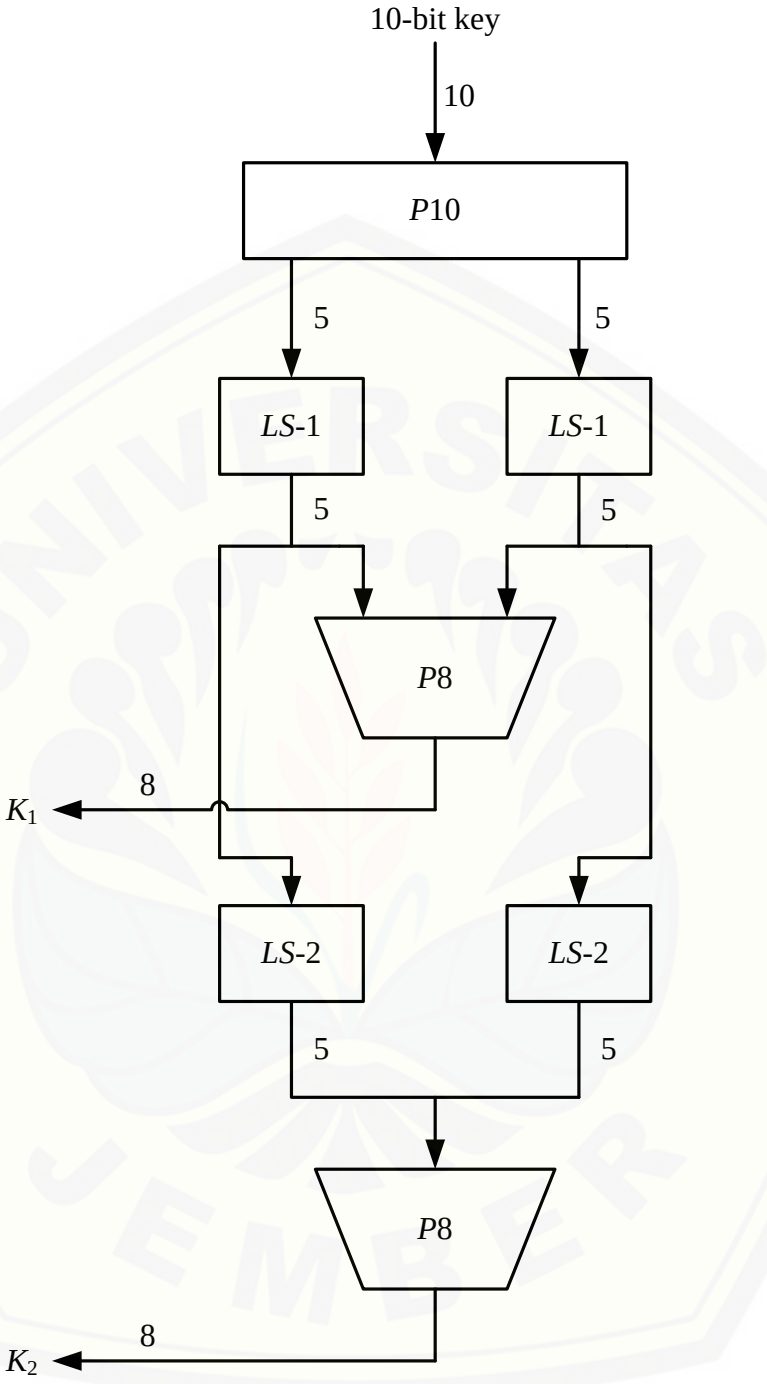
*P*10									
3	5	2	7	4	10	1	9	8	6

b. Lakukan pergeseran kiri (*LS*-1) masing-masing untuk 5-bit awal dan 5-bit akhir. *LS*-1 yaitu menggeser bit-bit ke kiri sekali.
c. Lakukan operasi permutasi *P*8 pada hasil (*LS*-1). Operasi *P*8 bertujuan untuk mengambil 8-bit dari 10-bit kunci. Tabel permutasi *P*8 dapat dilihat pada Tabel 2.3.

Tabel 2.3 Tabel permutasi *P*8

*P*8							
6	3	7	4	8	5	10	9

d. Hasil dari langkah (c) adalah sub kunci K_1
e. Hasil dari langkah (b) dilakukan pergeseran kiri sebanyak 2 kali (*LS*-2) masing-masing untuk 5-bit awal dan 5-bit akhir.
f. Operasikan hasil dari (e) dengan *P*8 untuk memperoleh sub kunci K_2.



Gambar 2.3 Alur proses pembangkitan kunci pada S-DES

2.5.2 Enkripsi S-DES

Sesuai dengan persamaan (2.3), maka proses enkripsi pada algoritma S-DES terdiri dari 5 langkah. Langkah-langkah enkripsi S-DES sebagai berikut:

a. Masukan berupa 8-bit *plaintext* yang kemudian dilakukan operasi *IP* agar data menjadi acak. Tabel permutasi *IP* dapat dilihat pada Tabel 2.4.

Tabel 2.4 Tabel permutasi *IP*

IP							
2	6	3	1	4	8	5	7

b. Fungsi f_K

Fungsi f_K adalah fungsi yang paling kompleks dari algoritma S-DES karena pada fungsi ini menggunakan fungsi permutasi dan substitusi. Fungsi f dapat dituliskan sebagai berikut.

$$f_K(L, R) = (L \oplus F(R, SK), R) \quad (2.5)$$

dimana L adalah 4-bit pertama dari *plaintext* hasil operasi *IP*, R adalah 4-bit terakhir dari *plaintext* hasil operasi *IP*, *SK* merupakan sub kunci, dan $\oplus$ adalah operasi XOR.

Proses perhitungan $F(R, SK)$ secara detail adalah sebagai berikut.

