PENGAMANAN CITRA BINER DENGAN ALGORITMA *DATA ENCRYPTION STANDARD* (*DES*) DAN *BERNOULLI MAP*

SKRIPSI

Oleh

Silmi Maulida
NIM 141810101062

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018

PENGAMANAN CITRA BINER DENGAN ALGORITMA *DATA ENCRYPTION STANDARD* (*DES*) DAN *BERNOULLI MAP*

SKRIPSI

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

Oleh

Silmi Maulida
NIM 141810101062

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018

PERSEMBAHAN

Dengan menyebut nama Allah SWT yang Maha Pengasih dan Maha Penyayang serta Sholawat serta salam kepada Nabi Muhammad SAW dengan kerendahan hati, skripsi ini saya persembahkan untuk :

1. Kedua orang tuaku tercinta Alm. Bapak Afandi dan Ibu Elya, kakakku Chintiya Nur Faisma dan seluruh keluarga yang telah mendoakan, memberi kasih sayang serta semangat selama perjalanan tugas akhir ini;
2. Abduh Riski, S.Si, M.Si selaku Dosen Pembimbing Utama dan Ahmad Kamsyakawuni, S.Si, M.Kom selaku Dosen Pembimbing Anggota yang telah membimbing dan menyempurnakan tugas akhir ini;
3. Almamater tercinta jurusan Matematika FMIPA Universitas Jember, SMK N Ihya Ulumudin Singojuruh, SMP Negeri 1 Genteng, SD Negeri 1 Genteng Wetan dan TK Fajar;
4. Teman-teman Extreme dan teman-teman kos Jawa 2B No.7B yang selalu memberikan semangat, dukungan, dan doa;
5. Semua pihak yang membantu penulis dalam menyelesaikan tugas akhir.

MOTTO

"Menyesali nasib tidak akan mengubah keadaan. Terus berkarya dan bekerjalah yang membuat kita BERHARGA"[1]

"*Sebaik-baik manusia adalah mereka yang bermanfaat bagi orang lain*"[2]

[1] Gusdur

[2] HR. Ahmad, ath-Thabrani

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama: Silmi Maulida

nim : 141810101062

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Pengamanan Citra Biner dengan Algoritma *Data Encryption Standard* (*DES*) dan *Bernoulli Map*" 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, November 2018

Yang menyatakan,

Silmi Maulida

NIM 141810101062

SKRIPSI

PENGAMANAN CITRA BINER DENGAN ALGORITMA *DATA ENCRYPTION STANDARD* (*DES*) DAN *BERNOULLI MAP*

Oleh

Silmi Maulida
NIM 1410101062

Pembimbing

Dosen Pembimbing Utama : Abduh Riski, S.Si, M.Si

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

PENGESAHAN

Skripsi berjudul "Pengamanan Citra Biner dengan Algoritma *Data Encryption Standard* (*DES*) dan *Bernoulli Map*" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Tim Penguji:

Ketua,	Anggota I,
Abduh Riski, S.Si, M.Si	Ahmad Kamsyakawuni, S.Si, M.Kom
NIP. 199004062015041001	NIP. 197211291998021001
Anggota II,	Anggota III,
Kusbudiono, S.Si., M.Si	Dr. Firdaus Ubaidillah, S.Si., M.Si
NIP. 197704302005011001	NIP. 197006061998031003

Mengesahkan
Dekan,

Drs. Sujito, Ph.D.

NIP. 196102041987111001

RINGKASAN

Pengamanan Citra Biner dengan Algoritma *Data Encryption Standard* (*DES*) dan *Bernoulli Map*; Silmi Maulida, 141810101062; 2018; 49 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyimpanan suatu data pada saat ini tidak lagi dalam bentuk dokumen-dokumen yang dicetak, melainkan dalam bentuk sebuah data yang disimpan di dalam komputer maupun ponsel. Keamanan data atau informasi saat ini diragukan oleh user atau pengguna. Untuk melindungi data tersebut dari pihak yang tidak berwenang, dibutuhkan suatu perlindungan.

Kriptografi adalah ilmu untuk menjaga kerahasiaan data dengan cara mengubahnya menjadi bentuk yang tidak lagi dipahami maknanya. Salah satu algoritma kriptografi yang digunakan dalam penelitian ini adalah DES (*Data Encryption Standard*). Namun, untuk melindungi suatu data seperti citra, algoritma DES tidaklah aman karena algoritma ini menggunakan kunci yang relatif pendek. Untuk mengatasi kelemahan ini, maka akan dilakukan suatu pembangkitan kunci sebelum kemudian dienkripsi lebih lanjut dengan menggunakan algoritma *Bernoulli Map*. *Bernoulli Map* mengenkripsi citra dengan cara memberikan 2 angka kunci rahasia yang dibangkitkan lalu dienkripsi dengan tujuan mempersulit orang yang tidak berhak untuk mengetahui isi data atau informasinya. Diperoleh 2^n kemungkinan kunci tertebak dimana n adalah ukuran atau *size* dari citra.

Data yang digunakan dalam penelitian ini adalah data berupa citra biner yang digunakan sebagai *plain image*. Citra tersebut kemudian dienkripsi dengan menggunakan algoritma DES dengan pembangkit kunci dari algoritma *Bernoulli Map*, setelah itu dienkripsi menggunakan Bernoulli Map.

Analisis keamanan dari algoritma yang diajukan menunjukkan bahwa tingkat keamanannya yang baik dan meningkat untuk hasil citra enkripsi dan dapat dikatakan aman karena butuh $2^{56}+2^{512}$ kali percobaan untuk menerobos kunci yang ada. Tidak hanya itu dengan membangkitkan kunci menggunakan *Bernoulli Map* dapat memanipulasi hacker yang tidak mengetahui ada algoritma di dalam pembangkitan kunci.

PRAKATA

Puji syukur kehadirat Allah SWT Tuhan yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul "Pengamanan Citra Biner dengan Algoritma *Data Encryption Standard* (*DES*) dan *Bernoulli Map*". Tugas akhir ini disusun untuk memenuhi salahsatu syarat pada program pendidikan strata satu (S1) Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

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

1. Abduh Riski, S.Si, M.Si selaku Dosen Pembimbing Utama dan Ahmad Kamsyakawuni, S.Si, M.Kom yang telah membimbing hingga selesainya tugas akhir ini;
2. Kusbudiono, S.Si., M.Si selaku Dosen Penguji 1 dan Dr. Firdaus Ubaidillah, S.Si., M.Si selaku Dosen Penguji II yang telah memberikan kritik dan saran yang membangun demi kesempurnaan tugas akhir ini;
3. Dr. Alfian Futuhul Hadi, S.Si., M.Si selaku Dosen Pembimbing Akademik yang selalu memberikan pengarahan selama penulis menjadi mahasiswa;
4. Seluruh Dosen dan Karyawan Jurusan Matematika FMIPA Universitas Jember yang telah memberikan ilmu selama dalam perkuliahan ini;
5. Orang tua tercinta serta keluarga yang selalu memberikan dukungan dan doa;
6. Teman-teman Extreme dan Kos Jawa 2B no. 7B yang telah memberikan banyak kenangan dan dukungan.
7. Semua pihak yang tidak dapat disebutkan satu per satu

Semoga bimbingan, bantuan, dan dorongan beliau dibalas berlipat ganda oleh Tuhan yang Maha Esa. Selain itu, penulis juga menerima segala kritik dan saran dari semua pihak demi kesempuranaan penyusunan tugas akhir ini. Akhirnya penulis berharap, semoga tugas akhir ini dapat bermanfaat.

Jember, November 2018 Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Teknologi era modern berkembang sangat cepat, saat ini hampir semua kalangan dari berbagai macam aspek yang berbeda menggunakan internet sebagai sarana komunikasi satu sama lain. Keamanan data atau informasi saat ini diragukan oleh user atau pengguna. Banyak pengguna baik individu, lembaga pendidikan, perusahaan maupun instansi lainnya tidak ingin data atau informasinya diketahui orang yang tidak berhak mengetahuinya. Data atau informasi yang tidak diamankan akan berisiko terhadap pencurian data atau informasi, oleh karena itu dikembangkanlah cabang ilmu yang mempelajari tentang cara-cara pengamanan data atau informasi salah satunya dikenal dengan istilah kriptografi.

Kriptografi meliputi dua proses utama yaitu enkripsi dan dekripsi (Scheiner, 1996). Proses menyandikan plainteks (data asli) menjadi cipherteks (data sandi) disebut enkripsi (*encryption*) atau *enciphering*, sedangkan proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi (*decryption*) atau deciphering. Penyandian pesan bertujuan agar kerahasiaan pesan lebih terjaga dan pesan yang dikirim tidak mudah diketahui oleh orang lain.

DES (*Data Encryption Standard*) merupakan sebuah algoritma untuk mengenkripsi data yang memiliki plainteks 64bit dan menggunakan kunci 56-bit dan 8bit sebagai paritas bit yang digunakan mengecek terjadinya kesalahan pada pembacaan bit. Selanjutnya blok plainteks dipermutasi dengan *initial permutation* atau IP, hasil permutasi awal di-*enciphering* sebanyak 16 putaran dengan menggunakan 8 *S-Box* di dalam prosesnya, kemudian dibalik dengan *invers initial permutation* atau IP^{-1} sehingga menjadi blok chiperteks. Panjang kunci 56-bit maka memiliki kemungkinan 2^{56} kali percobaan untuk menemukan kunci. DES dianggap tidak aman lagi, sejak tahun 1999 ditemukan kunci DES dengan menggunakan pencarian *brute force* yang memakan waktu 22 jam 15 menit, padahal awalnya diasumsikan bahwa pencarian *brute force* minimal membutuhkan waktu 1142 tahun untuk menemukan kunci yang benar (Ilie, 2011).

Penelitian selanjutnya meningkatkan algoritma DES menjadi 3DES (Triple DES) dalam algoritma ini metode enkripsi mirip dengan yang ada di DES asli namun diaplikasikan 3 kali dengan menggunakan ketiga kuncinya sama atau 2 kunci berbeda atau 3 kunci berbeda sehingga didapatkan 2^{112} atau 2^{168} kali percobaan untuk menemukan kunci tersebut (Kumar dan Rajaana, 2016). Penelitian selanjutnya meningkatkan algoritma di DES yaitu improvisasi algoritma 3DES (Triple DES) dengan bilangan irasional. Algoritma ini memberikan angka irasional sebagai kunci, misal $\sqrt{2} = 1.41421356237309504880168872420 97$ sehingga untuk menghasilkan kuncinya diambil dari bilangan tersebut. Kelemahan algoritma ini tetap pada panjang kunci yaitu 2^{56} yang diulang sebanyak tiga kali (Singh dan Alam, 2015).

Penelitian Ahmed dan El-aziem (2014) membandingkan hasil enkripsi beberapa algoritma dari *teori chaos* yaitu *Bernoulli Map, Genhous Map* dan *Logistic Map* dimana dihasilkan NPCR dari *Bernoulli Map* paling kecil diantara ketiga algoritma tersebut dengan nilai 99,21% sedangkan *Genhous Map* 99,23% dan *Logistic Map* 99,33%. Penelitian selanjutnya Laura, dkk. (2017) mengenkripsi citra menggunakan algoritma *Bernoulli Map* yang merupakan cabang dari *teori chaos. Bernoulli Map* mengenkripsi citra dengan cara memberikan 2 angka kunci rahasia yang dibangkitkan lalu dienkripsi dengan tujuan mempersulit orang yang tidak berhak untuk mengetahui isi data atau informasinya, diperoleh 2^n kemungkinan kunci tertebak dimana *n* adalah ukuran (*size*) citra. Penelitian kali ini penulis akan menggunakan algoritma DES dan algoritma *Bernoulli Map* dengan pembangkitan kunci dari algoritma *Bernoulli Map* yaitu memberikan 2 bilangan rahasia untuk membangkitkan kunci menggunakan algoritma *Bernoulli Map* selanjutnya pengenkripsian menggunakan algoritma DES. Hasil enkripsi citra menggunakan DES akan dienkripsi lagi dengan algoritma *Bernoulli*, sehingga algoritma ini dapat mengelabui *attacker* saat memecahkan kunci dan hasil citra yang terenkripsi memiliki perbedaan yang signifikan dari citra aslinya.

