UPT Perpustakaan Universitas Jember

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

SKRIPSI

Oleh

Ahmad Husnan Fanani
NIM 131810101021

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS JEMBER

2018

UPT Perpustakaan Universitas Jember

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

SKRIPSI

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

Oleh

Ahmad Husnan Fanani
NIM 131810101021

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018

UPT Perpustakaan Universitas Jember

PERSEMBAHAN

Dengan menyebut nama Allah S.W.T yang Maha Pengasih lagi Maha Penyayang, shalawat serta salam terhadap junjungan nabi beasr kita Muhammad SAW. Saya persembahkan skripsi ini sebagai wujud rasa syukur dan terima kasih kepada:

1. Orang tua saya, Bapak Suparto dan Ibu Holifah, yang selalu memberikan segala bentuk dukungan, motivasi, dan do'anya untuk kemudahan, kelancaran, dan kesuksesan saya.
2. Kakak-kakak saya yang senantiasa memberi dukungan dan motivasi.
3. Teman-teman ATLAS dan Smokers yang sudah berjuang bersama serta saling memberi motivasi sampai akhir.
4. Almamater tercinta Jurusan Matematika FMIPA Universitas Jember
5. Serta, beberapa pihak yang tidak bisa saya sebutkan satu persatu, yang telah memberikan dukungannya.

UPT Perpustakaan Universitas Jember

MOTTO

"*Lakukan apa yang menurutmu benar, biar Tuhan yang menentukan hasilnya*" (1)

"Hidup ini seperti sepeda. Agar tetap seimbang, kau harus terus bergerak" (2)

Ahmad Husnan Fanani (1)

Albert Einstein (2)

UPT Perpustakaan Universitas Jember

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama: Ahmad Husnan Fanani

NIM : 131810101021

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Pengamanan Citra Dengan Algoritma *Diffie-Hellman* dan Algoritma *Simplified-Data Encryption Standard* (S-DES)" 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, April 2018

Yang menyatakan,

Ahmad Husnan Fanani
NIM 131810101021

UPT Perpustakaan Universitas Jember

SKRIPSI

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

Oleh

Ahmad Husnan Fanani
NIM 131810101021

Pembimbing:

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

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

UPT Perpustakaan Universitas Jember

PENGESAHAN

Skripsi berjudul "Pengamanan Citra Dengan Algoritma *Diffie-Hellman* dan Algoritma *Simplified-Data Encryption Standard* (S-DES)" 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. NIP. 197211291998021001	Abduh Riski, S.Si., M.Si. NIP. 199004062015041001
Anggota II,	Anggota III,
Kusbudiono, S.Si., M.Si. NIP. 197704302005011001	Dr. Firdaus Ubaidillah, S.Si., M.Si. NIP. 197006061998031003

Mengesahkan
Dekan,

Drs. Sujito, Ph.D.
NIP. 196102041987111001

UPT Perpustakaan Universitas Jember

RINGKASAN

Pengamanan Citra dengan Algoritma *Diffie-Hellman* dan Algoritma *Simplified-Data Encryption Standard* (S-DES); Ahmad Husnan Fanani, 131810101021, 2018; 56 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Data (citra) yang bersifat rahasia yang disimpan melalui komputer tidak menutup kemungkinan mudah untuk diretas oleh pelaku kejahatan di zaman pesatnya perkembangan teknologi ini. Oleh karena itu, diperlukan ilmu yang mampu mengamankan data rahasia tersebut yakni kriptografi. Algoritma S-DES adalah salah satu algoritma dalam kriptografi yang digunakan untuk mengamankan suatu data. Algoritma *Diffie-Hellman* merupakan algoritma pertukaran kunci yang dapat digunakan untuk membangkitkan kunci untuk algoritma S-DES. Tujuan penelitian yaitu untuk (1) mengetahui proses pengamanan data menggunakan algoritma *Diffie-Hellman* dan algoritma S-DES dan (2) menganalisis tingkat keamanan dari penggabungan dua algoritma tersebut.

Algoritma *Diffie-Hellman* merupakan algoritma pertukaran kunci antara pengirim data tersandi dengan penerima data tersandi. Algoritma *Diffie-Hellman* digunakan untuk mendapatkan kunci enkripsi dan dekripsi yang dapat disebarkan secara rahasia. Algoritma S-DES adalah algoritma untuk mengamankan suatu data dengan mengambil 8 bit blok dari setiap data yang ingin diamankan dan 10 bit blok dari kunci enkripsi maupun dekripsi. Analisis diferensial (NPCR) merupakan metode yang digunakan untuk menganalisis perbedaan nilai antara data awal dengan data yang telah disandikan. Apabila nilai NPCR-nya lebih dari 90% maka dapat dikatakan algoritma yang digunakan untuk menyandikan data aman dari serangan statistik. Analisis sensitivitas kunci adalah metode analisis yang digunakan untuk mengetahui kesensitifan suatu kunci algoritma kriptografi.

Data yang digunakan dalam penelitian adalah lima buah *plainimage* berupa citra RGB dan *grayscale*. Kunci yang digunakan yakni dua buah *public key* yang

UPT Perpustakaan Universitas Jember

berupa citra dan dua buah *private key* berupa bilangan berukuran 10 bit. Citra tersebut dienkripsi menggunakan algoritma S-DES dengan kunci yang telah dibangkitkan menggunakan algoritma *Diffie-Hellman*.

Proses pembangkitan kunci dilakukan dengan cara setiap *pixel* dari *public key* dioperasikan dengan *private key* menggunakan operasi modulo sesuai algoritma *Diffie-Hellman*. Hasil dari proses pembangkitan kunci adalah nilai K_A dan K_B dimana K_A adalah kunci untuk proses enkripsi dan K_B kunci untuk proses dekripsi. Nilai K_A dan K_B kemudian ditampilkan dalam bentuk citra yang disebut *secret key*.

Proses enkripsi S-DES menggunakan kunci yang telah dibangkitkan sebelumnya yakni *secret key*. Penggunaan *secret key* terhadap proses enkripsi dilakukan dengan cara *pixel* pada baris ke *i* dan kolom ke *j* dari *secret key* adalah kunci enkripsi untuk *pixel* pada baris ke *i* dan kolom ke *j* dari *plainimage*. Proses enkripsi menggunakan algoritma S-DES menghasilkan *cipherimage* yang tidak memuat informasi awal dari *plainimage*.

Hasil analisis diferensial (NPCR) antara *plainimage* dan *cipherimage* untuk masing-masing data penelitian yakni 99,5675%, 99,5142%, 99,9467%, 99,98%, dan 99,605%. Nilai NPCR dari seluruh data penelitian berada di atas 90% sehingga dapat dikatakan bahwa penggabungan antara algoritma *Diffie-Hellman* dan S-DES aman dari serangan statistik. Sedangkan hasil analisis sensitivitas kunci antara *cipherimage* dengan perbedaan kunci yang sedikit yakni 95,925% dan 97,805%. Hal ini membuktikan bahwa kunci yang digunakan untuk enkripsi memiliki tingkat kesensitifan yang tinggi.