1) R adalah masukan berukuran 4-bit untuk fungsi f_K
2) Operasi pertama yang dilakukan adalah operasi ekspansi/permutasi (*E/P*). Operasi ini menyebabkan R yang awalnya berukuran 4-bit berubah menjadi berukuran 8-bit. Tabel ekspansi/permutasi (*E/P*) dapat dilihat pada Tabel 2.5.

Tabel 2.5 Tabel ekspansi/permutasi

E/P							
4	1	2	3	2	3	4	1

3) Hasil dari langkah ke (2) kemudian dilakukan operasi XOR dengan sub kunci K_1, keluarannya nantinya akan berukuran 8-bit.
4) Hasil dari langkah ke (3) nantinya akan dibagi menjadi 2 bagian yaitu bagian pertama yang berisi 4-bit pertama dan bagian kedua yang berisi 4-bit terakhir. Bagian pertama akan diproses dengan S-Box *S*0 dan bagian kedua

akan diproses dengan S-Box *S*1. Tabel S-box *S*0 dan tabel S-box *S*1 dapat dilihat pada Tabel 2.6 dan Tabel 2.7.

Tabel 2.6 S-Box *S*0

	0	1	2	3
0	1	0	3	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

Tabel 2.7 S-Box *S*1

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

5) Operasi pada S-Box berlaku demikian. Bit pertama dan keempat dari masukan diperlakukan seperti bilangan 2-bit yang dapat dilihat pada baris S-Box, sementara bit kedua dan ketiga dapat dilihat pada kolom S-Box. Nilai pada baris dan kolom yang bersangkutan merupakan keluaran 2-bit dalam basis 2.
6) 4-bit hasil penggabungan dari *S*0 dan *S*1 dilakukan permutasi *P*4 seperti pada Tabel 2.8.

Tabel 2.8 Tabel permutasi *P*4

*P*4			
2	4	3	1

7) Hasil dari langkah (6) merupakan keluaran dari fungsi *F*(*R*, *SK*).

c. Fungsi *SW*

Fungsi *SW* berfungsi menukar 4-bit awal dari hasil fungsi f_K pada proses sebelumnya dengan 4-bit akhir. Hal ini diperlukan sebab fungsi f_K hanya mengubah 4-bit awal, sehingga fungsi f_K berikutnya beroperasi pada 4-bit yang lain.

d. Fungsi f_K

Pada perulangan fungsi f_K yang kedua ini perhitungannya sama dengan fungsi f_K yang pertama, perbedaannya hanya terletak pada penggunaan sub kuncinya saja. Untuk fungsi f_K yang kedua ini sub kunci K_1 diganti dengan sub kunci K_2

e. Permutasi akhir

Hasil dari fungsi f_K yang kedua kemudian diproses dengan IP^{-1} sehingga diperoleh 8-bit *ciphertext*. Tabel permutasi akhir (IP^{-1}) dapat dilihat pada Tabel 2.9.

Tabel 2.9 Tabel permutasi Akhir IP^{-1}

IP^{-1}							
4	1	3	5	7	2	8	6

2.5.3 Dekripsi S-DES

Proses dekripsi pada algoritma S-DES tidak jauh berbeda dengan proses enkripsinya. Fungsi yang digunakan juga sama yaitu sebanyak 5 fungsi, perbedaannya terletak pada fungsi f_K, untuk proses dekripsi algoritma S-DES pada proses perhitungan fungsi f_K yang pertama menggunakan sub kunci K_2 dan proses perhitungan fungsi f_K yang kedua menggunakan sub kunci K_1.

2.6 Analisis Keamanan

Ada beberapa metode yang bisa digunakan untuk menganalisis keamanan suatu algoritma kaitannya dengan perlindungan data pada sebuah gambar. Berikut ini adalah beberapa analisis keamanan yang bisa digunakan.

2.6.1 Analisis dengan Histogram Derajat Keabuan

Histogram adalah gambaran informasi mengenai penyebaran nilai warna pada sebuah citra. Teknik analisis dengan histogram digunakan untuk melihat kesesuaian distribusi warna antara *plain image* dengan *cipher image*. Jika histogram *cipher image* memiliki distribusi yang mendekati seragam dan memiliki perbedaan yang signifikan dengan *plain image*, maka dapat dikatakan *cipher image* tidak memberikan petunjuk untuk melakukan *statistical attack* pada *cipher image* yang dihasilkan. *Statistical attack* ini menyebabkan histogram akan memberikan informasi yang dapat digunakan untuk mendeduksi *plain image* (Behnia dkk., 2007).

Teknik *statistical attack* adalah teknik yang digunakan untuk mendeduksi *plain image* dengan mencari kesamaan nilai warna pada citra. Berdasarkan uraian di atas, analisis dengan histogram derajat keabuan ini memperhatikan dua hal yaitu penyebaran *pixel plain image* dan *cipher image* serta signifikan tidaknya perbedaan histogram *plain image* dan histogram *cipher image*.

2.6.2 Analisis dengan Diferensial

Analisis ini digunakan untuk menentukan perbedaan dari dua buah citra, dengan cara menghitung nilai dari *number of pixels change rate* (*NPCR*). Tujuan dari pengujian ini yaitu untuk menjamin bahwa pada setiap titik matriks citra hasil enkripsi terdapat perubahan elemen warna dengan citra sebelum dienkripsi. *Number of pixels change rate* (*NPCR*) dirumuskan sebagai berikut:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \tag{2.6}$$

dimana W dan H merupakan lebar dan tinggi dari citra, sedangkan $D(i,j)$ adalah fungsi yang ditentukan dengan aturan sebagai berikut

$$D(i,j) = \begin{cases} 0, & C(i,j) = C'(i,j) \\ 1, & C(i,j) \neq C'(i,j) \end{cases}$$

dimana $C(i,j)$ dan $C'(i,j)$ masing–masing merupakan nilai derajat keabuan dari baris i dan kolom j dari citra C dan C'. C ini nantinya ditetapkan sebagai *plain image* dan C' ditetapkan sebagai *cipher image*.