1.2 Rumusan Masalah

Adapun rumusan masalah yang dibahas dalam penelitian meliputi:

a. Bagaimanakah penerapan algoritma *Bernoulli Map* untuk pembangkitan *keystream* yang akan digunakan pada algoritma DES?
b. Bagaimanakah penerapan enkripsi dan dekripsi algoritma DES dengan algoritma *Bernoulli Map* ?
c. Bagaimanakah hasil penerapan enkripsi dan dekripsi algoritma DES dengan *Bernoulli Map* ?
d. Bagaimana analisis keamanan dari algoritma DES dan *Bernoulli Map*?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

a. Untuk mengetahui proses enkripsi dan dekripsi citra biner dari algoritma DES dengan *Bernoulli Map*.
b. Untuk mengetahui hasil yang diperoleh dari enkripsi algoritma DES dengan *Bernoulli Map* pada citra biner.
c. Mengetahui tingkat keamanan dari algoritma DES dengan *Bernoulli Map* pada citra biner.

1.4 Manfaat Penelitian

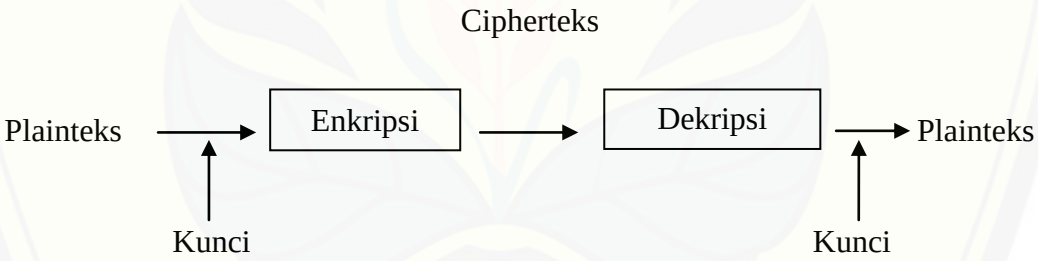
Adapun manfaat yang diperoleh dari penelitian ini adalah:

a. Mengetahui proses enkripsi dan dekripsi algoritma DES dengan *Bernoulli Map*.
b. Mendapatkan metode baru untuk meningkat algoritma DES
c. Menambah pengetahuan baru dalam kriptografi.
d. Mengetahui tingkat keamanan dari algoritma DES dengan *Bernoulli Map*.

BAB 2. TINJAUAN PUSTAKA

2.1 Kriptografi

Kata kriptografi berasal dari bahasa Yunani yaitu "*cryptos"* yang berarti rahasia dan "*graphein"* yang berarti tulisan. Jadi kriptografi berarti tulisan rahasia (Ariyus, 2008). Menurut Munir (2006) kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya, menjaga kerahasiaan data dengan kriptografi, data sederhana yang dikirim (plainteks) diubah ke dalam bentuk data sandi (cipherteks), kemudian data sandi tersebut hanya dapat dikembalikan ke bentuk data sebenarnya hanya dengan menggunakan kunci (*key*) tertentu yang dimiliki oleh pihak yang sah saja. Tentunya hal ini menyebabkan pihak lain yang tidak memiliki kunci tersebut tidak akan dapat membaca data yang sebenarnya sehingga dengan kata lain data akan tetap terjaga.



Gambar 2.1 Proses Enkripsi dan Dekripsi

2.1.1 Tujuan Kriptografi

Kriptografi bertujuan untuk memberikan layanan pada aspek-aspek keamanan antara lain (Menez dkk, 1996) :

a. Kerahasiaan (*confidentiality*), yaitu menjaga supaya pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
b. Integritas data (*data integrity*), yaitu memberikan jaminan bahwa untuk tiap bagian pesan tidak akan mengalami perubahan dari saat data dibuat/dikirim oleh pengirim sampai dengan saat data tersebut dibuka oleh penerima data.

c. Otentikasi (*authentication*), yaitu berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi maupun mengidentifikasi kebenaran sumber pesan
d. Non-repudiasi yaitu memberikan cara untuk membuktikan bahwa suatu dokumen datang dari seseorang tertentu sehingga apabila ada seseorang yang mencoba mengakui memiliki dokumen tersebut, dapat dibuktikan kebenarannya dari pengakuan orang tersebut.

2.1.2 Terminologi Kriptografi

Di dalam kriptografi sering ditemukan berbagai istilah atau *terminology*. Beberapa istilah yang harus diketahui adalah sebagai berikut (Schneiner, 1996):

a. Pengiriman dan Penerimaan pesan

   Seorang pengirim pesan (*sender*) ingin mengirim pesan kepada seorang penerima pesan (*receiver*). Pengirim menginginkan pesan dapat dikirim secara aman, yaitu ia yakin bahwa pihak lain tidak dapat membaca isi pesan.
b. Pesan, *Plaintext*, dan *Ciphertext*

   Pesan adalah data atau informasi yang dapat dimengerti maknanya. Nama lain untuk pesan adalah *Plaintext* (Plainteks). Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan disandikan ke bentuk lain. Bentuk pesan yang tersandi disebut *Ciphertext* (Ciperteks).
c. Enkripsi dan Dekripsi

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

   Kriptografi mengatasi masalah keamanan data dengan menggunakan kunci, yang dalam hal ini algoritma tidak dirahasiakan lagi, tetapi kunci harus tetap dijaga kerahasiaannya. Kunci (*key*) adalah parameter yang digunakan untuk transformasi enkripsi dan dekripsi.

2.1.3 Jenis Algoritma Kriptografi

Berdasarkan kunci yang digunakan untuk enkripsi dan dekripsi, kriptografi dapat dibedakan menjadi dua macam, yaitu kriptografi simetri (*symetric cryptography*) dan kriptografi asimetri (*asymetric cryptography*). Sistem kriptografi simetri, kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi. Keamanan sistem kriptografi simetri terletak pada kerahasiaan kunci. Istilah lain untuk kriptografi simetri adalah kriptografi kunci privat (*private key cryptography*) atau kriptografi konvensional (*conventional cryptography*). Sistem kriptografi asimetri, kunci untuk proses enkripsi tidak sama dengan kunci untuk proses dekripsi. Istilah lain untuk kriptografi asimetri adalah kriptografi kunci publik (*public key cryptography*), sebab kunci untuk enkripsi tidak rahasia dan dapat diketahui oleh siapapun, sementara kunci untuk dekripsi hanya diketahui oleh penerima pesan (Munir, 2006).

2.2 Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpan. (Sutoyo dkk., 2009) . Menurut arti secara harfiah, citra (*image*) adalah gambar pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya. Pantulan cahaya ini ditangkap oleh alat-alat optik, seperti mata pada manusia, kamera, pemindai (*scanner*), dan lain-lain sehingga bayangan objek dalam bentuk citra dapat terekam. Citra sebagai output dari suatu sistem perekaman data dapat bersifat:

a. Optik, berupa foto
b. Analog berupa sinyal video, seperti gambar pada monitor televisi.
c. Digital yang dapat langsung disimpan pada suatu pita magnetik.

Citra dapat dikelompokkan menjadi dua bagian yaitu citra diam (*still image*) dan citra bergerak (*moving image*). Citra diam adalah citra tunggal yang tidak bergerak. Sedangkan citra bergerak adalah rangkaian citra diam yang ditampilkan secara beruntun (sekuensial) sehingga memberi kesan pada mata sebagai gambar yang bergerak. Setiap citra di dalam rangkaian itu disebut frame. Gambar-gambar yang tampak pada film layar lebar atau televisi pada hakikatnya terdiri dari ratusan sampai ribuan frame. (Sawaluddin dkk., 2006) .

Komputer merupakan alat yang beroperasi dalam sistem digital yang menggunakan bit atau byte dalam pengukuran datanya. Komputer menggunakan sistem bilangan biner dalam pemecahan masalah ini. Penggunaan sistem bilangan biner ini, citra dapat diproses dalam komputer dengan sebelumnya mengekstrak informasi citra analog asli dan mengirimkannya ke komputer dalam bentuk biner. Proses ini disebut dengan digitalisasi. Digitalisasi dapat dilakukan oleh alat seperti kamera digital atau scanner. Kedua alat ini selain dapat mengambil atau menangkap sebuah citra, juga dapat bertindak sebagai alat input (masukan) bagi komputer. Alat penangkap citra digital ini dapat menyediakan aliran data biner bagi komputer yang didapatkan dari pembacaan tingkat kecerahan pada sebuah citra asli dalam interval sumbu *x* dan sumbu *y*.

Citra digital merupakan fungsi intensitas cahaya *f*(*x*,*y*), dimana harga *x* dan *y* merupakan koordinat spasial dan harga fungsi tersebut pada setiap titik (*x*,*y*) adalah tingkat kecemerlangan citra pada titik tersebut.

Citra digital dinyatakan dengan matriks berukuran $N \times M$

dimana :

N = Jumlah baris / tinggi, $0 \leq y \leq N-1$

M = Jumlah kolom/lebar, $0 \leq x \leq M-1$

Represansi citra dalam bentuk matriks seperti di bawah ini :

$$f(x,y) \approx \begin{bmatrix} f(0,0) & \cdots & f(0,M-1) \\ f(1,0) & \ldots & f(1,M-1) \\ \vdots & \vdots & \vdots \\ f(N-1,0) & \cdots & f(N-1,M-1) \end{bmatrix}$$

Setiap elemen pada citra digital disebut dengan pixel. Citra berukuran N x M mempunyai NM buah pixel (Dulimarta, 1997).

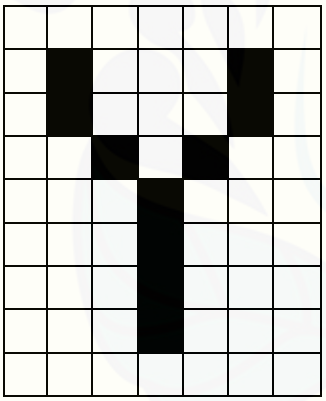
2.2.1 Macam Citra Berdasarkan Penyimpanan Nilai *Pixel*

Menurut nilai pixelnya citra terbagi tiga jenis yaitu :

a. Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai *pixel* yaitu hitam dan putih. Citra biner juga disebut sebagai citra B&W *(black and white*) atau monokrom. Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengambangan, morphologi, ataupun dithering. Setiap titik (*pixel*) dalam citra bernilai 0 atau 1. Warna hitam sinyatakan dengan 0, sedangkan warna putih dinyatakan dengan 1.