UPT Perpustakaan Universitas Jember

PRAKATA

Puji syukur kehadirat Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Pengamanan Citra Dengan Algoritma *Diffie-Hellman* dan Algoritma *Simplified-Data Encryption Standard* (S-DES)". Skripsi ini disusun untuk memenuhi salah satu syarat pada program pendidikan strata satu (S1) Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Skripsi ini tidak akan terselesaikan tanpa adanya bantuan dari beberapa pihak, serta kerja keras dan keuletan dari diri pribadi. Oleh karena itu, dalam kesempatan ini penulis mengucapkan terima kasih atas bantuan dan bimbingan dalam penyusunan skripsi 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, Abduh Riski, S.Si., M.Si., selaku Dosen Pembimbing Anggota, Kusbudiono, S.Si., M.Si., selaku Dosen Penguji I, Dr. Firdaus Ubaidillah, S.Si., M.Si., selaku Dosen Penguji II yang juga telah meluangkan waktu dan pikiran dalam membimbing penulisan skripsi ini sampai terselesaikan;
4. Dosen dan Karyawan Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember yang telah menyalurkan ilmunya;
5. Teman-teman dan semua pihak yang juga 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 menyadari masih banyak kekurangan dalam penulisan skripsi ini. Namun, suatu usaha tidak akan berakhir dan berhasil jika tidak dimulai. Oleh karena itu, penulis

UPT Perpustakaan Universitas Jember

mengharapkan kritik dan sarannya demi penyempurnaan skripsi ini. Penulis berharap semoga skripsi ini bermanfaat bagi para pembaca

Jember, April 2018

Penulis

UPT Perpustakaan Universitas Jember

DAFTAR ISI

Halaman

UPT Perpustakaan Universitas Jember

UPT Perpustakaan Universitas Jember

UPT Perpustakaan Universitas Jember

DAFTAR GAMBAR

Halaman

UPT Perpustakaan Universitas Jember

DAFTAR TABEL

Halaman

UPT Perpustakaan Universitas Jember

UPT Perpustakaan Universitas Jember

DAFTAR LAMPIRAN

Halaman

UPT Perpustakaan Universitas Jember

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Data seperti citra yang disimpan melalui komputer atau ponsel tidak menutup kemungkinan adanya resiko untuk diretas, karena di era globalisasi yang canggih ini mempermudah pelaku kejahatan dalam melakukan kejahatan. Data (citra) yang bersifat rahasia bisa disalahgunakan sehingga bisa menyebabkan terjadinya tindak kejahatan. Untuk meminimalisir akan hal itu, dikembangkan ilmu pengetahuan mengenai pengamanan informasi yang ada dalam data.

Kriptografi adalah salah satu ilmu yang mempelajari pengamanan infomasi pada data, sehingga hanya pihak tertentu yang bisa mendapatkan informasi tersebut. Menurut Tamam (2010), kriptografi merupakan metode untuk mengamankan data, baik data berupa teks maupun berupa gambar. Metode kriptografi dilakukan dengan penyandian data asli, sehingga pihak lain yang tidak memilik akses terhadap data tersebut tidak dapat memperoleh informasi yang ada di dalamnya.

Simplified-Data Encryption Standard (S-DES) adalah salah satu algoritma dalam kriptografi. Algoritma S-DES merupakan penyederhanaan dari algoritma *Data Encryption Standard* (DES) dimana algoritma S-DES mengambil 8 bit blok dari plainteks dan 10 bit kunci yang digunakan untuk proses enkripsi maupun dekripsi. Algoritma S-DES memiliki keunggulan jika dibandingkan dengan algoritma DES yaitu dari segi kecepatan proses enkripsi dan dekripsi serta kesederhanaannya karena algoritma S-DES hanya menggunakan 10 bit kunci untuk proses enkripsi dan dekripsinya. Tetapi berdasarkan penelitian Kumar dan Srivastava (2014), algoritma S-DES rentan terhadap serangan *statistic* apabila digunakan pada data seperti citra. Hal ini dikarenakan ukuran kunci dan proses algoritma S-DES yang sederhana sehingga nilai *pixel* yang memiliki derajat keabuan yang sama apabila dienkripsi akan menghasilkan nilai yang sama. Oleh karena itu, untuk mengatasi kelemahan tersebut diperlukan modifikasi atau pengembangan kunci algoritma S-DES.

UPT Perpustakaan Universitas Jember

Algoritma *Diffie-Hellman* adalah algoritma pertukaran kunci yang dapat digunakan untuk membangkitkan suatu kunci enkripsi dan kunci dekripsi dalam kriptografi. Menurut Gunawan (2012), algoritma *Diffie-Hellman* adalah proses matematika yang dilakukan untuk menghasilkan kunci rahasia yang dapat disebarkan secara bebas. Dasar dari algoritma *Diffie-Hellman* adalah aljabar dan aritmatika modulus. Kelebihan dari algoritma *Diffie-Hellman* yakni tidak adanya proses pertukaran kunci enkripsi dan kunci dekripsi antara pengirim dan penerima pesan, sehingga seorang peretas akan kesulitan untuk mendapatkan kunci enkripsi maupun kunci dekripsi tersebut.

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

Pada penelitian berikutnya, Sholeh (2017) mencoba memperbaiki kelemahan dari penelitian Hardjo (2016) yang membahas enkripsi citra *grayscale* dengan algoritma S-DES dan *Stream Cipher*. Hasil yang didapat menyatakan penggabungan dua algoritma tersebut aman diterapkan pada citra *grayscale* dengan proses enkripsi yang lebih sederhana. Kunci yang digunakan untuk proses enkripsi dibangkitkan dari bilangan biner *random* 10 bit yang kemudian dienkripsi menggunakan algoritma *Stream Cipher*. Proses tersebut nantinya akan menghasilkan *key citra* yang akan dijadikan kunci untuk proses enkripsi menggunakan algoritma S-DES. Kelemahan dari penelitian tersebut terdapat pada proses dekripsi. Proses dekripsi menggunakan *key citra* dari proses enkripsi sebelumnya. Pada proses dekripsi algoritma *Stream Cipher* tidak digunakan.

UPT Perpustakaan Universitas Jember

Pada penelitian ini, penulis akan mengajukan metode baru dengan proses yang sederhana dan tingkat keamanan yang lebih baik dibandingkan penelitian-penelitian sebelumnya. Metode yang dimaksud yaitu mengenkripsi data digital yang berupa citra dengan memodifikasi algoritma S-DES dengan algoritma *Diffie-Hellman*. Modifikasi dilakukan untuk mengatasi kelemahan algoritma S-DES yang terletak pada kuncinya yang pendek dan seragam jika digunakan pada data yang kompleks seperti citra. Proses modifikasi dilakukan dengan cara membangkitkan kunci yang berupa citra menggunakan algoritma *Diffie-Hellman* yang nantinya akan dijadikan sebagai kunci untuk proses enkripsi dan dekripsi pada algoritma S-DES. Dari penelitian ini, penulis berharap bahwa metode yang diajukan ini akan memberikan tingkat keamanan yang tinggi dengan proses yang lebih sederhana dari penelitian-penelitian sebelumnya.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan di atas, maka rumusan masalah yang dikemukan yaitu:

a. Bagaimana cara membangkitkan kunci yang berupa citra menggunakan algoritma *Diffie-Helman*, kemudian mengenkripsi citra dengan kunci yang telah dibangkitkan menggunakan algoritma S-DES.
b. Bagaimana cara mendekripsi citra yang telah dienkripsi menggunakan algoritma S-DES.
c. Bagaimana analisis keamanan dari metode yang diajukan.

1.3 Batasan Masalah

Adapun batasan masalah pada penilitian ini yakni citra yang digunakan untuk *public key* adalah citra *grayscale* dan citra RGB, sedangkan citra biner tidak digunakan karena citra biner hanya memiliki dua nilai derajat keabuan yaitu 0 atau 1. Apabila *public key* yang digunakan untuk proses enkripsi menggunakan nilai derajat keabuan pada citra biner yakni 0 atau 1, maka masih ada kemungkinan keseragaman nilai derajat keabuan pada *pixel cipherimage* dari hasil proses enkripsi.

UPT Perpustakaan Universitas Jember

1.4 Tujuan Penelitian

Tujuan dari penelitian ini yaitu:

a. Membangkitkan kunci yang berupa citra menggunakan algoritma *Diffie-Helman*, kemudian mengenkripsi citra dengan kunci yang telah dibangkitkan menggunakan algoritma S-DES.
b. Mendekripsi citra yang telah dienkripsi menggunakan algoritma S-DES.
c. Menganalisis keamanan dari metode yang diajukan.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu:

a. Mampu mengenkripsi citra RGB menggunakan metode yang diajukan.
b. Mampu mendekripsi citra hasil enkripsi menggunakan metode yang diajukan.
c. Mampu menganalisis keamanan dari modifikasi algoritma S-DES.