Nilai *NPCR* berkisar antara 0 – 100%. Ketika *NPCR* bernilai 0%, maka artinya citra asli sama persis dengan citra yang terenkripsi. Namun, ketika *NPCR* bernilai 100%, maka keseluruhan *pixel* dari citra awal berbeda dengan citra terenkripsi. *NPCR* dengan nilai diatas 90% akan menyulitkan kriptaanalisis dalam mencari hubungan statistik antara citra asli dengan citra terenkripsi (Irfan, 2016).

2.6.3 Analisis Sensitivitas Kunci

Analisis sensitivitas kunci ini digunakan untuk mengetahui seberapa sensitif suatu kunci dari suatu algoritma. Sensitif tidaknya sebuah kunci bisa dilihat pada hasil enkripsinya. Dua buah kunci yang sedikit berbeda akan menghasilkan *cipher image* yang sangat berbeda jika sensitivitas kunci-nya besar. Sensitivitas kunci yang besar ini juga akan berpengaruh pada hasil dekripsi *cipher image*, karena kunci yang salah akan menghasilkan citra yang sangat berbeda dengan *plain image*. Analisis sensitivitas kunci dihitung menggunakan *Number of pixels change rate* atau *NPCR* (Irfan, 2016).

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang penulis gunakan dalam penelitian ini adalah citra *grayscale* yang berlaku sebagai *plain image*. Gambar 3.1 merupakan citra *grayscale* yang digunakan sebagai *plain image.* Citra berdimensi 256x256 *pixels*.

Gambar 3.1 Cameraman.tif
(Sumber : MATLAB *library*)

3.2 Langkah-langkah Penelitian

Secara sistematis, langkah-langkah penelitian yang dilakukan adalah sebagai berikut.

a. Studi Literatur

Studi literatur dilakukan dengan melakukan pemahaman mengenai teori – teori yang berkaitan dengan penelitian meliputi: algoritma S-DES, citra *grayscale* dan teori mengenai metode *stream cipher*. Literatur pendukung ini berupa jurnal, artikel, buku, dan sumber lainnya.

b. Tahap Perancangan

Tahap perancangan program menggunakan *software* MATLAB R2009a dan melakukan perancangan desain GUI untuk membuat tampilan layaknya sebuah aplikasi, seperti tata letak tombol-tombol untuk setiap proses yang dibutuhkan,

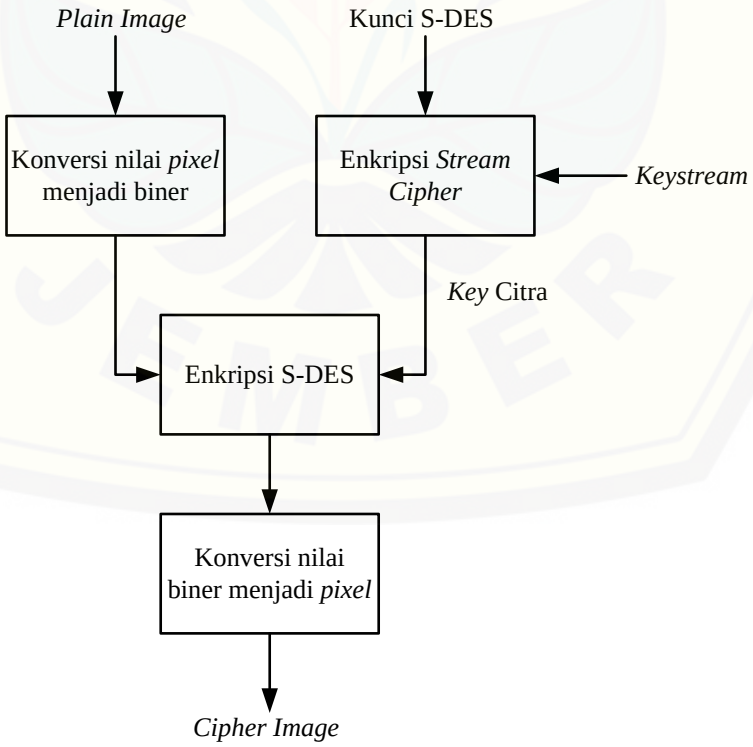
tata letak *properties* pendukung lainnya, serta pengaturan warna agar tampilan lebih bagus.

c. Pembuatan Program

Pembuatan program dilakukan berdasarkan konsep matriks sebagai pembangkit citra pada MATLAB R2009a, melakukan proses enkripsi pada kunci S-DES dengan menggunakan *stream cipher* kemudian dilanjutkan dengan mengenkripsi *plain image* dengan algoritma S-DES serta melakukan proses dekripsi pada citra yang sudah dienkripsi sebelumnya. Berikut uraian proses enkripsi dan proses dekripsi.

1) Proses enkripsi

Langkah-langkah pada proses enkripsi ini diterapkan pada seluruh *pixel* pada citra yang menjadi *plain image*. Proses enkripsi secara garis besar pada penelitian ini ada 2 bagian yaitu bagian pertama mengenai modifikasi kunci algoritma S-DES dengan *stream cipher* dan bagian kedua mengenai pengenkripsian *plain image* dengan kunci yang sudah dimodifikasi. Untuk lebih jelasnya, diberikan Gambar 3.2 berikut.