Setiap titik membutuhkan media penyimpanan 1-bit, contoh citra biner terlihat pada Gambar 2.2



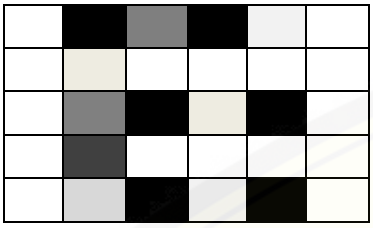
1	1	1	1	1	1	1
1	0	1	1	1	0	1
1	0	1	1	1	0	1
1	1	0	1	0	1	1
1	1	1	0	1	1	1
1	1	1	0	1	1	1
1	1	1	0	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	1	1

Gambar 2.2 Citra Biner (Hitam=0, Putih=1)

b. Citra Keabuan (*Grayscale*)

Citra grayscale merupakan suatu cara dalam merepresentasikan citra digital dengan menggunakan skala derajat keabuan. Citra skala keabuan mempunyai kemungkinan warna antara hitam (minimal) dan putih (maksimal). Jumlah maksimum warna sesuai dengan bit penyimpanan yang digunakan. Derajat keabuan yang ada merupakan hasil pemangkatan nilai bit yang ada terhadap angka 2.

Contoh : skala keabuan 4-bit maka jumlah kemungkinan total warnanya yaitu 2^4=16 dengan 0 warna minimal sampai 15 warna maksimal. Untuk lebih jelas contoh dari citra keabuan terlihat pada Gambar 2.3



15	0	6	0	13	15
15	12	15	15	15	15
15	5	0	12	0	15
15	8	15	15	15	15
15	10	0	13	0	15

Gambar 2.3 Citra Keabuan 4-bit

c. Citra Warna

Setiap titik (*pixel*) pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar yaitu merah hijau biru yaitu citra RGB (*Red Green Blue*). Setiap warna dasar mempunyai intensitas sendiri dengan nilai 0-255 atau 8-bit.

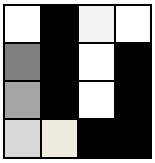
Red = Memiliki warna minimal putih dan warna maksimal merah

Green = Memiliki warna minimal putih dan warna maksimal hijau

Blue = Memiliki warna minimal putih dan warna maksimal biru

Misal warna kuning adalah kombinasi warna merah dan hijau sehingga memiliki nilai RGB 255 255 0, warna ungu muda adalah kombinasi warna merah dan biru sehingga memiliki nilai RGB 150 0 150.

Setiap titik (*pixel*) pada citra warna membutuhkan data 3 byte yaitu RGB (*Red Green Blue*). Data 3 byte memiliki 24-bit sehingga jumlah kemungkinan kombinasi warnanya yaitu 2^{24} > 16 juta warna. Ada perbedaan warna dasar untuk cahaya (misal *display* di monitor komputer) dan untuk cat/tinta (misal cetakan di atas kertas). Citra cahaya menggunakan warna dasar RGB = *Red Green Blue* sedangkan citra cat menggunakan warna dasar CMY = *Cyan Magenta Yellow*. Contoh gambar citra warna seperti terlihat pada Gambar 2.4.



255	255	255	0	0	0	128	128	128	128	128	0
0	255	255	0	0	0	204	255	255	0	0	255
150	150	150	51	51	51	255	255	255	95	95	95
255	204	153	255	204	153	128	0	0	255	0	255

Gambar 2.4 Citra warna

Gambar 2.5 merupakan contoh citra biner, citra keabuan dan citra warna

(a) (b) (c)

Gambar 2.5 (a) Contoh Citra Biner (b) Citra Keabuan (c) Citra Warna (Sumber : Su'a, 2015)

2.3 *Data Encryption Standard* (*DES*)

Algoritma DES dikembangkan di IBM di bawah kepemimpinan Tuchman pada tahun 1972. Algoritma ini didasarkan pada algoritma Lucifer yang dibuat oleh Horst Feistel. Algoritma *Data Encryption Standard* (*DES*) telah disetujui oleh *National Bureau of Standard* (*NBS*) setelah penilaian kekuatannya oleh *National Security Agency* (*NSA*) Amerika Serikat. DES termasuk ke dalam sistem kriptografi simetri dan tergolong jenis cipher blok, beroperasi pada ukuran blok 64-bit. DES mengenkripsikan 64-bit plainteks menjadi 64-bit cipherteks dengan menggunakan 56-bit kunci internal (*internal key*) atau *subkey*. Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64-bit (Munir, 2004).

2.3.1 Skema Global DES (*Data Encryption Standard*)

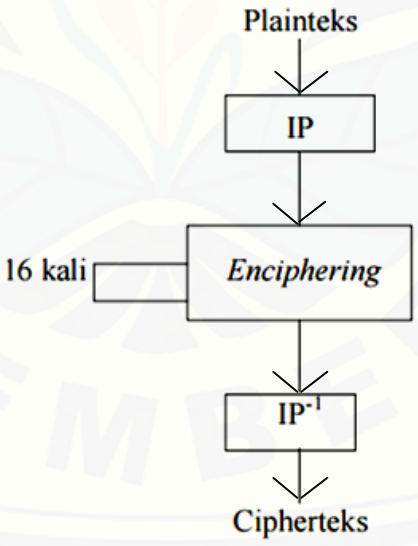
a. Blok plainteks dipermutasi dengan matriks permutasi awal (initial permutation atau IP), terlihat pada Tabel 2.1 tabel untuk initial permutation atau IP.

Tabel 2.1 Tabel Initial Permutation (IP)

58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

b. Hasil permutasi awal kemudian di-enciphering sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.

c. Hasil enciphering kemudian dipermutasi dengan matriks permutasi balikan (invers initial permutation atau IP^{-1}) menjadi blok cipherteks. Seperti terlihat pada Gambar 2.6.



Gambar 2.6 Skema Global DES (*Data Encryption Standard*)

Proses enkripsi menggunakan kunci internal yang dipermutasi menggunakan tabel PC-1 yang pada proses ini terjadi pembuangan 1-bit pain akhir pada masing-masing blok kunci dari 64-bit menjadi 56-bit.

Setelah didapatkan 56-bit maka akan dibagi menjadi dua bagian yaitu C_0 dan D_0. C_0 dan D_0 ini akan mengalami pergeseran kiri atau *leftshift* sesuai tabel yang telah ada dalam algoritma. Setiap hasil putaran digabungkan lagi menjadi C_iD_i yang selanjutnya dimasukkan ke dalam permutasi kompresi 2 atau *KPC-2*. Hasil dari permutasi dan kompresi tersebut adalah kunci eksternal yang akan digunakan untuk ekspansi citra sebanyak 16 kali putaran, sehingga akan diperoleh 16 kunci yaitu K_1, K_2, …, K_{15}, K_{16}.

Di dalam proses enciphering, blok plainteks terbagi menjadi dua bagian, kiri (*L*) dan kanan (*R*), yang masing-masing panjangnya 32-bit, R yang memiliki panjang 32-bit ini dirubah menjadi 48-bit menggunakan tabel ekspansi. Kedua bagian ini masuk ke dalam 16 putaran DES. Pada setiap putaran *i*, blok *R* merupakan masukan untuk fungsi transformasi yang disebut *f*. Pada fungsi *f*, blok *R* dikombinasikan dengan kunci internal K_i akan menghasilkan A_i dimana A_i akan dimasukkan ke dalam S-Box dan menghasilkan output B_i, sedangkan B_i ini akan dimasukkan ke dalam P-Box. Keluaran dari P-Box ialah fungsi f yang di-XOR-kan dengan blok *L* untuk mendapatkan blok *R* yang baru. Sedangkan blok *L* yang baru langsung diambil dari blok *R* sebelumnya. Ini adalah satu putaran DES, untuk iterasi selanjutnya akan seperti itu sampai 16 kali putaran.

Secara matematis, satu putaran DES dinyatakan sebagai

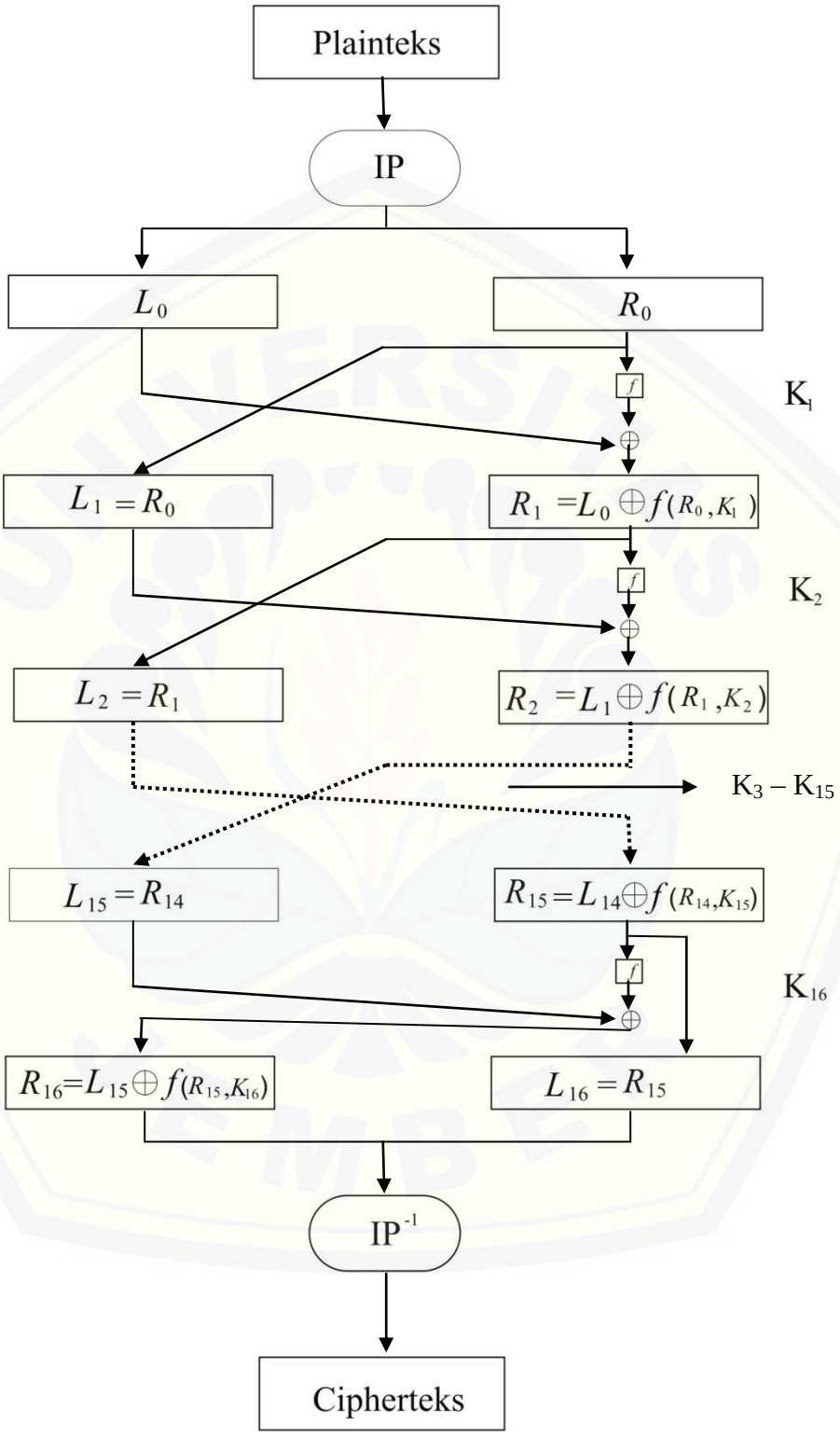
$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Keterangan :

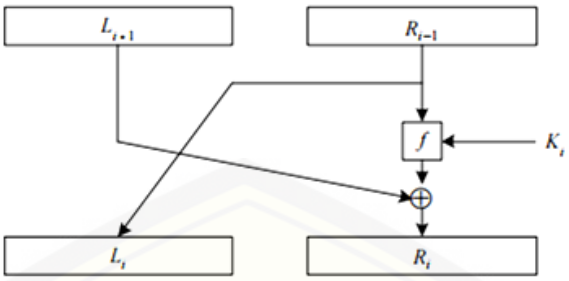
$\oplus$ = XOR

Adapun algoritma enkripsi DES pada Gambar 2.7.