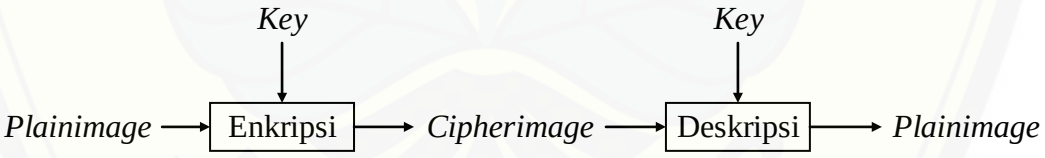
UPT Perpustakaan Universitas Jember

BAB 2. TINJAUAN PUSTAKA

2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani yaitu "*crypto*" yang artinya adalah "*secret*" atau "*rahasia*" dan "*graphy*" yang berarti "*writing*" atau "*tulisan*". Sehingga kriptografi adalah ilmu yang mempelajari tulisan rahasia atau pesan rahasia. Secara garis besar kriptografi diartikan sebagai ilmu yang mempelajari teknik untuk menyembunyikan, melindungi, dan mengamankan suatu informasi dengan cara membuat suatu bentuk baru (sandi) sehingga hanya orang tertentu yang bisa mendapatkan informasi tersebut (Munir, 2006).

Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan kunci enkripsi menjadi suatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi. Proses enkripsi dan dekripsi dilakukan menggunakan algoritma dengan beberapa parameter. Rahasia terletak di beberapa parameter yang digunakan, sehingga kunci enkripsi dan dekripsi ditentukan oleh parameter tersebut (Kromodimoeljo, 2009). Gambar 2.1 menunjukkan proses enkripsi dan dekripsi dalam kriptografi.



Gambar 2.1 Proses enkripsi dan dekripsi

Ada beberapa istilah dalam kriptografi yang harus dikenal antara lain adalah *plainimage/plaintext*, *cipherimage/ciphertext*, *key* (kunci), enkripsi, dan dekripsi. *Plainimage* merupakan data awal atau pesan yang asli berupa citra dan jika pesan berupa teks (tulisan) maka disebut plainteks (*plaintext*). Teknik yang digunakan untuk membuat *plainimage/plaintext* tidak dapat dibaca disebut enkripsi. Data hasil dari proses enkripsi disebut *cipherimage* jika data yang dienkripsi berupa *plainimage* dan cipherteks (*ciphertext*) jika data yang dienkripsi berupa plainteks.

UPT Perpustakaan Universitas Jember

Proses untuk mengembalikan *cipherimage/ciphertext* menjadi *plainimage/plaintext* disebut dekripsi (Prayudi, 2005).

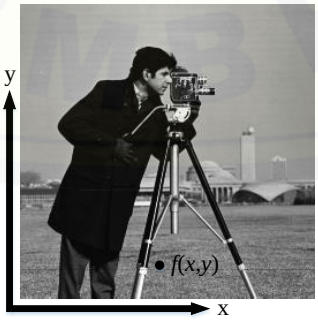
2.2 Citra

Secara harfiah citra adalah gambar pada bidang dua dimensi. Citra yang terlihat mata manusia merupakan cahaya yang direfleksikan dari sebuah objek. Sumber cahaya menerangi objek, kemudian objek memantulkan kembali sebagian dari berkas cahaya tersebut dan pantulan cahaya ditangkap dan direkam oleh alat-alat optik, seperti kamera, mata manusia, dan pemindai.

Menurut Murni (1992) output citra dari suatu sistem perekaman data dapat bersifat:

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

Secara matematis citra dapat dinyatakan dalam suatu fungsi dua dimensi $f(x,y)$ dimana x dan y adalah posisi koordinat, sedangkan f merupakan derajat keabuan pada posisi (x,y) yang sering dikenal sebagai intensitas. Derajat keabuan memiliki rentang nilai dari I_{min} sampai I_{max} atau $I_{min} < f < I_{max}$. Rentang nilai tersebut disebut sebagai selang keabuan. Ketika nilai x, y, dan f bernilai diskrit dan terbatas maka citra tersebut dikatakan sebagai citra digital. Gambar 2.2 merupakan salah satu contoh representasi citra dalam suatu fungsi $f(x,y)$.



Gambar 2.2 Citra *cameraman*

(Sumber: MATLAB *library*)

UPT Perpustakaan Universitas Jember

Agar citra dapat diolah pada komputer, citra harus direpresentasikan dalam bentuk numerik dengan nilai-nilai diskrit. Representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit disebut sebagai proses digitalisasi. Hasil citra dari representasi ini disebut citra digital. Citra digital dapat dituliskan dalam bentuk matriks berukuran *n* x *m* seperti 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 kolom (*j*) menyatakan suatu koordinat pada citra, dan *f*(*i*,*j*) menyatakan intensitas atau derajat keabuan pada titik (*i*,*j*). Setiap elemen pada matriks citra digital [*f*(*1*,*1*), *f*(*1*,*2*), …. , *f*(*n*,*m*)] disebut sebagai *pixel*. Jadi citra digital yang berukuran *n* x *m* mempunyai *nm* buah *pixel* (Dulimarta, 1997).

2.2.1 Citra RGB

Menurut (Munir, 2002), citra berwarna (RGB) merupakan suatu citra digital yang terdiri dari tiga kanal berupa matriks, yaitu kanal merah (*red*), hijau (*green*) dan biru (*blue*) dimana elemen-elemen matriksnya merepresentasikan nilai intensitas dari setiap *pixel* dan letak elemen pada matriks merepresentasikan letak *pixel* pada citra. Intensitas titik pada citra RGB merupakan gabungan dari tiga intensitas, yaitu:

a. derajat keabuan merah (*red*) ($f_r(x,y)$)
b. derajat keabuan hijau (*green*) ($f_g(x,y)$)
c. derajat keabuan biru (*blue*) ($f_b(x,y)$)

Gambar 2.3 Citra RGB "Wp*eppers*"
(Sumber: MATLAB *library*)

UPT Perpustakaan Universitas Jember

2.2.2 Citra *Grayscale*

Citra *grayscale* merupakan citra digital yang hanya memiliki satu kanal pada setiap *pixel*, dimana kanal merah (*red*) = hijau (*green*) = biru (*blue*). Citra *grayscale* memiliki variasi warna antara hitam dan putih. Citra *grayscale* memiliki nilai minimum dan maksimum yang bergantung pada jumlah bit yang digunakan. Contohnya untuk skala keabuan 8 bit, maka jumlah kemungkinan nilainya adalah $2^8 = 256$, dimana nilai minummnya adalah 0 (nol) dan nilai maksimumnya adalah $2^8 – 1 = 255$.

Suatu citra *grayscale* dapat direpresentasikan dalam bentuk matriks. Tiap elemen dalam matriks merepresentasikan nilai intensitas dari citra dan letak elemen pada matriks merepesentasikan posisi koordinat yang bersesuaian dari citra. Apabila suatu citra direpresentasikan dalam 8 bit maka pada citra tersebut terdapat 2^8 atau 256 level *grayscale* (bernilai 0 – 255), dimana 0 menunjukkan level intensitas paling gelap dan 255 menunjukkan intensitas paling terang. Warna yang dipakai adalah antara hitam sebagai warna minimum dan warna putih sebagai warna maksimum sehingga warna diantara nilai minimum dan maksimum adalah abu-abu (Sholeh, 2017).

Gambar 2.4 Citra *grayscale* "*Mandi*"
(Sumber: MATLAB *library*)

2.3 ASCII (*American Standard Code Information Interchange*)

ASCII (*American Standard Code Information Interchange*) merupakan kumpulan kode yang digunakan untuk mempermudah interaksi antara *user* dan komputer dimana kode tersebut berupa karakter seperti huruf maupun simbol. Kode

UPT Perpustakaan Universitas Jember

ASCII memiliki komponen bilangan biner sebanyak 8 bit. Dimulai dari 00000000 hingga 11111111. Total kombinasi yang dihasilkan sebanyak 256, dimulai dari kode 0 hingga 255 dalam bilangan desimal.

Pada dasarnya kode ASCII merepresentasikan kode-kode untuk angka, huruf, simbol, tombol (enter, esc, dll), karakter, dan kode komunikasi (ETX, STX, dll). Dari representasi kode-kode tersebut ASCII membaginya ke dalam tiga bagian, yakni ASCII *Control Characters*, ASCII *Printable Character*, dan *The Extended* ASCII *Codes*. Ketiga macam kode ASCII tersebut dapat dilihat pada Lampiran A (A1, A2, dan A3).