Gambar 3.2 Proses enkripsi

Proses enkripsi pada Gambar 3.2 di atas diuraikan sebagai berikut.

a) Konversi nilai *pixel* menjadi biner

   Setiap derajat keabuan pada *pixel plain image* dikonversi menjadi bilangan biner 8 digit.

b) Enkripsi *stream cipher*

   Kunci S-DES yang merupakan bilangan biner 10 digit dienkripsi terlebih dahulu dengan *stream cipher* menggunakan persamaan (2.2), *keystream* dibangkitkan dari bilangan biner *random* 10 digit. Proses pengenkripsian kunci S-DES dengan *stream cipher* ini nantinya akan menghasilkan *key* citra berukuran sama dengan citra pada *plain image*. Langkah ini adalah bentuk modifikasi yang penulis ajukan, kunci S-DES yang awalnya hanya menggunakan satu kunci saja diganti dengan *key* citra yang banyaknya kunci menyesuaikan ukuran dari citra *plain image*.

c) Enkripsi dengan algoritma S-DES

   Nilai derajat keabuan pada setiap *pixel* pada *plain image* akan dienkripsi dengan algoritma S-DES (seperti penjelasan pada subsubbab 2.5.2). Pengenkripsiannya menggunakan *key* citra hasil proses (b).

d) Konversi nilai biner menjadi *pixel*

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

2) Proses dekripsi

a) Konversi nilai *pixel* menjadi biner

   Setiap derajat keabuan pada *pixel cipher image* dikonversi menjadi bilangan biner 8 digit.

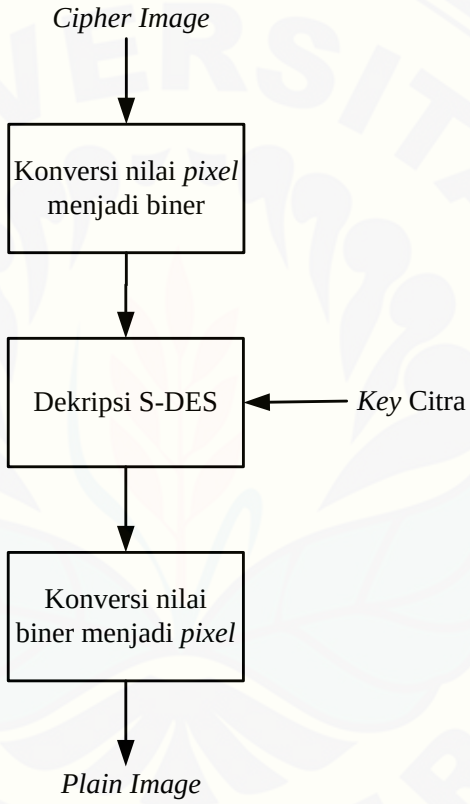
b) Dekripsi dengan algoritma S-DES

   Cipher image akan didekripsi dengan algoritma S-DES menggunakan *key* citra dari proses enkripsi sebelumnya, proses dekripsi dengan

algoritma S-DES ini ditunjukkan pada persamaan (2.4). Hasil dari dekripsi ini adalah *plain image* yang berupa bilangan biner 8 bit.

c) Konversi nilai biner menjadi *pixel*

Setiap bilangan biner 8 bit hasil dari dekripsi algoritma S-DES pada proses (c) dikonversi menjadi bilangan desimal yang nantinya akan mengisi nilai derajat keabuan pada *pixel* dan merupakan hasil akhir dari proses dekripsi. Untuk lebih jelasnya, perhatikan Gambar 3.3 berikut.



Gambar 3.3 Proses dekripsi

Langkah-langkah pada proses dekripsi ini diterapkan pada seluruh *pixel* pada citra yang menjadi *cipher image*.

d. Analisis hasil

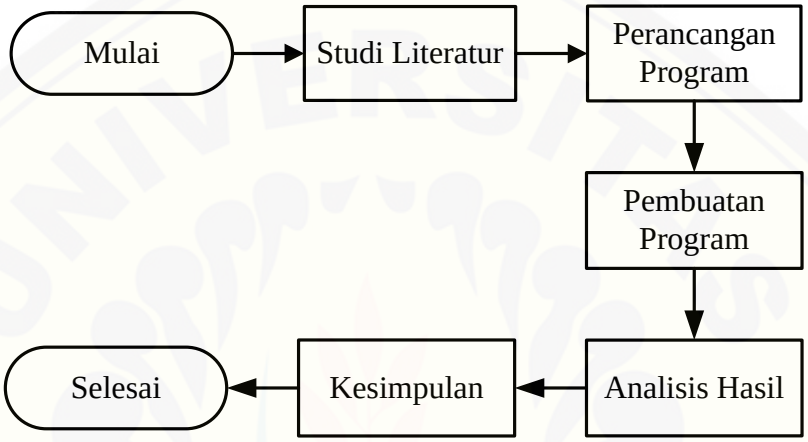
Pada tahap ini, penulis akan menguji program dengan cara menjalankan program tersebut, mengecek apakah *properties* berjalan sesuai dengan apa yang penulis kehendaki atau tidak. Pada tahap ini penulis juga menganalisis proses enkripsi dan dekripsi, selain itu penulis juga menganalisis data yang diperoleh melalui proses enkripsi dan dekripsi menggunakan analisis dengan

histogram derajat keabuan dan analisis dengan diferensial. Tahap ini dilakukan guna melihat aman tidaknya algoritma yang diajukan.

e. Kesimpulan

Membuat kesimpulan berdasarkan analisis yang dibuat.

Skema langkah – langkah penelitian yang akan dilakukan diberikan dalam Gambar 3.4



Gambar 3.4 Skema langkah – langkah penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

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

a. Proses enkripsi algoritma S-DES dengan memodifikasi kuncinya terlebih dahulu dengan metode *stream cipher* dapat dilakukan serta memberikan keamanan bagi data karena modifikasi kunci memungkinkan hasil enkripsi sebuah data lebih bervariasi sehingga menghasilkan *cipher image* yang sangat berbeda dari *plain image*.
b. Proses dekripsi tidak lagi menggunakan kunci pada proses enkripsi tapi menggunakan kunci yang sudah dimodifikasi dan tersimpan dalam bentuk citra yaitu *key* citra. Proses dekripsi mampu mengembalikan *cipher image* menjadi *plain image* tanpa menghilangkan informasi yang ada.
c. Metode yang diajukan memiliki kunci yang sensitif dan aman terhadap serangan. Hal ini dilihat dari hasil analisis keamanan yang telah dilakukan.