Gambar 2.7 Algoritma Enkripsi dengan DES

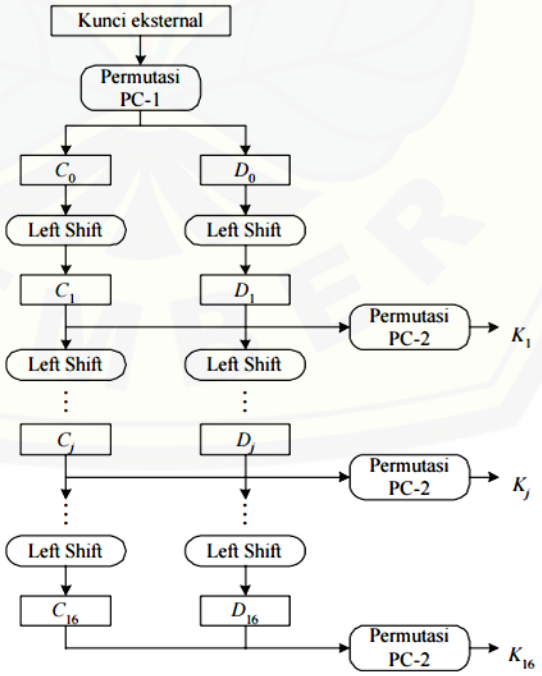
Satu putaran DES merupakan model jaringan *Feistel* (lihat Gambar 2.8).



Gambar 2.8 Jaringan *Feistel* untuk satu putaran DES

2.3.2 Pembangkitan Kunci Internal

DES memerlukan 16 putaran untuk enkripsi, maka dibutuhkan kunci internal sebanyak 16 buah, yaitu diinisialkan sebagai K_1, K_2, …, K_{15}, K_{16}. Kunci-kunci internal ini dapat dibangkitkan sebelum proses enkripsi atau bersamaan dengan proses enkripsi. Kunci internal dibangkitkan dari kunci eksternal yang diberikan oleh pengguna. Kunci eksternal panjangnya 64-bit atau 8 karakter. Terlihat pada Gambar 2.9 pembangkitan kunci internal.



Gambar 2.9 Proses pembangkitan kunci-kunci internal DES

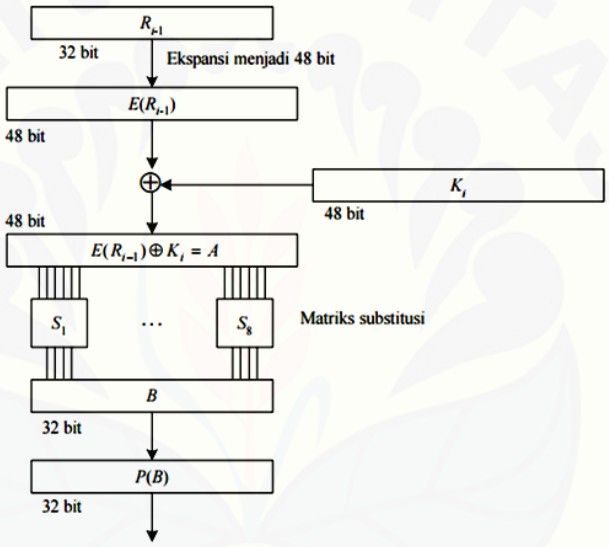
2.3.3 Enkripsi dengan Algoritma DES

Proses enciphering terhadap blok plainteks dilakukan setelah permutasi awal. Setiap blok plainteks mengalami 16 kali putaran, setiap putaran *enciphering* merupakan jaringan *Feistel* yang secara matematis dinyatakan sebagai berikut :

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

Diagram komputasi fungsi *f* diperlihatkan pada Gambar 2.10



Gambar 2.10 Komputasi Fungsi *f*

E adalah fungsi ekspansi yang memperluas blok R_{i-1} yang panjangnya 32-bit menjadi blok 48-bit. Fungsi ekspansi direalisasikan dengan matriks permutasi ekspansi dari kunci. Selanjutnya, hasil ekpansi, yaitu $E(R_{i-1})$, yang panjangnya 48 bit di-XOR-kan dengan K_i yang panjangnya 48-bit menghasilkan matriks *A* yang panjangnya 48-bit:

$$E(R_{i-1}) \oplus K_i = A$$

Matriks *A* dikelompokkan menjadi 8 kelompok, masing-masing 6-bit, dan menjadi masukan bagi proses substitusi. Proses substitusi dilakukan dengan menggunakan delapan buah kotak-S (S-box), S1 sampai S8. Setiap kotak-S

menerima masukan 6-bit dan menghasilkan keluaran 4-bit. Kelompok 6-bit pertama menggunakan S1, kelompok 6-bit kedua menggunakan S2, dan seterusnya.

Tabel 2.2 S-box algoritma DES

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	16
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Keluaran proses substitusi adalah matriks *B* yang panjangnya 48-bit. Matriks *B* menjadi masukan untuk proses permutasi. Tujuan permutasi adalah untuk mengacak hasil proses substitusi kotak-S. Permutasi dilakukan dengan menggunakan matriks permutasi P (P-box) yaitu Tabel 2.3.

Tabel 2.3 Permutasi P (P-box)

16	7	20	21	29	12	28	17	1	15	23	26	5	8	31	10
2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

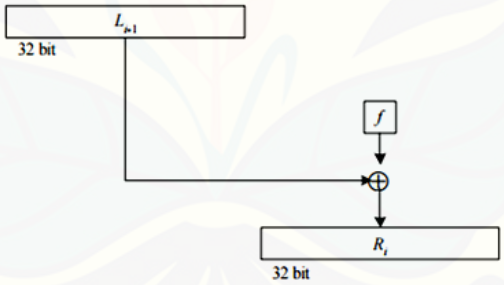
P(B) merupakan keluaran dari fungsi f. Bit-bit $P(B)$ di-XOR-kan dengan L_{i-1} menghasilkan

$$R_i = L_{i-1} \oplus P(B)$$

Jadi, keluaran dari putaran ke-i adalah seperti akan menghasilkan

$$(L_i, R_i) = (R_{i-1}, L_{i-1} \oplus P(B))$$

Skema perolehan R terlihat pada Gambar 2.11.



Gambar 2.11 Skema Perolehan R

Permutasi terakhir dilakukan setelah 16 kali putaran terhadap gabungan blok kiri dan blok kanan. Proses permutasi menggunakan matriks permutasi awal balikan (inverse initial permutation atau IP^{-1}) terlihat pada Tabel 2.4.

Tabel 2.4 Inverse Initial Permutation atau IP^{-1}

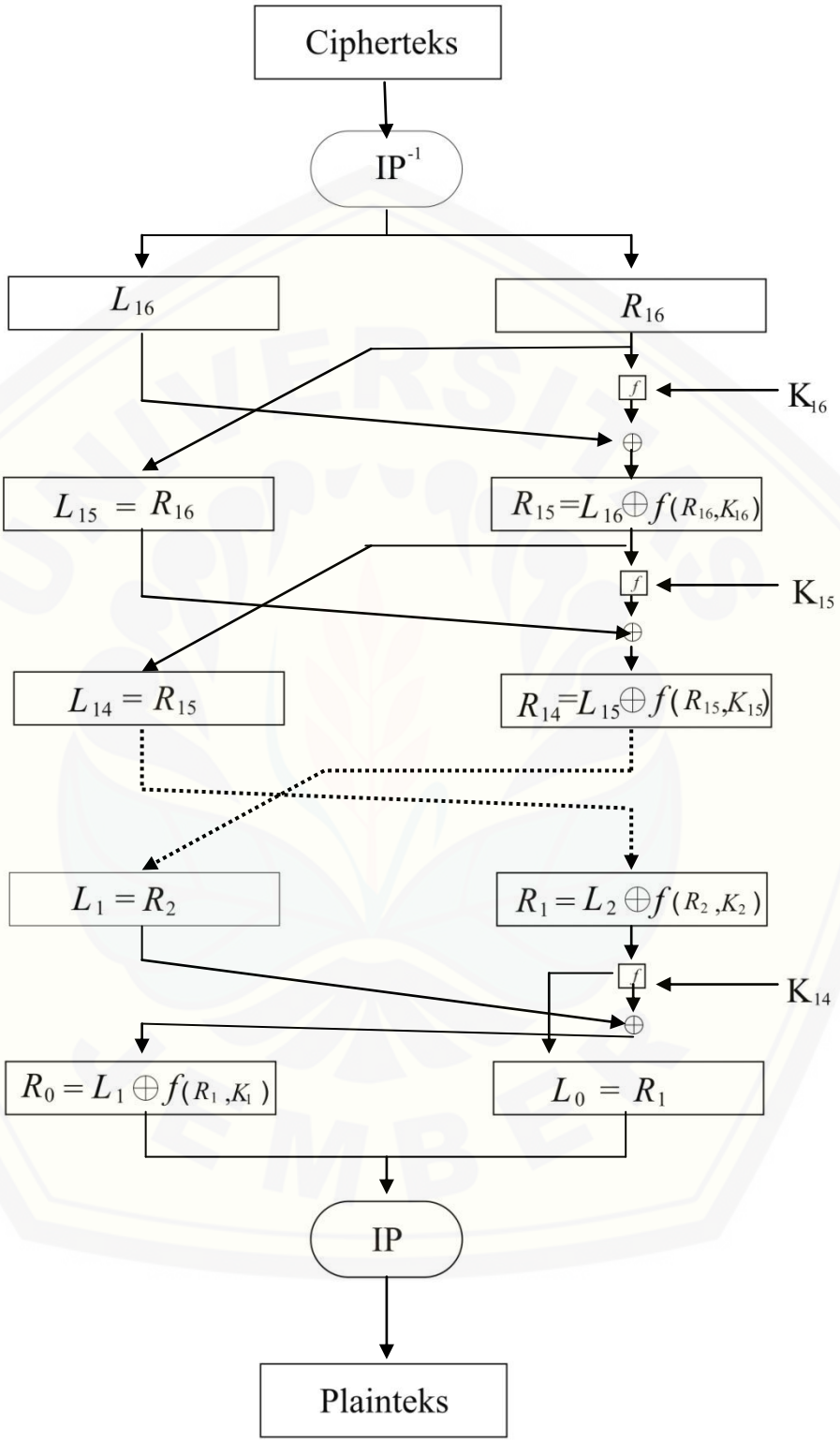
40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

2.3.4 Dekripsi

Proses dekripsi terhadap cipherteks merupakan kebalikan dari proses enkripsi. DES menggunakan algoritma yang sama untuk proses enkripsi dan dekripsi. Jika pada proses enkripsi urutan kunci internal yang digunakan adalah K_1, K_2, …, K_{15}, K_{16} maka pada proses dekripsi urutan kunci yang digunakan adalah K_{16}, K_{15}, …, K_2, K_1. Untuk tiap putaran 16, 15, …, 2, 1 keluaran pada setiap putaran *deciphering* adalah

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

yang dalam hal ini, (R_{16},L_{16}) adalah blok masukan awal untuk deciphering. Blok (R_{16},L_{16}) diperoleh dengan mempermutasikan cipherteks dengan matriks permutasi IP^{-1}. Prakeluaran dari *deciphering* adalah (L_0, R_0). Dengan permutasi awal IP akan didapatkan kembali blok plainteks semula. Tinjau kembali proses pembangkitan kunci internal. Selama *deciphering*, K_{16} dihasilkan dari (C_{16}, D_{16}) dengan permutasi PC-2. Tentu saja ($C16$, $D16$) tidak dapat diperoleh langsung pada permulaan *deciphering*. Tetapi karena (C_{16}, D_{16}) = (C_0, D_0), maka K_{16} dapat dihasilkan dari (C_0, D_0) tanpa perlu lagi melakukan pergeseran bit. Catatlah bahwa (C_0, D_0) yang merupakan bit-bit dari kunci eksternal K yang diberikan pengguna pada waktu dekripsi. Selanjutnya, K_{15} dihasilkan dari (C_{15}, D_{15}) yang mana (C_{15}, D_{15}) diperoleh dengan menggeser C_{16} (yang sama dengan C_0) dan D_{16} (yang sama dengan C_0) satu bit ke kanan. Sisanya, K_{14} sampai K_1 dihasilkan dari (C_{14}, D_{14}) sampai (C_1, D_1). Catatlah bahwa (C_{i-1},D_{i-1}) diperoleh dengan menggeser Ci dan Di dengan cara yang sama seperti enkripsi, tetapi pergeseran kiri (*left shift*) diganti menjadi pergeseran kanan (*right shift*). Adapun skema proses dekripsi pada Gambar 2.12.