2.4 Sistem Basis Bilangan

Sistem basis bilangan adalah suatu cara untuk mewakili besaran dari suatu item fisik yang menggunakan suatu bilangan dasar atau basis tertentu. Dalam ilmu komputer ada empat jenis sistem bilangan yang dikenal, yakni bilangan *desimal*, bilangan *biner*, bilangan *oktal*, dan bilangan *hexadesimal*. Bilangan desimal merupakan suatu bilangan yang memiliki basis 10, yaitu: 0 1 2 3 4 5 6 7 8 dan 9. Bilangan biner merupakan suatu bilangan yang memiliki basis 2, yaitu: 0 dan 1. Bilangan oktal merupakan suatu bilangan yang memiliki basis 8, yaitu: 0 1 2 3 4 5 6 dan 7. Sedangkan bilangan hexadesimal merupakan suatu bilangan yang memiliki basis 16, yaitu: 0 1 2 3 4 5 6 7 8 9 A B C D E dan F (Price dan Peselnick, 1987).

Suatu sistem basis bilangan dapat dikonversikan ke dalam sistem basis bilangan yang lainnya. Berikut merupakan beberapa langkah untuk mengkonversikan sistem basis bilangan khususnya bilangan biner.

a. Konversi bilangan desimal ke bilangan biner

   Cara untuk mengkonversi bilangan desimal ke dalam bentuk bilangan biner yaitu dengan membagi bilangan desimal dengan 2 dan menyimpan sisa bagi per setiap pembagian hingga tersisa 0. Hasil konversi adalah urutan sisa bagi dari yang terakhir hingga awal.

   Contoh:

   $17_{(10)} = 10001_{(2)}$

   17 : 2 = 8 sisa 1

UPT Perpustakaan Universitas Jember

8 : 2 = 4 sisa 0

4 : 2 = 2 sisa 0

2 : 2 = 1 sisa 0

1 : 2 = 0 sisa 1

Maka hasil konversi $17_{(10)}$ adalah $10001_{(2)}$

b. Konversi bilangan biner ke bilangan desimal

Cara untuk mengkonversi bilangan biner ke dalam bentuk bilangan desimal adalah dengan mengalikan satu-satu bilangan biner dengan 2 (basis biner) pangkat 0 atau 1 atau 2 dan seterusnya dimulai dari bilangan biner yang paling kanan. Selanjutnya hasil dari perkalian-perkalian tersebut dijumlahkan.

Contoh:

$10001_{(2)} = 17_{(10)}$

$1 \times 2^0 = 1$

$0 \times 2^1 = 0$

$0 \times 2^2 = 0$

$0 \times 2^3 = 0$

$1 \times 2^4 = 16$ +

$17_{(10)}$

2.5 Algoritma *Diffie-Hellman*

Algoritma *Diffie-Hellman* diperkenalkan pertama kali oleh Whitfield Diffie dan Martin Hellman pada tahun 1975. *Diffie-Hellman* adalah algoritma matematika yang digunakan untuk pertukaran kunci antara pengirim dan penerima pesan. Algoritma *Diffie-Hellman* tidak berdasarkan pada proses enkripsi maupun dekripsi, melainkan dalam proses matematika yang dilakukan untuk mendapatkan kunci rahasia (*secret key*) yang dapat disebarkan secara bebas tanpa harus khawatir akan keamanannya.

Seorang peretas (*hacker*), meretas kunci kriptografi ketika proses pengiriman pesan tersandi dan kunci dekripsinya dari pengirim ke penerima, dengan algoritma *Diffie-Hellman* peretas akan kesulitan untuk mendapatkan kunci tersebut karena dalam melakukan pengiriman kunci, kunci yang dikirim antara pengirim dan

UPT Perpustakaan Universitas Jember

penerima pesan bukan kunci yang digunakan untuk proses enkripsi maupun dekripsi melainkan hanya kunci yang telah diproses sebagian untuk mendapatkan kunci enkripsi dan dekripsi, sehingga hanya pengirim dan penerima pesan yang memiliki kunci tersebut. Oleh karena itu algoritma *Diffie-Hellman* juga dapat digunakan untuk membangkitkan suatu kunci simetris dalam kriptografi.

Berikut adalah langkah-langkah dalam melakukan pertukaran dan pembangkitan kunci menggunakan algoritma *Diffie-Hellman* dengan alur proses seperti pada Gambar 2.5.

a. Menentukan bilangan prima p, dan bilangan bulat tak nol yang kurang dari p yaitu g. Bilangan p dan g dapat diketahui oleh orang lain (bersifat publik).
b. Menentukan sebarang bilangan bulat positif x untuk pengirim pesan, dan y untuk penerima pesan. Bilangan x dan y ini tidak boleh diketahui oleh orang lain (*private key*).
c. Pengirim menghitung $A = g^x \bmod p$ dan penerima menghitung $B = g^y \bmod p$.
d. Bilangan A dan B merupakan *shared key* yang kemudian ditukarkan, A diberikan ke penerima dan B diberikan ke pengirim.
e. Pengirim menghitung $K_A = B^x \bmod p$ dan penerima menghitung $K_B = A^y \bmod p$

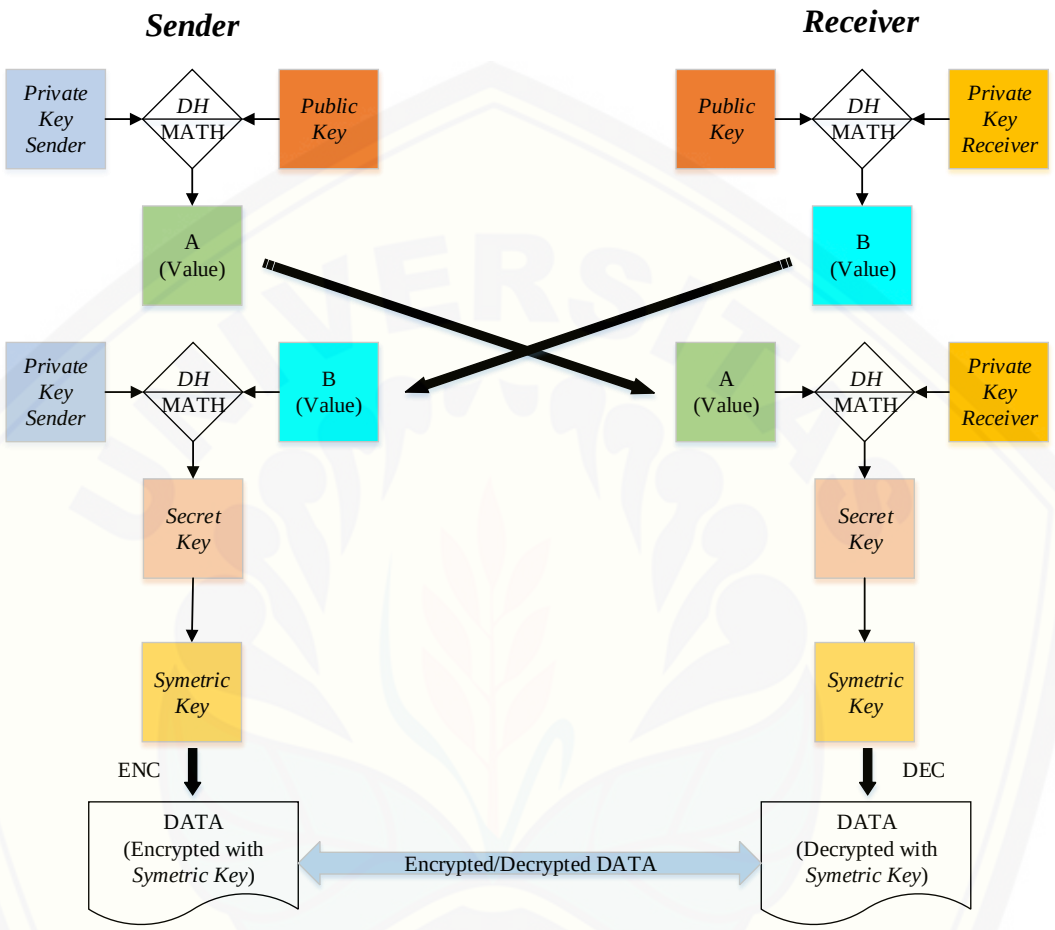
Berdasarkan hukum aljabar nilai K_A sama dengan K_B dimana $K_A = K_B = K$ karena $K_A = B^x \bmod p = (g^y)^x \bmod p = (g^y)^x \bmod p = g^{yx} \bmod p = g^{xy} \bmod p = (g^x)^y \bmod p = A^y \bmod p = K_B$. Sehingga pengirim dan penerima mengetahui *secret key* yang nantinya akan digunakan sebagai kunci untuk proses enkripsi dan dekripsi pada algoritma simetri dalam kriptografi (Abbadi, 2016).