5.2 Saran

Saran yang dapat diberikan untuk penelitian selanjutnya yaitu menerapkan algoritma yang sudah dimodifikasi pada data yang lain seperti *text* dan juga pada *barcode*.

DAFTAR PUSTAKA

Behnia, S., A. Akhsani, S. Ahadpour, H. Mahmodi, & A. Akhavan. 2007. A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps. *Physics Letters A* 366: 391-396.

Dulimarta, H. S. 1997. *Diktat Kuliah Pengolahan Citra*. Bandung : Jurusan Teknik Informatika Institut Teknologi Bandung.

Firmansyah, E. R. 2012. *Algoritma Kriptografi & Contohnya.* Jakarta : Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta.

Gonzalez, R.C. 1977. *Digital Image Processing.* USA : Addison-Wesley Publishing.

Hardjo, A. B. 2016. *Enkripsi Citra RGB Menggunakan Algoritma Simplified Data Encryption Standard (S-DES) dan DNA-Vigenere Cipher*. Tidak dipublikasikan. Skripsi. Jember : Jurusan Matematika Fakultas MIPA Universitas Jember.

Irfan, P. 2016. Aplikasi enkripsi citra menggunakan Algoritma Kriptografi Arnold Cat Map dan Logistic Map. *JURNAL MATRIK* 16(1): 96-104.

Kaplan, D. 2015. *Wallpapers for Grayscale | Resolution* 600x400. http://www.lanlinglaurel.com/grayscale-image/5704969.html. [Diakses pada 2 Maret 2017].

Kromodimoeljo, S. 2010. *Teori dan Aplikasi Kriptografi*. Jakarta : SPK IT Consulting.

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

Lestari, D., & M. Z. Riyanto. 2012. Suatu Algoritma Kriptografi Stream Cipher Berdasarkan Fungsi Chaos. *Prosiding, Seminar Nasional Matematika dan Pendidikan Matematika FMIPA UNY*.

Munir, R. 2004. *Diktat Kuliah IF5054 Kriptografi*. Jakarta : Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika.

Murni, A. 1992. *Pengantar Pengolahan Citra*. Jakarta : PT Elex Media Komputindo.

Sharma, I. R., & V. Gupta. 2013. Comparative Analysis of DES and S-DES Encryption Algorithm Using Verilog Coding. *International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering* 1(9): 469-473.

Wu, H. 2008. *Cryptanalysis and Design of Stream Ciphers.* Leuven : Katholieke Universiteit Leuven.

LAMPIRAN

LAMPIRAN A. Matriks derajat keabuan dari Gambar 4.1 (a)

Ukuran matriks : 256x256

	1	2	3	4	5	6	7	8	9	10	11	12	...	256
1	**156**	**159**	**158**	155	158	156	159	158	157	158	158	159	…	152
2	**160**	**154**	**157**	158	157	159	158	158	158	160	155	156	...	153
3	**156**	**159**	**158**	155	158	156	159	158	157	158	158	159	...	152
4	160	154	157	158	157	159	158	158	158	160	155	156	...	153
5	156	153	155	159	159	155	156	155	155	157	155	154	...	151
6	155	155	155	157	156	159	152	158	156	158	152	153	...	153
7	156	153	157	156	153	155	154	155	157	156	155	156	...	149
8	159	159	156	158	156	159	157	161	162	157	157	159	...	147
9	158	155	158	154	156	160	162	155	159	161	156	161	...	147
10	155	154	157	158	160	160	159	160	158	161	160	160	...	153
11	154	157	157	157	156	155	159	154	159	158	161	158	...	155
12	152	150	155	154	152	156	157	156	157	154	157	159	...	148
13	157	153	156	155	157	160	160	157	159	159	160	161	...	151
14	151	154	157	156	156	158	158	156	157	159	158	156	...	148
15	156	157	157	160	159	159	156	158	159	162	161	160	...	150
16	157	158	159	157	157	154	153	158	159	155	160	159	...	156
17	154	154	156	157	158	159	157	160	158	158	156	157	...	152
18	151	153	157	152	156	156	155	156	157	157	155	157	...	152
19	153	155	154	153	156	155	153	155	153	155	154	156	...	155
20	152	154	152	156	159	154	156	155	161	157	157	161	...	151
21	154	157	155	156	157	154	158	158	158	158	158	162	...	155
22	155	153	155	155	159	160	159	161	158	159	160	161	...	150
23	151	151	153	155	153	156	155	155	157	156	157	156	...	156
24	150	151	155	154	155	154	156	152	158	157	158	159	...	157
25	153	154	151	155	154	153	155	157	158	157	157	157	...	153
26	154	154	155	156	155	156	155	156	158	154	159	161	...	155
27	162	157	155	154	156	155	156	157	155	161	157	161	...	151
28	158	155	156	157	160	157	157	162	157	160	158	163	...	154
29	161	157	158	157	159	156	156	157	160	159	162	159	...	153
30	158	159	163	157	158	155	163	159	158	158	162	162	...	156
31	154	154	156	156	159	155	156	159	157	159	159	157	...	148
32	152	155	155	156	158	155	157	160	161	158	161	161	...	147
33	154	155	156	155	156	154	160	154	156	160	157	161	...	153
34	155	158	160	155	159	160	156	158	162	164	162	163	...	153
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
255	121	126	130	162	102	160	172	133	150	164	147	116	...	113