Gambar 2.12 Skema Proses Dekripsi

2.4 *Bernoulli Map*

Bernoulli Map merupakan salah satu fungsi *chaos* yang digunakan dalam aplikasi kriptografi (Ahmed dan El-aziem, 2014). Fungsinya dinyatakan seperti pada persamaan (2.1).

$$X_{n+1} = r \times X_n \; mod \; 1 \tag{2.1}$$

dimana :

X_n : Mengambil nilai dari rentang 0 sampai 1, $X_n \in (0,1)$, namun untuk nilai awal X_1= 0,1.

r : Variabel r mengambil nilai dari 0 sampai ∞ , $r \in (0,\infty)$.

mod 1 : Modulus 1 yang berarti menghilangkan bilangan bulat agar menghasilkan bilangan desimal yang memiliki rentang 0 sampai 1 (Elaydi, 2007).

Algoritma ini membangkitkan barisan bilangan real, agar barisan bilangan ini dapat digunakan sebagai *keystream*. Algoritma *Bernoulli Map* yang telah dirumuskan tersebut untuk mencari pembangkit keystream yang digunakan untuk melangkah ke algoritma selanjutnya dengan membatasi bilangan yang dibangkitkan yaitu 8 angka. Algoritma ini membangkitkan deret bilangan real, agar deret bilangan ini dapat digunakan sebagai *keystream*, maka bilangan-bilangan tersebut harus dikonversikan menjadi bilangan bulat dengan rentang 0 sampai 255. Proses tersebut dilakukan dengan cara mengabsolutkan barisan kunci yang dihasilkan dari Persamaan (2.1), lalu masing- masing dari kunci tersebut dikalikan dengan 10000 dan dibulatkan ke bawah (*floor*) untuk menghasilkan bilangan bulat, setelah didapatkan barisan bilangan bulat, kunci tersebut dipetakan pada rentang antara 0 sampai 255.

$$\begin{aligned} E_n &= X_n \times 10000 \\ F_n &= \lfloor E_n \rfloor \\ K_n &= F_n \; mod \; 256 \end{aligned} \tag{2.2}$$

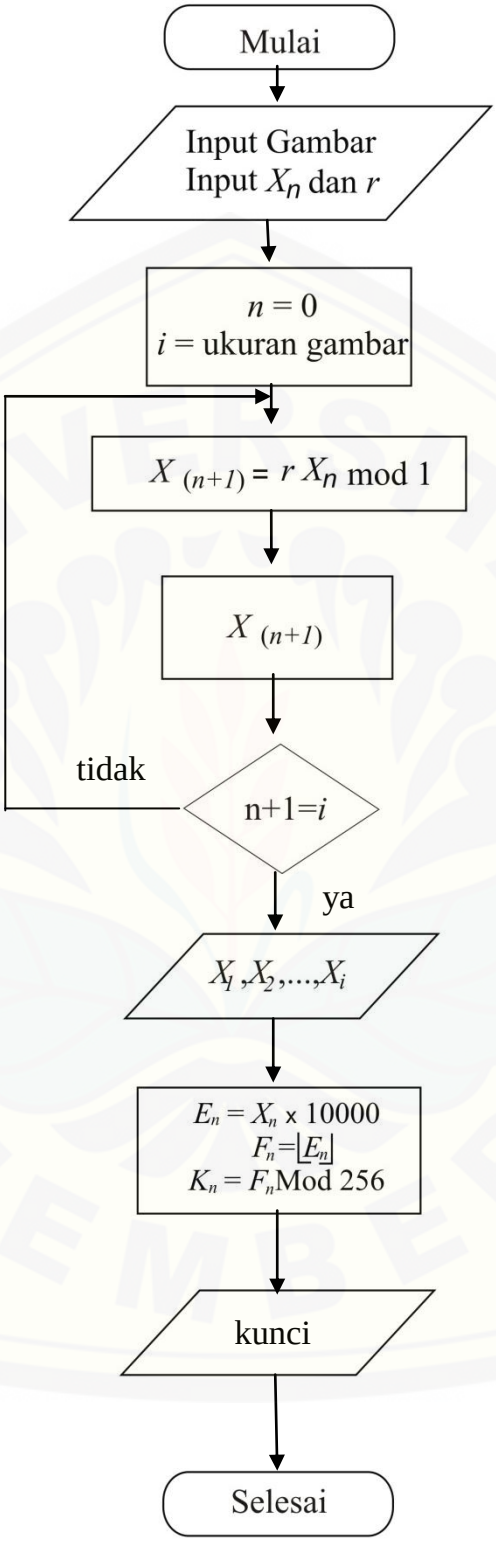
dimana :

E_n = Hasil absolut

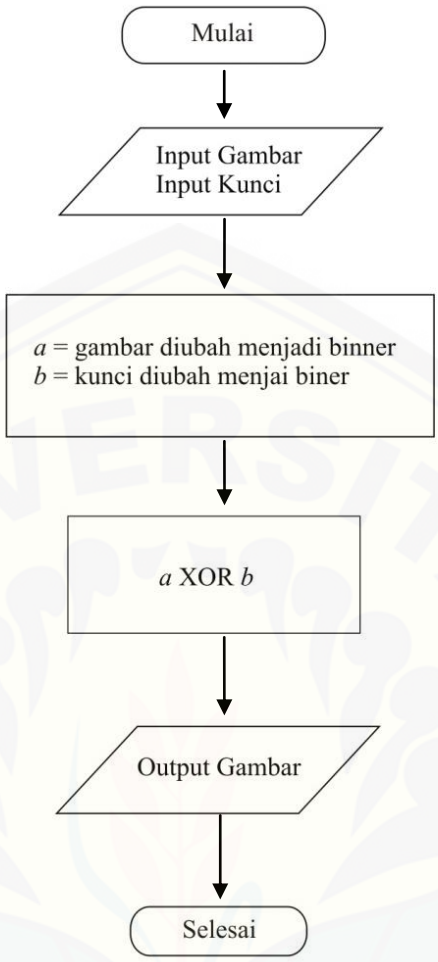
F_n = Hasil pembulatan ke bawah dari E_n

K_n = Pemetaan hasil kunci pada rentan 0 – 255.

Adapun flowchart untuk pembangkit keystream dan enkripsi *Bernoulli Map* terlihat pada Gambar 2.13 dan 2.14.



Gambar 2.13 *Flowchart* Pembangkit *Keystream* Algoritma *Bernoulli Map*



Gambar 2.14 *Flowchart* Enkripsi Citra Algoritma *Bernoulli Map*

2.5 Analisis Keamanan

2.5.1 Analisis Diferensial

Analisis diferensial dengan menentukan perbedaan dari citra sebelum dan sesudah dilakukan enkripsi yaitu dengan menghitung nilai dari *Number of Pixels Change Rate* (*NPCR*). Berikut persamaan yang digunakan :

$$NPCR = \frac{\sum_{i,j} D_{(i,j)}}{W \times H} \times 100\% \quad (2.3)$$

dimana W dan H adalah lebar dan tinggi dan $D(i,j)$ adalah nilai biner baris ke-i dan kolom ke-j dari citra.

Jika nilai yang diperoleh dari perhitungan NPCR melebihi 50% untuk cinta biner maka dikatakan enkripsi tersebut baik dan mengindikasikan terjadinya

perubahan pixel gambar cukup merata. Nilai NPCR tinggi dapat diartikan bahwa algoritma tersebut memiliki resistensi yang tinggi terhadap *differensial attack* dan nilai NPCR ideal adalah pada keadaan minimal 50% atau berada diatas 50% (Ahmed, 2007) dan (Wu, 2011).

2.5.2 Analisis Ruang Kunci

Salah satu metode analisis ruang kunci yaitu *Brute force Attack*. *Brute force Attack* adalah metode untuk meretas password (*password cracking*) dengan cara mencoba semua kemungkinan kombinasi yang ada pada "wordlist". Metode ini dijamin akan berhasil menemukan password yang ingin diretas. Namun, proses untuk meretas password dengan menggunakan metode ini akan memakan banyak waktu. *Brute force Attack* bergantung pada panjang dan kerumitan kunci yang digunakan. *Attacker* akan mencoba semua kemungkinan kunci dengan menganggap bahwa algoritma yang digunakan telah diketahui, sehingga a*ttacker* hanya berfokus pada pembentukan kunci dari algoritma tersebut. Misal untuk algoritma DES memiliki panjang kunci 56-bit maka jumlah kombinasi kunci yang dilakukan sebanyak 2^{56}=72057594037927936 kali percobaan.

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang digunakan dalam penelitian ini adalah citra biner berformat jpg.

3.2 Langkah-langkah Penelitian

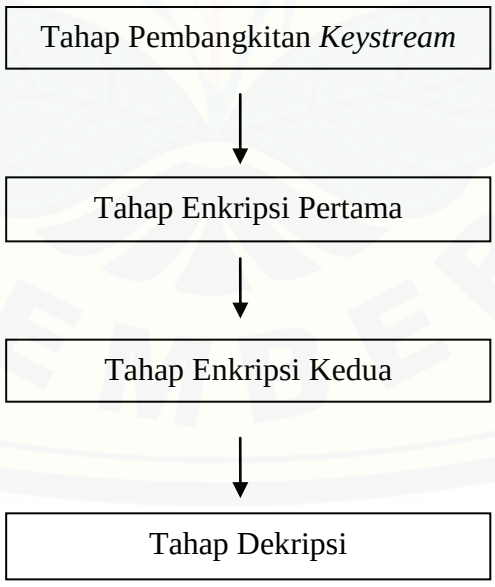
Langkah-langkah penelitian yang dilakukan sebagai berikut :

a. Studi Literatur

Studi literatur dilakukan dengan mempelajari teori dan konsep yang berkaitan dengan penelitian. Teori yang dipelajari yaitu teknik implementasi pembangkit kunci dengan algoritma *Bernoulli Map*, serta algoritma *Data Encryption Standard* (*DES*) untuk mengenkripsi citra selanjutnya hasil dari enkripsi tersebut dipelajari enkripsi citra hasil enkripsi menggunakan algoritma *Bernoulli Map*. Literatur pendukung ini berupa jurnal, artikel, buku, dan sumber lainnya.

b. Analisis Data

Proses analisis data terdiri dari beberapa tahapan terlihat pada Gambar 3.1



Gambar 3.1 Skema Analisis Data

1) Tahap Pembangkitan Kunci

Tahap pembangkitan kunci menggunakan algoritma *Bernoulli Map*. Pengguna memberikan nilai X_n dan r untuk membangkitan 8 kunci, diambil 64-bit kunci pertama sisanya disimpan untuk tahap enkripsi selanjutnya yaitu dengan menggunakan persamaan (2.1) dan (2.2)

2) Tahap Enkripsi

Tahap enkripsi pertama merupakan tahap citra semula atau *plain image* diubah menjadi citra terenkripsi atau *chiper image* dengan menggunakan algoritma DES dan menggunakan *keystream* yang telah dibangkitkan. Tahap enkripsi kedua yaitu hasil citra enkripsi dari algoritma DES dienkripsi menggunakan algoritma *Bernoulli Map*.