Berikut adalah contoh penggunaan algoritma *Diffie-Hellman*.

a. Dony dan Bima menetapkan $p = 23$ dan $g = 12$
b. Dony memilih nilai $x = 7$ dan menghitungnya
   $A = 12^7 \bmod 23 = 16$
c. Bima memilih nilai $y = 9$ dan menghitungnya
   $B = 12^9 \bmod 23 = 4$
d. Dony mengirim nilai A ke Bima, dan Bima mengirim nilai B ke Dony
e. Dony melakukan perhitungan $K_A = 4^7 \bmod 23 = 8$
f. Bima melakukan perhitungan $K_B = 16^9 \bmod 23 = 8$

UPT Perpustakaan Universitas Jember

Angka 8 adalah *secret key* yang dapat digunakan sebagai kunci untuk proses enkripsi dan dekripsi pada algoritma simetri dalam kriptografi.



Gambar 2.5 Alur proses pertukaran kunci algoritma *Diffie-Hellman*

2.6 Algoritma S-DES (*Simplified Data Encryption Standard*)

Algoritma S-DES (*Simplified Data Encryption Standard*) dikenalkan pertama kali oleh Edward Schaefer dari Universitas Santa Clara. Algoritma S-DES merupakan penyederhanaan dari algoritma DES (*Data Encryption Standard*). Algoritma S-DES memiliki keunggulan jika dibandingkan dengan algoritma DES yaitu dari segi kecepatan proses enkripsi dan dekripsi serta kesederhanaannya karena algoritma S-DES hanya menggunakan 10 bit kunci untuk proses enkripsi dan dekripsinya. Algoritma S-DES mengambil 8 bit blok dari plainteks dan 10 bit kunci yang digunakan untuk proses enkripsi. Hasil dari proses enkripsi ini akan

UPT Perpustakaan Universitas Jember

menghasilkan 8 bit blok cipherteks. Sedangkan proses dekripsi algoritma S-DES mengambil 8 bit blok cipherteks dan 10 bit kunci yang sama yang digunakan saat proses enkripsi. Hasil dari proses dekripsi ini akan menghasilkan 8 bit blok plainteks (Garg, 2015).

Dalam proses enkripsi dan dekripsi, ada lima fungsi yang digunakan, yakni:

a. Permutasi awal (*IP*)
b. Fungsi komplek f_K yang melibatkan operasi permutasi dan subtitusi. Fungsi f_K ini bergantung pada kunci yang dimasukkan. Sub kunci yang digunakan adalah sub kunci K_1.
c. Fungsi permutasi sederhana (*SW*) yang merubah posisi dari dua bagian data
d. Fungsi komplek f_K. Sub kunci yang digunakan adalah sub kunci K_2.
e. Fungsi permutasi yang merupakan invers dari permutasi awal (IP^{-1})

Secara singkat, proses enkripsi dapat ditulis dalam fungsi komposisi sebagai berikut:

$$IP^{-1} \circ f_{K_2} \circ SW \circ f_{K_1} \circ IP \tag{2.1}$$

Fungsi komposisi tersebut juga dapat ditulis sebagai berikut:

$$\text{Cipherteks} = IP^{-1}\,(f_{K_2}\,(SW\,(f_{K_1}\,(IP\,(\text{plainteks})))))\tag{2.2}$$

dimana

$$K_1 = P8\,(Shift\,(P10\,(\text{kunci})))$$

$$K_2 = P8\,(Shift\,(Shift\,(Shift\,(P10\,(\text{kunci})))))$$

sehingga proses dekripsi dapat ditulis sebagai berikut:

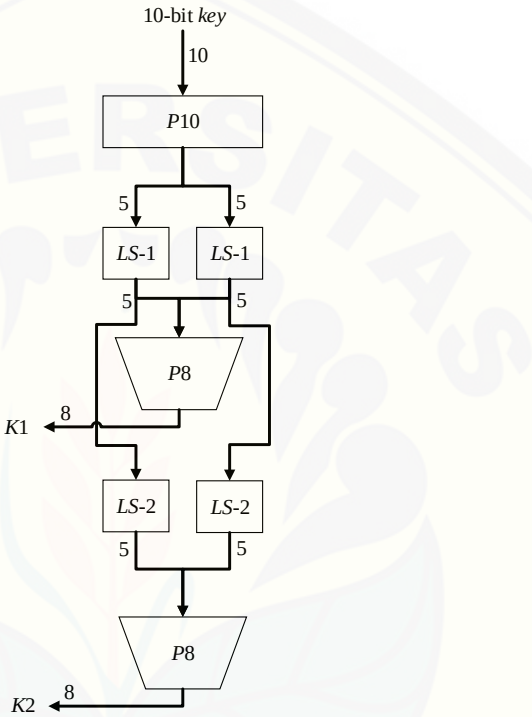
$$\text{Plainteks} = IP^{-1}\,(f_{K_1}^{-1}\,(SW^{-1}\,(IP\,(\text{cipherteks}))))\tag{2.3}$$

pada persamaan (2.2) yang bertindak sebagai *domain* adalah plainteks yang direlasikan ke *kodomain* yaitu cipherteks dengan menggunakan lima fungsi yang telah disebutkan sebelumnya. Perlakuan fungsinya secara berurutan dari *IP*, f_{K_1}, SW, f_{K_2}, dan IP^{-1} untuk persamaan (2.2), sedangkan pada persamaan (2.3) merupakan invers dari persamaan (2.2) (Hardjo, 2016).

UPT Perpustakaan Universitas Jember

2.6.1 Pembangkitan Kunci S-DES

Algoritma S-DES bergantung pada penggunaan 10 bit kunci yang dibagikan antara pengirim dan penerima. Dari 10 bit kunci ini akan diperoleh 8 bit sub kunci yang nantinya akan digunakan untuk proses enkripsi maupun dekripsi. Gambar 2.6 merupakan alur proses pembangkitan kunci S-DES.



Gambar 2.6 Alur proses pembangkitan kunci S-DES

Langkah-langkah pembangkitan kunci pada Gambar 2.6 dijelaskan sebagai berikut:

a. Kunci 10-bit yang diasumsikan sebagai (k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) akan dilakukan permutasi $P10$. Permutasi $P10$ didefinisikan sebagai berikut:
   $P10(k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9, k_{10}) = (k_3, k_5, k_2, k_7, k_4, k_{10}, k_1, k_9, k_8, k_6)$
   $P10$ dapat ditulis dalam Tabel 2.1.

Tabel 2.1 Tabel Permutasi $P10$

$P10$									
3	5	2	7	4	10	1	9	8	6

UPT Perpustakaan Universitas Jember

Tabel tersebut dibaca dari kiri ke kanan. Setiap posisi pada tabel memberikan identitas dari bit masukan yang menghasilkan bit keluaran pada posisi tersebut. Sehingga bit keluaran pertama adalah bit ke 3 dari bit masukan.

b. Lakukan pergeseran kiri (*LS*-1) untuk masing-masing 5-bit awal dan 5-bit akhir. *LS*-1 di sini maksudnya adalah menggeser setiap bit ke kiri sejauh satu langkah.
c. Lakukan operasi permutasi *P*8 dari hasil *LS*-1 seperti langkah (a) dengan mengikuti aturan pada Tabel 2.2. Operasi ini bertujuan untuk mengambil 8-bit dari 10-bit kunci.