LAMPIRAN B. *Keystream random* 10-bit

Ukuran matriks : 256x256

	1	2	3	4	5	6	7	8	9	10	11	12	...	256
1	**681**	**642**	**741**	629	641	578	732	547	674	752	699	751	…	664
2	**552**	**714**	**546**	592	653	692	762	605	534	547	733	618	...	710
3	**663**	**664**	**523**	623	697	624	647	650	592	718	610	573	...	691
4	652	634	719	695	696	582	719	546	632	540	753	637	...	646
5	650	527	727	558	730	755	619	728	717	623	588	728	...	715
6	688	606	555	520	566	520	534	700	675	755	717	594	...	728
7	630	680	745	704	754	592	525	629	684	692	569	619	...	591
8	520	512	749	724	598	540	647	543	540	707	588	734	...	574
9	537	594	601	563	732	610	738	539	613	688	565	535	...	523
10	585	764	654	530	759	582	672	621	520	767	752	559	...	753
11	681	525	591	598	604	566	712	542	677	536	618	748	...	703
12	608	763	718	729	624	605	539	586	763	655	670	702	...	540
13	673	733	699	741	701	635	621	528	560	556	527	713	...	617
14	675	704	523	633	756	626	701	588	703	749	740	680	...	525
15	570	693	536	720	650	632	675	685	533	562	659	706	...	622
16	602	568	592	756	708	728	588	527	716	556	625	767	...	559
17	561	620	758	521	734	741	579	602	694	680	516	737	...	696
18	639	652	646	683	744	741	704	686	752	520	747	524	...	641
19	712	569	558	680	563	583	655	612	531	586	567	747	...	548
20	682	700	666	514	693	734	590	737	656	551	630	526	...	675
21	552	627	536	733	536	755	522	599	545	667	718	636	...	632
22	710	699	576	517	600	657	631	530	690	522	666	633	...	755
23	736	664	633	601	673	665	761	684	753	762	637	580	...	717
24	578	762	618	644	678	714	699	717	745	625	579	621	...	520
25	668	597	653	743	524	546	767	543	630	683	739	549	...	744
26	708	570	755	708	752	585	635	718	526	648	715	605	...	697
27	647	536	624	573	569	708	657	747	567	663	573	700	...	714
28	660	694	730	632	758	562	609	752	565	572	550	660	...	605
29	684	734	575	637	530	609	649	636	681	751	542	733	...	714
30	560	628	739	671	622	766	519	674	730	680	687	550	...	665
31	599	568	614	561	633	737	654	534	720	575	687	756	...	631
32	649	634	554	633	537	613	709	700	720	636	692	692	...	562
33	703	519	595	654	637	552	561	577	606	578	717	589	...	646
34	627	750	518	698	544	566	652	731	667	749	627	765	...	708
35	576	731	707	531	677	652	677	529	625	593	657	567	…	700
36	523	582	553	595	607	646	528	681	620	597	670	609	…	591
37	592	561	560	728	537	517	666	540	584	682	551	740	…	593
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
255	616	675	719	575	679	623	512	549	595	729	690	655	…	675

LAMPIRAN C. Derajat keabuan *key* citra

Ukuran matriks : 256x256

	1	2	3	4	5	6	7	8	9	10	11	12	...	256
1	**84**	**127**	**24**	136	124	191	33	222	95	13	70	18	…	101
2	**213**	**55**	**223**	173	112	73	7	160	235	222	32	151	...	59
3	**106**	**101**	**246**	146	68	141	122	119	173	51	159	192	...	78
4	113	135	50	74	69	187	50	223	133	225	12	128	...	123
5	119	242	42	211	39	14	150	37	48	146	177	37	...	54
6	77	163	214	245	203	245	235	65	94	14	48	175	...	37
7	139	85	20	61	15	173	240	136	81	73	196	150	...	178
8	245	253	16	41	171	225	122	226	225	62	177	35	...	195
9	228	175	164	206	33	159	31	230	152	77	200	234	...	246
10	180	1	115	239	10	187	93	144	245	2	13	210	...	12
11	84	240	178	171	161	203	53	227	88	229	151	17	...	66
12	157	6	51	36	141	160	230	183	6	114	99	67	...	225
13	92	32	70	24	64	134	144	237	205	209	242	52	...	148
14	94	61	246	132	9	143	64	177	66	16	25	85	...	240
15	199	72	229	45	119	133	94	80	232	207	110	63	...	147
16	167	197	173	9	57	37	177	242	49	209	140	2	...	210
17	204	145	11	244	35	24	190	167	75	85	249	28	...	69
18	130	113	123	86	21	24	61	83	13	245	22	241	...	124
19	53	196	211	85	206	186	114	153	238	183	202	22	...	217
20	87	65	103	255	72	35	179	28	109	218	139	243	...	94
21	213	142	229	32	229	14	247	170	220	102	51	129	...	133
22	59	70	189	248	165	108	138	239	79	247	103	132	...	14
23	29	101	132	164	92	100	4	81	12	7	128	185	...	48
24	191	7	151	121	91	55	70	48	20	140	190	144	...	245
25	97	168	112	26	241	223	2	226	139	86	30	216	...	21
26	57	199	14	57	13	180	134	51	243	117	54	160	...	68
27	122	229	141	192	196	57	108	22	202	106	192	65	...	55
28	105	75	39	133	11	207	156	13	200	193	219	105	...	160
29	81	35	194	128	239	156	116	129	84	18	227	32	...	55
30	205	137	30	98	147	3	250	95	39	85	82	219	...	100
31	170	197	155	204	132	28	115	235	45	194	82	9	...	138
32	116	135	215	132	228	152	56	65	45	129	73	73	...	207
33	66	250	174	115	128	213	204	188	163	191	48	176	...	123
34	142	19	251	71	221	203	113	38	102	16	142	0	...	57
35	189	38	62	238	88	113	88	236	140	172	108	202	…	65
36	246	187	212	174	162	123	237	84	145	168	99	156	…	178
37	173	204	205	37	228	248	103	225	181	87	218	25	…	172
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
256	149	94	50	194	90	146	253	216	174	36	79	114	...	94