Adapun tahapannya sebagai berikut :

a) Citra dirubah menjadi biner dengan menentukan derajat keabuan setiap *pixel*.
b) Enkripsi citra menggunakan algoritma DES
c) Hasil citra enkripsi disimpan
d) Membangkitkan kembali kunci awal yang diberikan sesuai algoritma *Bernoulli Map* dengan iterasi ke-*n* (*n* : size ukuran citra hasil enkripsi)
e) Citra hasil enkripsi dirubah menjadi biner
f) Enkripsi citra menggunakan algoritma *Bernoulli Map*

3) Tahap Dekripsi

Proses dekripsi merupakan sistem untuk mengolah *cipherimage* menjadi *plainimage*. Proses dekripsi pada dasarnya sama dengan proses enkripsinya.

a) Masukkan Xn dan r dengan algoritma Bernoulli Map
b) Citra diubah menjadi biner
c) XOR-kan citra yang telah menjadi biner dengan *keystream*
d) Dekripsi menggunakan algoritma DES

c. Tahap Perancangan Program

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

d. Tahap Pembuatan Program

Pembuatan program dilakukan berdasarkan konsep memasukkan dua kunci rahasia dan membangkitkan kunci tersebut dengan algoritma *Bernoulli Map*, selanjutnya proses enkripsi dan dekripsi menggunakan algoritma *Data Encryption Standard* (DES) dan *Bernoulli Map*.

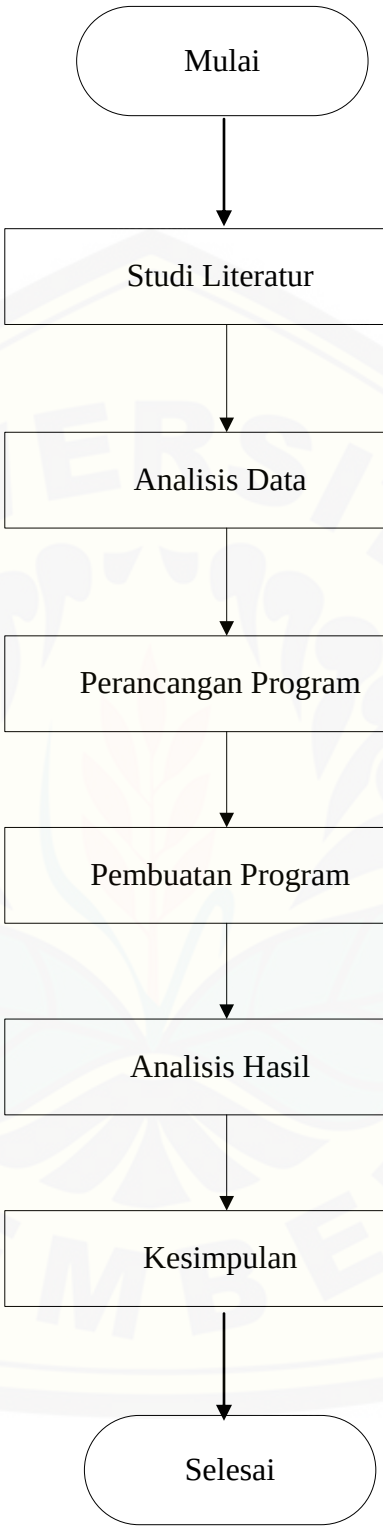
e. Analisis Hasil

Tahap analisis hasil serta pengujian hasil dibuat untuk menganalisa tingkat keamanan dengan pembangkitan kunci, serta sejauh mana tingkat perbedaan hasil enkripsi jika menggunakan algoritma DES dan *Bernoulli Map*.

f. Kesimpulan

Mengambil kesimpulan dari penelitian yang dilakukan, yaitu menganalisis proses enkripsi dan dekripsi dalam mengubah *plain image* menjadi *cipher image* dan sebaliknya menggunakan metode yang diajukan.

Tahap-tahapan di atas dapat terlihat pada Gambar 3.2.



Gambar 3.2 *Flowchart* Penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

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

a. Proses pembangkitan kunci menggunakan algoritma *Bernoulli Map* menambah kesulitan hacker untuk melakukan pengungkapan citra, hacker akan mencoba berbagi kunci namun sedikit kemungkinan bahwa hacker akan mengetahui kalau pembangkit kunci yang digunakan adalah algoritma *Bernoulli Map*
b. Proses enkripsi dilakukan dua tahap yaitu pertama enkripsi dengan menggunakan algoritma DES dan yang keddua menggunakan algoritma *Bernoulli Map*. Pengenkripsian dua tahap ini menghasilkan output yang cukup baik dimana saat pengenkripian pertama, citra hasil enkripsi memiliki tingat perbedaan 31,941% dan meningkat saat pengenkripsian kedua yaitu 63,746%.
c. Proses dekripsi merupakan kebalikan proses enkripsi dimana tetap menggunakan pembangkit kunci dari algoritma *Bernoulli Map*, namun proses algoritmanya dibalik. Hasil dekripsi dikatakan cukup bagus karna hasil yang diperoleh mirip dengan citra asli.
d. Berdasarkan analisis keamanan, algoritma yang diajukan dalam proses enkripsi aman untuk proses perlindungan citra karena memiliki kemungkinan $2^{56}+2^{512}$ kali percobaan dan kemungkinan kecil hacker mengetahui algoritma pembangkit kunci yang digunakan.

5.2 Saran

Saran yang diberikan untuk penelitian selanjutnya yaitu dapat menerapkan enkripsi ini ke dalam citra grayscale dan citra RGB, serta menerapkan ke citra format lain seperti bitmap,png,tiff dan lainnya. Sehingga penelitian akan berlanjut ke tingkat yang lebih baik.

DAFTAR PUSTAKA

Ahmed, H. 2007. Encryption Efficiency Analysis and Security Evaluation of RC6 Block Cipher for Digital Images. *Optical Engineering*. 3(1): 6–7.

Ahmed, H. E. dan El-aziem, A. H. 2014. Image Encryption Using Development of Chaotic Logistic Map Based on Feedback *Stream* Cipher. *Recent Advances In Telecommunications, Informatics And Educational Technologies*. 11: 9-10.

Ariyus, D. 2008. *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi*. Yogyakarta: CV Andi Offset.

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

Elaydi, S. B. 2007. *Discrete Chaos with Applications in Science and Engineering*. Edisi 2. Kindle Edition.

Ilie, L. 2011. *Cryptography and Security*. University Of Western Ontario Department Of Computer Science.

Kumar, P. B. dan Rajaana, N. S. 2016. Data Encryption and Decryption Using By Triple DES Performance Efficiency Analysis of Cryptosystem. *International Journal of Innovative Research in Computer and Communication Engineering*. 11: 3-6.

Laura, N. S., Sukirman, E. dan Suryadi M.T. 2017. Penerapan Algoritma Bernoulli Map dalam Program Aplikasi Enkripsi Citra Digital. *Jurnal Informatika dan Komputer*. 22(1): 1-3.

Menez, J,A. P,C,van Oorschot dan S, A, Vanstone. 1996. *Handbook of Applied Cryptography*. CRC Press.Inc.

Munir, R. 2004. *Data Encryption Standard* (*DES*). Bandung: Departemen Teknik Informatika, Institut Teknologi Bandung.

Munir, R. 2004.*Pengolahan Citra Digital dengan Pendekatan Algoritmik.* Bandung: Departemen Teknik Informatika, Institut Teknologi Bandung.

Munir, R. 2006. *Kriptografi*. Bandung: Departemen Teknik Informatika, Institut Teknologi Bandung.

Putra, D. 2010. *Pengolahan Citra Digital*. Edisi I. Yogyakarta: ANDI.

Schneiner, B. 1996. *Applied Cryptography*. John Wiley & Sons, Inc.

Singh, J. dan Alam, M. F. 2015. Encryption and Decryption of Textual Data with Symmetric Key Cryptography and Improved Des Method Based on Irrational Number. *International Journal of Research.* 7: 1-3.

Sawaluddin, S., Harahap, dan J, Hutagalung. 2006. *Buku Ajar Pengolahan Citra Digital*. Universitas Sumatera Utara, Medan.

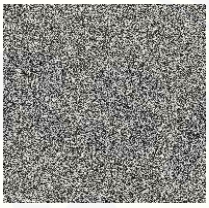
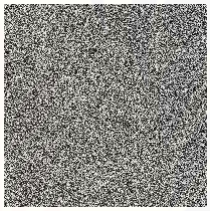
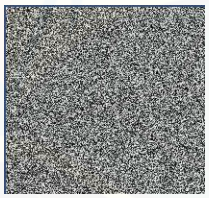
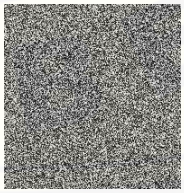
Sutoyo, T., Mulyanto, E., Suhartono, V., Nurhayati, O. D., dan Wijanarto.2009. *Teori Pengolahan Citra Digital*. Yogyakarta: CV. Andi Offset.

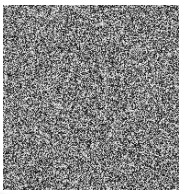
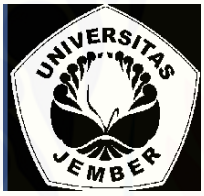
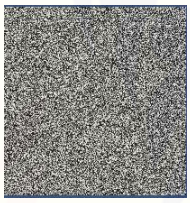
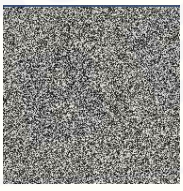
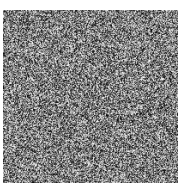
Su'a, T. 2015. *Visual Computing Computer Vision 2*. INFO410 & INFO350.

Wu, Y. 2011. NPCR and UACI Randomness Tests for Image Encryption. *Cyber Journals: Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT)* 8: 1-5.