Tabel 2.2 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 (*LS*-2) untuk masing-masing 5-bit awal dan 5-bit akhir. *LS*-2 di sini maksudnya adalah menggeser setiap bit ke kiri sejauh dua langkah.
f. Lakukan operasi permutasi *P*8 dari hasil langkah (e) untuk memperoleh sub kunci K_2.

2.6.2 Enkripsi S-DES

Dalam melakukan proses enkripsi menggunakan algoritma S-DES digunakan lima fungsi. Berikut adalah langkah-langkah untuk menggunakan lima fungsi tersebut.

a. Permutasi Awal

Masukan 8-bit plainteks, kemudian dilakukan operasi permutasi *IP* menggunakan fungsi *IP* seperti Tabel 2.3 berikut.

Tabel 2.3 Tabel Permutasi *IP*

IP							
2	6	3	1	4	8	5	7

UPT Perpustakaan Universitas Jember

b. Fungsi f_K

Fungsi f_K adalah fungsi yang kompleks dari algoritma S-DES, karena pada fungsi ini melibatkan operasi fungsi permutasi dan subtitusi. Fungsi f dapat dituliskan seperti pada persamaan (2.4). Misalkan L dan R adalah 4-bit paling kiri dan 4-bit paling kanan dari 8-bit hasil operasi permutasi awal (IP) dan F dipetakan (tidak harus satu-satu) dari deretan 4-bit yang satu ke deretan 4-bit yang lainnya, maka

$$f_K(L,R) = (L \oplus F(R,SK),R) \tag{2.4}$$

dimana SK adalah sub kunci (K_1, K_2) dan $\oplus$ adalah operasi XOR.

Secara detail, proses perhitungan $F(R,SK)$ dapat dideskripsikan sebagai berikut:

1. R adalah 4-bit paling kanan dari 8-bit hasil operasi permutasi awal (IP).
2. Operasi pertama yakni operasi ekspansi/permutasi (E/P) seperti terlihat pada Tabel 2.4. Operasi ini menyebabkan R yang awalnya berukuran 4-bit menjadi 8-bit.

Tabel 2.4 Tabel Ekspansi/Permutasi (E/P)

E/P							
4	1	2	3	2	3	4	1

3. Hasil dari langkah (2) kemudian dilakukan operasi XOR dengan sub kunci K_1 yang nantinya akan menghasilkan keluaran berukuran 8-bit.
4. 4-bit paling kiri dari hasil pada langkah (3) akan diproses dengan S-Box $S0$ (Tabel 2.5), dan 4-bit paling kanan akan diproses dengan S-Box $S1$ (Tabel 2.6).

Tabel 2.5 Tabel S-Box $S0$

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

UPT Perpustakaan Universitas Jember

Tabel 2.6 Tabel 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. Hasil 2-bit dari masing-masing operasi pada S-Box kemudian digabungkan sehingga menjadi 4-bit. 4-bit tersebut kemudian dilakukan operasi permutasi *P*4 seperti pada Tabel 2.7.

Tabel 2.7 Tabel Permutasi *P*4

*P*4			
2	4	3	1

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

c. Fungsi Permutasi Sederhana (*SW*)

Fungsi permutasi sederhana (*SW*) berfungsi untuk menukar 4-bit paling kiri dari hasil fungsi f_K sebelumnya dengan 4-bit paling kanan.

d. Fungsi f_K

Pada perulangan fungsi f_K ini proses perhitungannya sama dengan proses perhitungan fungsi f_K yang pertama, namun sub kunci yang digunakan adalah sub kunci K_2.

e. Permutasi Akhir

Hasil dari langkah (d) kemudian dilakukan operasi permutasi IP^{-1} seperti Tabel 2.9 dan akan diperoleh cipherteks berukuran 8-bit.

UPT Perpustakaan Universitas Jember

Tabel 2.9 Tabel Permutasi IP^{-1}

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

2.6.3 Dekripsi S-DES

Proses dekripsi pada algoritma S-DES hampir sama dengan proses enkripsinya. Fungsi yang digunakan untuk proses dekripsi juga sama seperti pada proses enkripsi. Yang membedakan adalah pada saat proses perhitungan permutasi awal dan proses fungsi f_K. Pada saat proses permutasi awal masukan 8-bit berupa cipherteks dan pada saat perhitungan proses fungsi f_K yang pertama menggunakan sub kunci K_2 sedangkan perhitungan proses fungsi f_K yang kedua menggunakan sub kunci K_1.

2.7 Analisis Keamanan

Dalam proses perlindungan data pada sebuah citra (gambar), terdapat beberapa metode yang dapat digunakan untuk menganalisis keamanan dari suatu algoritma yang digunakan. Berikut ini merupakan beberapa analisis keamanan yang digunakan.

2.7.1 Analisis Diferensial

Analisis ini digunakan untuk mengetahui perbedaan dari dua buah citra, dengan cara menghitung nilai dari *number of pixels change rate* (*NPCR*). Hal ini dilakukan guna mengetahui apakah pada setiap *pixel* citra hasil enkripsi terdapat perubahan elemen warna dengan citra sebelum dienkripsi. Jika nilai *NPCR* antara *chiperimage* dan *plainimage* lebih dari 90%, maka dapat dikatakan bahwa *cipherimage* aman dari serangan *statistic*. Berikut merupakan cara untuk menghitung *number of pixels change rate* (*NPCR*).

$$NPCR = \frac{\sum_{i,j,k} D(i,j,k)}{W \times H \times d} \times 100\% \qquad (2.7)$$

UPT Perpustakaan Universitas Jember

dimana W melambangkan panjang citra, H melambangkan tinggi citra, dan d melambangkan dimensi citra. Nilai $D(i,j,k)$ ditentukan sebagai berikut.

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

dimana $C(i,j,k)$ dan $C'(i,j,k)$ masing-masing menunjukkan nilai derajat keabuan dari baris ke *i*, kolom ke *j* , dan kanal ke *k* dari citra *C* dan *C'* (Akhavan A. dkk., 2011).

2.7.2 Analisis Sensitivitas Kunci

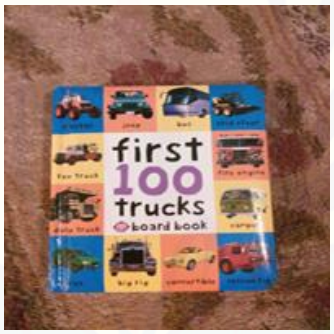
Analisis ini digunakan untuk mengetahui seberapa besar pengaruh kunci yang digunakan terhadap hasil enkripsi suatu algoritma jika kunci tersebut sedikit dirubah. Dua buah kunci yang sedikit berbeda akan menghasilkan *cipherimage* yang sangat berbeda jika tingkat sensitivitas kuncinya besar. Besarnya kesensitivitasan suatu kunci sangat berpengaruh pada hasil dekripsi *cipherimage*, karena jika ada perbedaan kunci antara proses enkripsi dan dekripsi maka tidak akan memperoleh *plainimage* yang diinginkan. Pengukuran perbedaan citra diukur menggunakan persamaan (2.7) (Irfan, 2016).

UPT Perpustakaan Universitas Jember

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang penulis gunakan dalam penelitian ini adalah lima buah citra dimana tiga buah citra RGB dan dua buah citra *grayscale* yang dijadikan sebagai *plainimage*, dan sepuluh buah citra dimana enam buah citra RGB dan empat buah citra *grayscale* yang dijadikan sebagai kunci publik (*public key*) dimana untuk setiap *plainimage* digunakan dua buah citra *public key*. Berikut adalah sampel citra RGB dari *plainimage* dan *public key* yang penulis gunakan dalam penelitian dengan ukuran dimensi yaitu 200×200 *pixel*.