LAMPIRAN D. Derajat keabuan hasil enkripsi

Ukuran matriks : 256x256

	1	2	3	4	5	6	7	8	9	10	11	12	...	256
1	**144**	**143**	**252**	115	35	165	101	161	126	17	28	49	…	189
2	**49**	**37**	**86**	125	228	229	101	81	77	42	233	163	...	112
3	**162**	**20**	**53**	244	158	80	3	116	229	141	122	148	...	34
4	152	40	214	12	168	216	204	104	189	177	219	31	...	214
5	165	25	89	190	48	89	74	165	100	58	13	16	...	246
6	101	152	151	192	163	21	134	182	98	190	158	176	...	75
7	229	141	175	145	217	154	55	115	37	158	212	74	...	212
8	21	37	159	144	101	249	214	57	112	25	163	245	...	144
9	50	216	16	33	30	69	84	22	243	171	31	159	...	249
10	1	81	53	13	236	69	13	255	222	247	216	73	...	193
11	114	204	250	176	51	42	205	156	129	223	119	189	...	127
12	208	144	181	57	209	223	217	35	150	29	152	23	...	126
13	175	162	225	54	65	15	255	99	77	124	201	130	...	152
14	20	17	177	223	56	127	30	54	241	129	189	17	...	127
15	32	231	229	88	59	77	98	94	164	45	218	45	...	61
16	215	95	219	168	165	16	230	112	64	57	89	152	...	161
17	99	208	168	129	32	179	177	206	160	118	54	169	...	253
18	26	192	151	97	179	57	90	232	168	192	25	37	...	55
19	11	212	244	141	111	116	49	27	58	176	36	98	...	141
20	224	51	205	35	12	101	198	155	43	58	48	213	...	20
21	31	25	252	178	229	251	92	147	35	156	141	194	...	40
22	181	177	114	228	13	23	180	183	101	23	8	194	...	196
23	197	109	10	36	202	209	169	21	100	174	239	208	...	31
24	64	219	112	106	231	37	225	158	145	233	19	1	...	192
25	77	16	33	116	182	23	217	89	191	153	153	202	...	139
26	120	28	89	248	119	94	244	238	49	179	24	207	...	125
27	172	229	26	242	57	55	119	63	3	183	105	200	...	183
28	34	194	46	227	173	87	1	19	239	49	37	5	...	16
29	237	62	245	239	79	60	208	38	240	49	114	140	...	187
30	207	155	154	113	37	48	81	15	165	118	30	237	...	209
31	18	30	224	223	1	155	104	15	104	214	151	168	...	2
32	145	24	214	223	50	118	228	29	173	125	206	206	...	85
33	221	102	169	23	31	31	191	144	227	197	34	234	...	214
34	244	13	195	87	165	195	248	204	94	87	124	108	...	192
35	55	204	25	127	210	152	142	239	208	254	106	95	…	205
36	23	19	60	18	52	151	182	55	232	240	218	215	…	55
37	144	1	227	46	119	35	150	177	37	52	65	50	…	124
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
256	185	187	4	51	91	207	184	71	172	82	56	189	...	238

LAMPIRAN E. Derajat keabuan hasil dekripsi

Ukuran matriks : 256x256

	1	2	3	4	5	6	7	8	9	10	11	12	...	256
1	**156**	**159**	**158**	155	158	156	159	158	157	158	158	159	...	152
2	**160**	**154**	**157**	158	157	159	158	158	158	160	155	156	...	153
3	**156**	**159**	**158**	155	158	156	159	158	157	158	158	159	...	152
4	160	154	157	158	157	159	158	158	158	160	155	156	...	153
5	156	153	155	159	159	155	156	155	155	157	155	154	...	151
6	155	155	155	157	156	159	152	158	156	158	152	153	...	153
7	156	153	157	156	153	155	154	155	157	156	155	156	...	149
8	159	159	156	158	156	159	157	161	162	157	157	159	...	147
9	158	155	158	154	156	160	162	155	159	161	156	161	...	147
10	155	154	157	158	160	160	159	160	158	161	160	160	...	153
11	154	157	157	157	156	155	159	154	159	158	161	158	...	155
12	152	150	155	154	152	156	157	156	157	154	157	159	...	148
13	157	153	156	155	157	160	160	157	159	159	160	161	...	151
14	151	154	157	156	156	158	158	156	157	159	158	156	...	148
15	156	157	157	160	159	159	156	158	159	162	161	160	...	150
16	157	158	159	157	157	154	153	158	159	155	160	159	...	156
17	154	154	156	157	158	159	157	160	158	158	156	157	...	152
18	151	153	157	152	156	156	155	156	157	157	155	157	...	152
19	153	155	154	153	156	155	153	155	153	155	154	156	...	155
20	152	154	152	156	159	154	156	155	161	157	157	161	...	151
21	154	157	155	156	157	154	158	158	158	158	158	162	...	155
22	155	153	155	155	159	160	159	161	158	159	160	161	...	150
23	151	151	153	155	153	156	155	155	157	156	157	156	...	156
24	150	151	155	154	155	154	156	152	158	157	158	159	...	157
25	153	154	151	155	154	153	155	157	158	157	157	157	...	153
26	154	154	155	156	155	156	155	156	158	154	159	161	...	155
27	162	157	155	154	156	155	156	157	155	161	157	161	...	151
28	158	155	156	157	160	157	157	162	157	160	158	163	...	154
29	161	157	158	157	159	156	156	157	160	159	162	159	...	153
30	158	159	163	157	158	155	163	159	158	158	162	162	...	156
31	154	154	156	156	159	155	156	159	157	159	159	157	...	148
32	152	155	155	156	158	155	157	160	161	158	161	161	...	147
33	154	155	156	155	156	154	160	154	156	160	157	161	...	153
34	155	158	160	155	159	160	156	158	162	164	162	163	...	153
35	159	158	157	157	160	160	158	157	158	162	161	161	...	153
36	155	158	156	158	159	157	156	158	159	160	162	162	...	151
37	156	159	157	157	154	154	159	160	159	158	162	159	...	152
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
256	121	126	130	162	102	160	172	133	150	164	147	116	...	113