LAMPIRAN

LAMPIRAN A. Tabel Uji dengan Beberapa Citra

No	Nama Citra	Citra Asli	Cintra Enkripsi	NPCR Enkirpsi Pertama	NPCR Enkirpsi Kedua
1	Lenna			31,941%	63,746%
2	Kucing1			32,134%	63,753%
3	Macan			32,003%	63,744%
4	Angka			29,382 %	63,745 %

5	Babon			31,950 %	63,744 %
6	Daun			32,072 %	63,756 %
7	Unej			32,250 %	63,754 %
8	Kameramen			31,716 %	63,746 %
9	Halaman			32,126 %	63,754 %
10	Kapal			31,920 %	63,760 %

LAMPIRAN B. Pembangkit Kunci Menggunakan Algoritma *Bernoulli Map*

```
function kunci=pkunci(gambar,a1,b11)
uy=imread('kucing1.jpg');
[m,n]=size(uy)
citra=m*n
an1=input('a1      = ','s');
an2=input('b11      = ','s');
a1=str2num(get(handles.edit1,'string'));
b11=str2num(get(handles.edit2,'string'));
for i=1:citra
    ki=mod(a1*b11,1)
    b11=ki
    k(i)=mod(floor(ki*1000),256)
end
```

LAMPIRAN C. Enkripsi Citra

```
PC2=[14 17 11 24 1 5 3 28 15 6 21 10 23 19 12 4 26 8 16 7 27 20 13
2 41 52 31 37 47 55 30 40 51 45 33 48 44 49 39 56 34 53 46 42 50
36 29 32];
M =[0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1 0 1 1 0 0 1 1
1 1 0 0 0 1 0 0 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1];
IP=[58 50 42 34 26 18 10 2 60 52 44 36 28 20 12 4 62 54 46 38 30
22 14 6 64 56 48 40 32 24 16 8 57 49 41 33 25 17 9 1 59 51 43 35
27 19 11 3 61 53 45 37 29 21 13 5 63 55 47 39 31 23 15 7];
E=[32 1 2 3 4 5 4 5 6 7 8 9 8 9 10 11 12 13 12 13 14 15 16 17 16
17 18 19 20 21 20 21 22 23 24 25 24 25 26 27 28 29 28 29 30 31 32
1];
S1=[14 4 13 1 2 15 11 8 3 10 6 12 5 9 0 7; 0 15 7 4 14 2 13 1 10 6
12 11 9 5 3 8; 4 1 14 8 13 6 2 11 15 12 9 7 3 10 5 0; 15 12 8 2 4
9 1 7 5 11 3 14 10 0 6 13];
S2=[15 1 8 14 6 11 3 4 9 7 2 13 12 0 5 10; 3 13 4 7 15 2 8 14 12 0
1 10 6 9 11 5; 0 14 7 11 10 4 13 1 5 8 12 6 9 3 2 15; 13 8 10 1 3
15 4 2 11 6 7 12 0 5 14 9];
S3=[10 0 9 14 6 3 15 5 1 13 12 7 11 4 2 8; 13 7 0 9 3 4 6 10 2 8 5
14 12 11 15 1; 13 6 4 9 8 15 3 0 11 1 2 12 5 10 14 7; 1 10 13 0 6
9 8 7 4 15 14 3 11 5 2 12];
S4=[7 13 14 3 0 6 9 10 1 2 8 5 11 12 4 15; 13 8 11 5 6 15 0 3 4 7
2 12 1 10 14 9; 10 6 9 0 12 11 7 13 15 1 3 14 5 2 8 4; 3 15 0 6 10
1 13 8 9 4 5 11 12 7 2 14];
S5=[2 12 4 1 7 10 11 6 8 5 3 15 13 0 14 9; 14 11 2 12 4 7 13 1 5 0
15 10 3 9 8 6; 4 2 1 11 10 13 7 8 15 9 12 5 6 3 0 14; 11 8 12 7 1
14 2 13 6 15 0 9 10 4 5 3];
S6=[12 1 10 15 9 2 6 8 0 13 3 4 14 7 5 11; 10 15 4 2 7 12 9 5 6 1
13 14 0 11 3 8; 9 14 15 5 2 8 12 3 7 0 4 10 1 13 11 6; 4 3 2 12 9
5 15 10 11 14 1 7 6 0 8 13];
S7=[4 11 2 14 15 0 8 13 3 12 9 7 5 10 6 1; 13 0 11 7 4 9 1 10 14 3
5 12 2 15 8 6; 1 4 11 13 12 3 7 14 10 15 6 8 0 5 9 2; 6 11 13 8 1
4 10 7 9 5 0 15 14 2 3 12];
S8=[13 2 8 4 6 15 11 1 10 9 3 14 5 0 12 7; 1 15 13 8 10 3 7 4 12 5
6 11 0 14 9 2; 7 11 4 1 9 12 14 2 0 6 10 13 15 3 5 8; 2 1 14 7 4
10 8 13 15 12 9 0 3 5 6 11];
P=[16 7 20 21 29 12 28 17 1 15 23 26 5 18 31 10 2 8 24 14 32 27 3
9 19 13 30 6 22 11 4 25];
IPinv=[40 8 48 16 56 24 64 32 39 7 47 15 55 23 63 31 38 6 46 14 54
22 62 30 37 5 45 13 53 21 61 29 36 4 44 12 52 20 60 28 35 3 43 11
51 19 59 27 34 2 42 10 50 18 58 26 33 1 41 9 49 17 57 25];

kpc1=k(PC1);
Co=kpc1(1:28);
Do=kpc1(29:56);
kpc2=zeros(16,48);
Lo=zeros(16,32);
Ro=zeros(16,32);
uy=getimage(handles.axes1);
for i=1:16
    if(i==1 || i==2 || i==9 || i==16)
        Co=Co([2:end 1]);
        Do=Do([2:end 1]);
    else
        Co=Co([3:end 1 2]);
```

```
        Do=Do([3:end 1 2]);
    end
    BFK=horzcat(Co,Do);
    kpc2(i,:)=BFK(PC2);
end
img=reshape(uy,[4096 64]);
for m=1:4096
M=img(m,:);
IP1=M(IP);
Lo=IP1(1:32);
Ro=IP1(33:64);

for i=1:16
     ERo=Ro(E);
     g=xor(ERo,kpc2(i,:));
     B1=g(1:6);
     B2=g(7:12);
     B3=g(13:18);
     B4=g(19:24);
     B5=g(25:30);
     B6=g(31:36);
     B7=g(37:42);
     B8=g(43:48);

    if(B1(1)==0 && B1(6)==0)
         row=1;
        elseif (B1(1)==0 && B1(6)==1)
        row=2;
        elseif(B1(1)==1 && B1(6)==0)
        row=3;
        elseif(B1(1)==1 && B1(6)==1)
        row=4;
    end
        c=B1(2:5);
        col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
        S1B1=S1(row,col);
        S1B1=de2bi(S1B1,4);
        S1_B1(1)=S1B1(4);
        S1_B1(2)=S1B1(3);
        S1_B1(3)=S1B1(2);
        S1_B1(4)=S1B1(1);

       if(B2(1)==0 && B2(6)==0)
            row=1;
        elseif (B2(1)==0 && B2(6)==1)
            row=2;
        elseif(B2(1)==1 && B2(6)==0)
            row=3;
        elseif(B2(1)==1 && B2(6)==1)
            row=4;
        end
        c=B2(2:5);
        col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
        S2B2=S2(row,col);
        S2B2=de2bi(S2B2,4);
        S2_B2(1)=S2B2(4);
```

```
S2_B2(2)=S2B2(3);
S2_B2(3)=S2B2(2);
S2_B2(4)=S2B2(1);

if(B3(1)==0 && B3(6)==0)
    row=1;
elseif (B3(1)==0 && B3(6)==1)
    row=2;
elseif(B3(1)==1 && B3(6)==0)
    row=3;
elseif(B3(1)==1 && B3(6)==1)
    row=4;
end
c=B3(2:5);
col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
S3B3=S3(row,col);
S3B3=de2bi(S3B3,4);
S3_B3(1)=S3B3(4);
S3_B3(2)=S3B3(3);
S3_B3(3)=S3B3(2);
S3_B3(4)=S3B3(1);


if(B4(1)==0 && B4(6)==0)
    row=1;
elseif (B4(1)==0 && B4(6)==1)
    row=2;
elseif(B4(1)==1 && B4(6)==0)
    row=3;
elseif(B4(1)==1 && B4(6)==1)
    row=4;
end
c=B4(2:5);
col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
S4B4=S4(row,col);
S4B4=de2bi(S4B4,4);
S4_B4(1)=S4B4(4);
S4_B4(2)=S4B4(3);
S4_B4(3)=S4B4(2);
S4_B4(4)=S4B4(1);

if(B5(1)==0 && B5(6)==0)
    row=1;
elseif (B5(1)==0 && B5(6)==1)
    row=2;
elseif(B5(1)==1 && B5(6)==0)
    row=3;
elseif(B5(1)==1 && B5(6)==1)
    row=4;
end
c=B5(2:5);
col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
S5B5=S5(row,col);
S5B5=de2bi(S5B5,4);
S5_B5(1)=S5B5(4);
S5_B5(2)=S5B5(3);
```

```
S5_B5(3)=S5B5(2);
S5_B5(4)=S5B5(1);


if(B6(1)==0 && B6(6)==0)
    row=1;
elseif (B6(1)==0 && B6(6)==1)
    row=2;
elseif(B6(1)==1 && B6(6)==0)
    row=3;
elseif(B6(1)==1 && B6(6)==1)
    row=4;
end
c=B6(2:5);
col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
S6B6=S6(row,col);
S6B6=de2bi(S6B6,4);
S6_B6(1)=S6B6(4);
S6_B6(2)=S6B6(3);
S6_B6(3)=S6B6(2);
S6_B6(4)=S6B6(1);

if(B7(1)==0 && B7(6)==0)
    row=1;
elseif (B7(1)==0 && B7(6)==1)
    row=2;
elseif(B7(1)==1 && B7(6)==0)
    row=3;
elseif(B7(1)==1 && B7(6)==1)
    row=4;
end
c=B7(2:5);
col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
S7B7=S7(row,col);
S7B7=de2bi(S7B7,4);
S7_B7(1)=S7B7(4);
S7_B7(2)=S7B7(3);
S7_B7(3)=S7B7(2);
S7_B7(4)=S7B7(1);

if(B8(1)==0 && B8(6)==0)
    row=1;
elseif (B8(1)==0 && B8(6)==1)
    row=2;
elseif(B8(1)==1 && B8(6)==0)
    row=3;
elseif(B8(1)==1 && B8(6)==1)
    row=4;
end
c=B8(2:5);
col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
S8B8=S8(row,col);
S8B8=de2bi(S8B8,4);
S8_B8(1)=S8B8(4);
S8_B8(2)=S8B8(3);
S8_B8(3)=S8B8(2);
```

```
            S8_B8(4)=S8B8(1);
            SBoxed(1:4)=S1_B1;
            SBoxed(5:8)=S2_B2;
            SBoxed(9:12)=S3_B3;
            SBoxed(13:16)=S4_B4;
            SBoxed(17:20)=S5_B5;
            SBoxed(21:24)=S6_B6;
            SBoxed(25:28)=S7_B7;
            SBoxed(29:32)=S8_B8;
            f=SBoxed(P);
            temp=xor(Lo,f);
            Lo=Ro;
            Ro=temp;
end
RoLo=horzcat(Ro,Lo);
Final=RoLo(IPinv);
imi(m,:)=Final;
end
img1=reshape(imi,[512 512]);
imwrite(img1,'Enci1.jpg','jpg');

global Img
global ay;
global key;
Img = imread('Enci1.jpg');
[n m k] = size(Img);
  key = keyGen(512*512);

EncImg = imageProcess(Img,key);
axes(handles.axes2)
imshow(EncImg);
imwrite(EncImg,'Encoded.jpg','jpg');
guidata(hObject, handles);
```