Gambar 3.1 *trucks*.png

(Sumber: MATLAB *library*)

Gambar 3.2 *building*.png

(Sumber: MATLAB *library*)

UPT Perpustakaan Universitas Jember

Gambar 3.3 *holdingCup*.png
(Sumber: MATLAB *library*)

3.2 Langkah-langkah Penelitian

Langkah-langkah penelitian yang dilakukan dalam penelitian ini adalah sebagai berikut:

a. Studi Literatur

Tahap ini dilakukan sebagai pembelajaran mengenai teori-teori terkait dengan penelitian yang dilakukan. Teori-teori tersebut adalah citra RGB, citra *grayscale*, algoritma S-DES, dan algoritma pertukaran kunci *Diffie-Hellman*. Literatur yang digunakan berupa buku, jurnal, artikel, dan sumber lainnya.

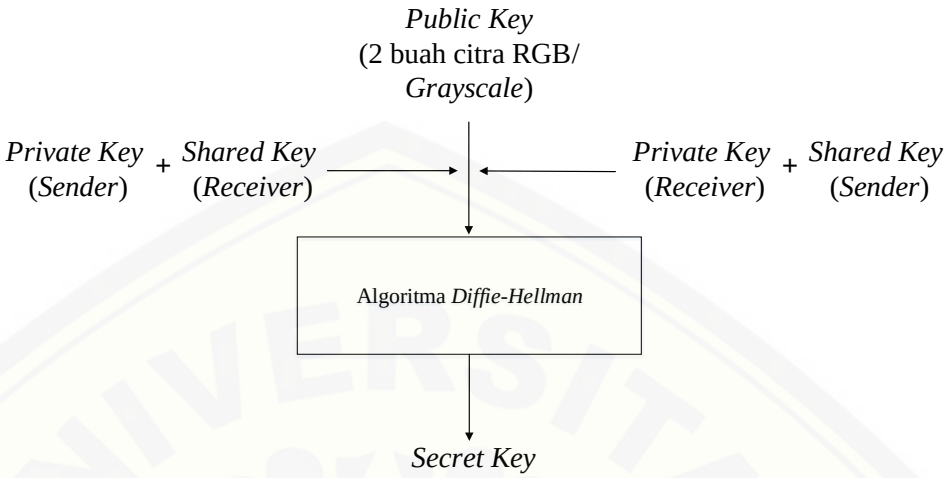
b. Analisis Data

Langkah-langkah berikut ini akan diterapkan pada masing-masing *pixel* di setiap kanal dari citra RGB.

1) Proses Pembangkitan Kunci

Proses pembangkitan kunci dilakukan sesuai dengan alur dari Gambar 3.4. Proses tersebut akan diuraikan sebagai berikut.

UPT Perpustakaan Universitas Jember



Gambar 3.4 Proses pembangkitan kunci

a) Pembentukan nilai *p* dan *g* dari *public key*

Misalkan terdapat dua buah citra yaitu citra A dan citra B. Dua buah citra yang diinput nantinya akan dijadikan sebagai *public key* dalam proses pembangkitan kunci menggunakan algoritma *Diffie-Hellman*. Nilai derajat keabuan pada setiap *pixel* dari *public key* tersebut akan dijadikan nilai *p* dan *g* (subbab 2.4 Algoritma *Diffie-Hellman*). Ada beberapa perlakuan mengenai penentuan nilai *p* dan *g* dari dua buah citra tersebut. Berikut merupakan beberapa perlakuan dalam penentuan nilai *p* dan *g*.

(1) Jika nilai derajat keabuan pada setiap *pixel* citra A atau citra B sama dengan 0 (nol) maka nilai derajat keabuan *pixel* citra A atau citra B akan ditambah 2 (dua).

(2) Jika nilai derajat keabuan pada setiap *pixel* citra A sama dengan citra B maka nilai derajat keabuan *pixel* citra A akan ditambah 2 (dua).

(3) Jika nilai derajat keabuan pada setiap *pixel* citra A lebih besar dari citra B maka nilai derajat keabuan *pixel* citra A akan dijadikan sebagai *p* dan nilai derajat keabuan *pixel* citra B akan dijadikan sebagai *g*, begitupun sebaliknya.

UPT Perpustakaan Universitas Jember

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

b) Pembentukan *shared key*

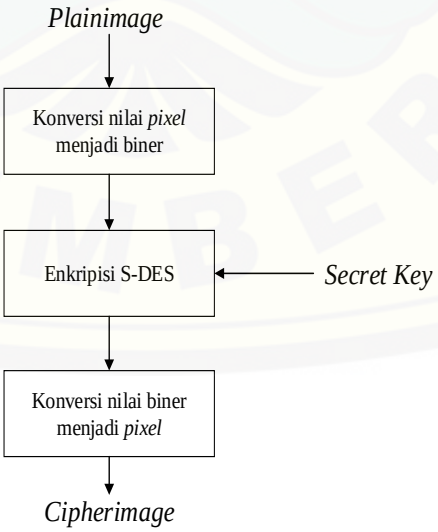
Untuk mendapatkan *shared key*, setiap nilai *p* dan *g* dioperasikan dengan *private key* sesuai subbab 2.4 Algoritma *Diffie-Hellman* pada poin c. Hasil dari proses pembentukan *shared key* berupa sebuah citra RGB atau citra *grayscale*.

c) Pembentukan *secret key*

Setelah didapatkan nilai *p* dan *g* serta *shared key*, dilakukan proses pembangkitan kunci menggunakan algoritma *Diffie-Hellman* (seperti yang dijelaskan pada subbab 2.4 Algoritma *Diffie-Hellman*). Hasil dari proses pembangkitan kunci ini penulis sebut sebagai *secret key* yang berupa sebuah citra RGB atau citra *grayscale*.

2) Proses Enkripsi

Proses enkripsi dilakukan seperti pada Gambar 3.5. Proses tersebut akan diuraikan sebagai berikut.



Gambar 3.5 Proses enkripsi

UPT Perpustakaan Universitas Jember

a) Konversi nilai *pixel* menjadi biner

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

b) Enkripsi S-DES

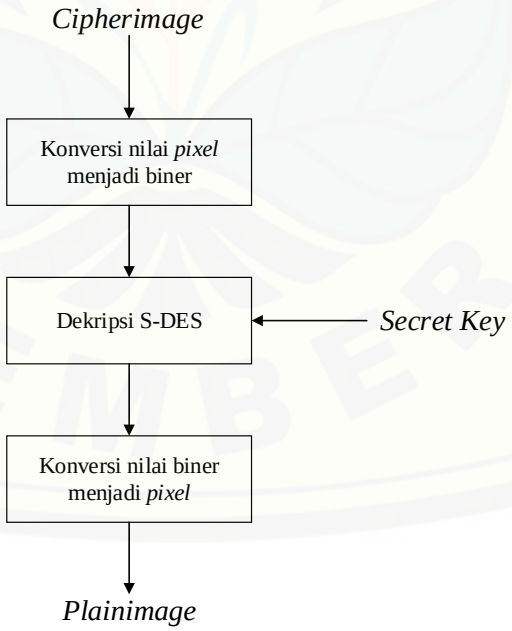
Nilai derajat keabuan pada setiap *pixel* akan dienkripsi dengan algoritma S-DES (seperti pada penjelasan subsubbab 2.6.2 Enkripsi S-DES) dengan kunci yang telah dibangkitkan sebelumnya menggunakan algoritma *Diffie-Hellman* (*secret key*).

c) Konversi nilai biner menjadi *pixel*

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

3) Proses Dekripsi

Proses dekripsi dilakukan sesuai dengan proses pada Gambar 3.6. Proses dekripsi dilakukan pada setiap *pixel* pada *cipherimage*.