LAMPIRAN F. Skrip Program Proses Enkripsi

Enkripsi.m

```
y=imread('plain_image.jpg');
[z zz zzz]=size(y);
ksds=input('masukkan desimal : ');
bin=de2bi(ksds,10,'left-msb');
%--------------------------------
%pembangkitan kunci
P10 = [3 5 2 7 4 10 1 9 8 6];
P8 = [6 3 7 4 8 5 10 9];
%--------------------------------
%S-DES
IP = [2 6 3 1 4 8 5 7];
EP = [4 1 2 3 2 3 4 1];
s0 = [1 0 3 2; 3 2 1 0; 0 2 1 3; 3 1 3 2];
s1 = [0 1 2 3; 2 0 1 3; 3 0 1 0; 2 1 0 3];
P = [2 4 3 1];
IPI = [4 1 3 5 7 2 8 6];
%--------------------------------
for k=1:zzz
    for j=1:zz
        for i=1:z
            for m=1:10
                kunci(m)=round(rand(1));
            end
            kk=xor(bin,kunci);
            kkk=bi2de(kk,'left-msb');
            B(i,j,k)=kkk;
        end
    end
end
imwrite(mat2gray(B,[0 1023]),'key_citra.jpg');

B=imread('citra.tif');
for q=1:zzz
for qq=1:zz
for qqq=1:z
%----------------------
key = B(qqq,qq,q);
key = de2bi(key,10,'left-msb');

key = key(P10);
L = key(1:5);
R = key(6:10);

LSL1 = [L(2:5) L(1)];
LSR1 = [R(2:5) R(1)];
```

```
K1 = [LSL1,LSR1];
K1 = K1(P8);

LSL2 = [LSL1(3:5) LSL1(1:2)];
LSR2 = [LSR1(3:5) LSR1(1:2)];

K2 = [LSL2,LSR2];
K2 = K2(P8);

K = [K1,K2];
%----------------------
in=de2bi(y(qqq,qq,q),8,'left-msb');

LR0 = in(IP);

L0 = LR0(1:4);
R0 = LR0(5:8);

j = 0;
for s = 1: 2

ER0 = R0(EP);

a = xor(ER0,K(1+j*2:8+j*2));

u = s0((a(2)*2 + a(3))*4 + (a(1)*2 + a(4)) + 1);
v = s1((a(6)*2 + a(7))*4 + (a(5)*2 + a(8)) + 1);
w = [floor(u/2) rem(u,2) floor(v/2) rem(v,2)];

b = w(P);

R1(1+j:4+j) = xor(L0,b);

L0 = R0;
R0 = R1;
j = j+4;
end

R2L2 = [R1(5:8) R1(1:4)];

op = R2L2(IPI);
op = bi2de(op,'left-msb');

H(qqq,qq,q)=op;
end
end
end

imwrite(mat2gray(H,[0 255]),'cipher_image.tif');
```

LAMPIRAN G. Skrip Program Proses Dekripsi

Dekripsi.m

```
y=imread('cipher_image.jpg');
[z zz zzz]=size(y);
B=imread('key_citra.jpg');
%------------------------------
%pembangkitan kunci

P10 = [3 5 2 7 4 10 1 9 8 6];
P8 = [6 3 7 4 8 5 10 9];
%------------------------------
%S-DES

IP = [2 6 3 1 4 8 5 7];
EP = [4 1 2 3 2 3 4 1];
s0 = [1 0 3 2; 3 2 1 0; 0 2 1 3; 3 1 3 2];
s1 = [0 1 2 3; 2 0 1 3; 3 0 1 0; 2 1 0 3];
P = [2 4 3 1];
IPI = [4 1 3 5 7 2 8 6];
%------------------------------

for q=1:zzz
for qq=1:zz
for qqq=1:z

%----------------------
key = de2bi(B(qqq,qq,q),10,'left-msb');

key = key(P10);
L = key(1:5);
R = key(6:10);

LSL1 = [L(2:5) L(1)];
LSR1 = [R(2:5) R(1)];

K1 = [LSL1,LSR1];
K1 = K1(P8);

LSL2 = [LSL1(3:5) LSL1(1:2)];
LSR2 = [LSR1(3:5) LSR1(1:2)];

K2 = [LSL2,LSR2];
K2 = K2(P8);

K = [K2,K1];
%----------------------
```

```
in=de2bi(y(qqq,qq,q),8,'left-msb');

LR0 = in(IP);

L0 = LR0(1:4);
R0 = LR0(5:8);

j = 0;
for s = 1: 2

ER0 = R0(EP);

a = xor(ER0,K(1+j*2:8+j*2));

u = s0((a(2)*2 + a(3))*4 + (a(1)*2 + a(4)) + 1);
v = s1((a(6)*2 + a(7))*4 + (a(5)*2 + a(8)) + 1);
w = [floor(u/2) rem(u,2) floor(v/2) rem(v,2)];

b = w(P);

R1(1+j:4+j) = xor(L0,b);

L0 = R0;
R0 = R1;
j = j+4;
end

R2L2 = [R1(5:8) R1(1:4)];

op = R2L2(IPI);
op = bi2de(op,'left-msb');

H(qqq,qq,q)=op;
end
end
end

imwrite(mat2gray(H,[0 255]),'plain_image.tif');
```

LAMPIRAN H. Skrip Program Analisis dengan Diferensial

NPCR_AD.m

```
a=imread('cameraman.tif');           %Plain Image
b=imread('Cip_Cam_170423_K765.tif'); %Cipher Image
[c d e]=size(a);

for i=1:e
    for j=1:c
        for k=1:d
            if a(j,k,i)==b(j,k,i);
                beda(j,k,i)=0;
            else
                beda(j,k,i)=1;
            end
        end
    end
end
c=(c*d*e)-sum(sum(sum(beda)));
imwrite(mat2gray(beda,[0 1]),'AD.tif');
npcr=(sum(sum(sum(beda)))/(c*d*e))*100
```