LAMPIRAN D. Deskripsi Citra

```
global Img
global ay;
global key;
Img = imread('Encoded.jpg');
[n m k] = size(Img);
key = keyGen(512*512);
DecImg = imageProcess(Img,key);
imwrite(DecImg,'Decoded.jpg','jpg');
PC1=[57 49 41 33 25 17 9 1 58 50 42 34 26 18 10 2 59 51 43 35 27
19 11 3 60 52 44 36 63 55 47 39 31 23 15 7 62 54 46 38 30 22 14 6
61 53 45 37 29 21 13 5 28 20 12 4];
PC2=[14 17 11 24 1 5 3 28 15 6 21 10 23 19 12 4 26 8 16 7 27 20 13
2 41 52 31 37 47 55 30 40 51 45 33 48 44 49 39 56 34 53 46 42 50
36 29 32];
M =[0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 1 0 1 0 0 0 1 0 1 0 1 1 0 0 1 1
1 1 0 0 0 1 0 0 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1];
IP=[58 50 42 34 26 18 10 2 60 52 44 36 28 20 12 4 62 54 46 38 30
22 14 6 64 56 48 40 32 24 16 8 57 49 41 33 25 17 9 1 59 51 43 35
27 19 11 3 61 53 45 37 29 21 13 5 63 55 47 39 31 23 15 7];
E=[32 1 2 3 4 5 4 5 6 7 8 9 8 9 10 11 12 13 12 13 14 15 16 17 16
17 18 19 20 21 20 21 22 23 24 25 24 25 26 27 28 29 28 29 30 31 32
1];
S1=[14 4 13 1 2 15 11 8 3 10 6 12 5 9 0 7; 0 15 7 4 14 2 13 1 10 6
12 11 9 5 3 8; 4 1 14 8 13 6 2 11 15 12 9 7 3 10 5 0; 15 12 8 2 4
9 1 7 5 11 3 14 10 0 6 13];
S2=[15 1 8 14 6 11 3 4 9 7 2 13 12 0 5 10; 3 13 4 7 15 2 8 14 12 0
1 10 6 9 11 5; 0 14 7 11 10 4 13 1 5 8 12 6 9 3 2 15; 13 8 10 1 3
15 4 2 11 6 7 12 0 5 14 9];
S3=[10 0 9 14 6 3 15 5 1 13 12 7 11 4 2 8; 13 7 0 9 3 4 6 10 2 8 5
14 12 11 15 1; 13 6 4 9 8 15 3 0 11 1 2 12 5 10 14 7; 1 10 13 0 6
9 8 7 4 15 14 3 11 5 2 12];
S4=[7 13 14 3 0 6 9 10 1 2 8 5 11 12 4 15; 13 8 11 5 6 15 0 3 4 7
2 12 1 10 14 9; 10 6 9 0 12 11 7 13 15 1 3 14 5 2 8 4; 3 15 0 6 10
1 13 8 9 4 5 11 12 7 2 14];
S5=[2 12 4 1 7 10 11 6 8 5 3 15 13 0 14 9; 14 11 2 12 4 7 13 1 5 0
15 10 3 9 8 6; 4 2 1 11 10 13 7 8 15 9 12 5 6 3 0 14; 11 8 12 7 1
14 2 13 6 15 0 9 10 4 5 3];
S6=[12 1 10 15 9 2 6 8 0 13 3 4 14 7 5 11; 10 15 4 2 7 12 9 5 6 1
13 14 0 11 3 8; 9 14 15 5 2 8 12 3 7 0 4 10 1 13 11 6; 4 3 2 12 9
5 15 10 11 14 1 7 6 0 8 13];
S7=[4 11 2 14 15 0 8 13 3 12 9 7 5 10 6 1; 13 0 11 7 4 9 1 10 14 3
5 12 2 15 8 6; 1 4 11 13 12 3 7 14 10 15 6 8 0 5 9 2; 6 11 13 8 1
4 10 7 9 5 0 15 14 2 3 12];
S8=[13 2 8 4 6 15 11 1 10 9 3 14 5 0 12 7; 1 15 13 8 10 3 7 4 12 5
6 11 0 14 9 2; 7 11 4 1 9 12 14 2 0 6 10 13 15 3 5 8; 2 1 14 7 4
10 8 13 15 12 9 0 3 5 6 11];
P=[16 7 20 21 29 12 28 17 1 15 23 26 5 18 31 10 2 8 24 14 32 27 3
9 19 13 30 6 22 11 4 25];
IPinv=[40 8 48 16 56 24 64 32 39 7 47 15 55 23 63 31 38 6 46 14 54
22 62 30 37 5 45 13 53 21 61 29 36 4 44 12 52 20 60 28 35 3 43 11
51 19 59 27 34 2 42 10 50 18 58 26 33 1 41 9 49 17 57 25];

kpc1=k(PC1);
Co=kpc1(1:28);
```

```
Do=kpc1(29:56);
kpc2=zeros(16,48);
Lo=zeros(16,32);
Ro=zeros(16,32);
uy=getimage(handles.axes2);
for i=1:16
    if(i==1 || i==2 || i==9 || i==16)
        Co=Co([2:end 1]);
        Do=Do([2:end 1]);
    else
        Co=Co([3:end 1 2]);
        Do=Do([3:end 1 2]);
    end
    BFK=horzcat(Co,Do);
    kpc2(i,:)=BFK(PC2);
end
img=reshape(uy,[4096 64]);
for m=1:4096
M=img(m,:);
IP1=M(IP);
Lo=IP1(1:32);
Ro=IP1(33:64);

kpc3(1,:)=kpc2(16,:);kpc3(2,:)=kpc2(15,:);kpc3(3,:)=kpc2(14,:);kpc
3(4,:)=kpc2(13,:);kpc3(5,:)=kpc2(12,:);kpc3(6,:)=kpc2(11,:);kpc3(7
,:)=kpc2(10,:);
kpc3(8,:)=kpc2(9,:);kpc3(9,:)=kpc2(8,:);kpc3(10,:)=kpc2(7,:);kpc3(
11,:)=kpc2(6,:);kpc3(12,:)=kpc2(5,:);kpc3(13,:)=kpc2(4,:);kpc3(14,
:)=kpc2(3,:);
kpc3(15,:)=kpc2(2,:);kpc3(16,:)=kpc2(1,:);
img1=reshape(img1,[4096 64]);

for m=1:4096
M=img1(m,:);
IP1=M(IP);
Lo=IP1(1:32);
Ro=IP1(33:64);

for i=1:16
     ERo=Ro(E);
     g=xor(ERo,kpc3(i,:));
     B1=g(1:6);
     B2=g(7:12);
     B3=g(13:18);
     B4=g(19:24);
     B5=g(25:30);
     B6=g(31:36);
     B7=g(37:42);
     B8=g(43:48);

    if(B1(1)==0 && B1(6)==0)
         row=1;
        elseif (B1(1)==0 && B1(6)==1)
        row=2;
        elseif(B1(1)==1 && B1(6)==0)
        row=3;
```

```
    elseif(B1(1)==1 && B1(6)==1)
    row=4;
end
    c=B1(2:5);
    col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
    S1B1=S1(row,col);
    S1B1=de2bi(S1B1,4);
    S1_B1(1)=S1B1(4);
    S1_B1(2)=S1B1(3);
    S1_B1(3)=S1B1(2);
    S1_B1(4)=S1B1(1);

   if(B2(1)==0 && B2(6)==0)
        row=1;
    elseif (B2(1)==0 && B2(6)==1)
        row=2;
    elseif(B2(1)==1 && B2(6)==0)
        row=3;
    elseif(B2(1)==1 && B2(6)==1)
        row=4;
    end
    c=B2(2:5);
    col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
    S2B2=S2(row,col);
    S2B2=de2bi(S2B2,4);
    S2_B2(1)=S2B2(4);
    S2_B2(2)=S2B2(3);
    S2_B2(3)=S2B2(2);
    S2_B2(4)=S2B2(1);

    if(B3(1)==0 && B3(6)==0)
        row=1;
    elseif (B3(1)==0 && B3(6)==1)
        row=2;
    elseif(B3(1)==1 && B3(6)==0)
        row=3;
    elseif(B3(1)==1 && B3(6)==1)
        row=4;
    end
    c=B3(2:5);
    col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
    S3B3=S3(row,col);
    S3B3=de2bi(S3B3,4);
    S3_B3(1)=S3B3(4);
    S3_B3(2)=S3B3(3);
    S3_B3(3)=S3B3(2);
    S3_B3(4)=S3B3(1);


    if(B4(1)==0 && B4(6)==0)
        row=1;
    elseif (B4(1)==0 && B4(6)==1)
        row=2;
    elseif(B4(1)==1 && B4(6)==0)
        row=3;
    elseif(B4(1)==1 && B4(6)==1)
```

```
    row=4;
end
c=B4(2:5);
col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
S4B4=S4(row,col);
S4B4=de2bi(S4B4,4);
S4_B4(1)=S4B4(4);
S4_B4(2)=S4B4(3);
S4_B4(3)=S4B4(2);
S4_B4(4)=S4B4(1);

if(B5(1)==0 && B5(6)==0)
    row=1;
elseif (B5(1)==0 && B5(6)==1)
    row=2;
elseif(B5(1)==1 && B5(6)==0)
    row=3;
elseif(B5(1)==1 && B5(6)==1)
    row=4;
end
c=B5(2:5);
col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
S5B5=S5(row,col);
S5B5=de2bi(S5B5,4);
S5_B5(1)=S5B5(4);
S5_B5(2)=S5B5(3);
S5_B5(3)=S5B5(2);
S5_B5(4)=S5B5(1);


if(B6(1)==0 && B6(6)==0)
    row=1;
elseif (B6(1)==0 && B6(6)==1)
    row=2;
elseif(B6(1)==1 && B6(6)==0)
    row=3;
elseif(B6(1)==1 && B6(6)==1)
    row=4;
end
c=B6(2:5);
col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
S6B6=S6(row,col);
S6B6=de2bi(S6B6,4);
S6_B6(1)=S6B6(4);
S6_B6(2)=S6B6(3);
S6_B6(3)=S6B6(2);
S6_B6(4)=S6B6(1);

if(B7(1)==0 && B7(6)==0)
    row=1;
elseif (B7(1)==0 && B7(6)==1)
    row=2;
elseif(B7(1)==1 && B7(6)==0)
    row=3;
elseif(B7(1)==1 && B7(6)==1)
    row=4;
```

```
        end
        c=B7(2:5);
        col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
        S7B7=S7(row,col);
        S7B7=de2bi(S7B7,4);
        S7_B7(1)=S7B7(4);
        S7_B7(2)=S7B7(3);
        S7_B7(3)=S7B7(2);
        S7_B7(4)=S7B7(1);

        if(B8(1)==0 && B8(6)==0)
            row=1;
        elseif (B8(1)==0 && B8(6)==1)
            row=2;
        elseif(B8(1)==1 && B8(6)==0)
            row=3;
        elseif(B8(1)==1 && B8(6)==1)
            row=4;
        end
        c=B8(2:5);
        col=(c(1)*2^3)+(c(2)*2^2)+(c(3)*2^1)+(c(4)*2^0)+1;
        S8B8=S8(row,col);
        S8B8=de2bi(S8B8,4);
        S8_B8(1)=S8B8(4);
        S8_B8(2)=S8B8(3);
        S8_B8(3)=S8B8(2);
        S8_B8(4)=S8B8(1);
        SBoxed(1:4)=S1_B1;
        SBoxed(5:8)=S2_B2;
        SBoxed(9:12)=S3_B3;
        SBoxed(13:16)=S4_B4;
        SBoxed(17:20)=S5_B5;
        SBoxed(21:24)=S6_B6;
        SBoxed(25:28)=S7_B7;
        SBoxed(29:32)=S8_B8;
        f=SBoxed(P);
        temp=xor(Lo,f);
        Lo=Ro;
        Ro=temp;
end
RoLo=horzcat(Ro,Lo);
Final=RoLo(IPinv);
imi(m,:)=Final;
end
img2=reshape(imi,[512 512]);
axes(handles.axes3);
imshow(img2);
```

LAMPIRAN E. NPCR Score

```
npcr_score = (sum( double( uy(:) ~= img1(:) ) ) / 4096)/100
set(handles.edit3,'string',num2str(npcr_score));
npcr_score1 = (sum(double ( img1(:) ~= EncImg(:) ) ) / 4096)/100;
set(handles.edit4,'string',num2str(npcr_score1));

%Dimana
%uy adalah gambar asli
%img1 adalah gambar enkripsi pertama (DES)
%EncImg adalah gambar enkripsi kedua (Bernoulli Map)
```