Gambar 3.6 Proses dekripsi

UPT Perpustakaan Universitas Jember

a) Konversi nilai *pixel* menjadi biner

Setiap nilai *pixel* pada *cipherimage* dikonversi ke dalam bentuk biner.

b) Dekripsi S-DES

Nilai derajat keabuan pada setiap *pixel* akan didekripsi dengan algoritma S-DES (seperti pada penjelasan subsubbab 2.6.3 Dekripsi S-DES) dengan kunci yang telah dibangkitkan sebelumnya menggunakan algoritma *Diffie-Hellman* (*secret key*).

c) Konversi nilai biner menjadi *pixel*

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

c. Perancangan Program

Pada tahap ini dilakukan perancangan desain GUI (*Guide User Interface*) dengan menggunakan software MATLAB 2016b. Perancangan desain meliputi pengaturan tata letak tombol, pengaturan warna, dan pengaturan latar belakang. Hal ini dilakukan agar form *layout* dan desain interaksi program lebih menarik.

d. Pembuatan Program

Pembuatan program menggunakan konsep matriks sebagai pembangkit citra pada MATLAB, melakukan proses pembangkitan kunci menggunakan algoritma *Diffie-Hellman* kemudian dilanjutkan melakukan proses enkripsi menggunakan algoritma S-DES dengan kunci yang telah dibangkitkan untuk setiap *pixel*.

e. Analisis Hasil

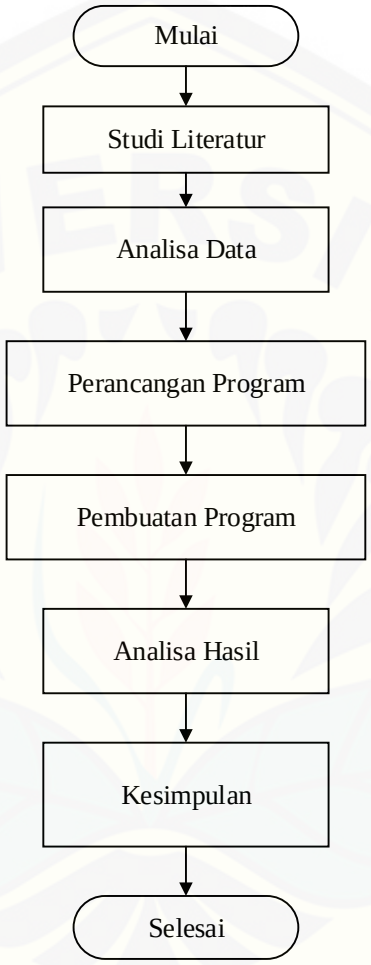
Pada tahap ini dilakukan pengujian dan menganalisa program yang telah dibuat untuk memastikan setiap proses telah berjalan sesuai dengan konsep teori dan sesuai dengan hasil yang diinginkan.

f. Kesimpulan

Mengambil kesimpulan dari penelitian yang telah dilakukan, yaitu dengan menganalisis proses enkripsi dan dekripsi suatu *plainimage* dan *cipherimage* menggunakan metode yang diajukan.

UPT Perpustakaan Universitas Jember

Flowchart dari langkah-langkah penelitian yang dilakukan dapat diamati pada Gambar 3.7.



Gambar 3.7 *Flowchart* penelitian

UPT Perpustakaan Universitas Jember

BAB 5. PENUTUP

5.1 Kesimpulan

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

a. Algoritma *Diffie-Hellman* mampu digunakan untuk membangkitkan kunci simetris dari beberapa kunci *input* yang bersifat rahasia (*secret key*), yang dapat disebarkan tanpa harus khawatir akan keamanannya.
b. Proses enkripsi citra RGB dan citra *grayscale* mampu memberikan keamanan sebab modifikasi kunci yang berupa citra memungkinkan hasil enkripsi sebuah citra lebih bervariasi.
c. Proses dekripsi sudah berjalan sesuai yang diharapkan, dibuktikan dengan *cipherimage* yang dapat dikembalikan menjadi *plainimage* tanpa menghilangkan sedikit pun informasi awal.
d. Metode yang diajukan penulis aman terhadap serangan statistik dan memiliki kunci yang sensitif, hal ini dibuktikan dari hasil analisis keamanan yang telah dilakukan yakni analisis diferensial dan analisis sensitivitas kunci.

5.2 Saran

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

UPT Perpustakaan Universitas Jember

DAFTAR PUSTAKA

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

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

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

Garg, P., S. Varshney, dan M. Bhardwaj. 2015. Cryptanalisis of Simplified Data Encryption Standard Using Genetic Algorithm. *American Journal of Networks and Communications* 4(3): 32-36.

Gunawan, M. I. *Penggunaan Algoritma Diffie-Hellman dalam Melakukan Pertukaran Kunci*.http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2012-2013/Makalah2012/Makalah-IF2091-2012-034.pdf. [Diakses pada 02 September 2017].

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.

Kromodimoeljo, S. 2009. *Teori & Aplikasi Kriptografi*. Jakarta: SPK IT Consulting.

UPT Perpustakaan Universitas Jember

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

Munir, R. 2002. *Pengolahan Citra Digital*. Bandung: Departemen Teknik Informatika ITB.

Munir, R. 2006. *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.

Prayudi, Y., dan I. Halik. 2005. Studi Analisis Algoritma Rivest Code 6 (RC6) Dalam Enkripsi/Dekripsi Data. *Seminar Nasional Aplikasi Teknologi Informasi 2005* (*SNATI 2005*). Yogyakarta.

Price, W. T., dan C. Peselnick. 1987. *Elements of Data Processing Mathematics*. *3rd edition*. USA: Harcourt.

Sholeh, H. 2017. *Penerapan Modifikasi Algoritma Simplified Data Encryption Standard (S-DES) Dengan Stream Cipher Pada Citra Grayscale*. Tidak dipublikasikan. Skripsi. Jember: Jurusan Matematika Fakultas MIPA Universitas Jember.

Tamam, M. T., W. Dwiono, dan T. Hartono. 2010. *Penerapan Algortima ElGamal untuk Pengamanan File Citra*. Purwokerto. Universitas Muhamadiyah Purwokerto.

UPT Perpustakaan Universitas Jember

LAMPIRAN

LAMPIRAN A. Kode ASCII

A1. Kode ASCII *Control Character* (Character code 0-31)

Dec	*Hex*	*Bin*	*Symbol*	*Dec*	*Hex*	*Bin*	*Symbol*
0	0	0000 0000	NUL	16	10	0001 0000	DLE
1	1	0000 0001	SOH	17	11	0001 0001	DC1
2	2	0000 0010	STX	18	12	0001 0010	DC2
3	3	0000 0011	ETX	19	13	0001 0011	DC3
4	4	0000 0100	EOT	20	14	0001 0100	DC4
5	5	0000 0101	ENQ	21	15	0001 0101	NAK
6	6	0000 0110	ACK	22	16	0001 0110	SYN
7	7	0000 0111	BEL	23	17	0001 0111	ETB
8	8	0000 1000	BS	24	18	0001 1000	CAN
9	9	0000 1001	HT	25	19	0001 1001	EM
10	A	0000 1010	LF	26	1A	0001 1010	SUB
11	B	0000 1011	VT	27	1B	0001 1011	ESC
12	C	0000 1100	FF	28	1C	0001 1100	FS
13	D	0000 1101	CR	29	1D	0001 1101	GS
14	E	0000 1110	SO	30	1E	0001 1110	RS
15	F	0000 1111	SI	31	1F	0001 1111	US

A2. Kode ASCII *Printable Character* (Character code 32-127)

Dec	*Hex*	*Bin*	*Symbol*	*Dec*	*Hex*	*Bin*	*Symbol*
32	20	0010 0000	Spasi	46	2E	0010 1110	.
33	21	0010 0001	!	47	2F	0010 1111	/
34	22	0010 0010	”	48	30	0011 0000	0
35	23	0010 0011	#	49	31	0011 0001	1
36	24	0010 0100	$	50	32	0011 0010	2
37	25	0010 0101	%	51	33	0011 0011	3
38	26	0010 0110	&	52	34	0011 0100	4
39	27	0010 0111	’	53	35	0011 0101	5
40	28	0010 1000	(	54	36	0011 0110	6
41	29	0010 1001	)	55	37	0011 0111	7
42	2A	0010 1010	*	56	38	0011 1000	8
43	2B	0010 1011	+	57	39	0011 1001	9
44	2C	0010 1100	,	58	3A	0011 1010	:
45	2D	0010 1101	-	59	3B	0011 1011	;

UPT Perpustakaan Universitas Jember

Dec	*Hex*	*Bin*	*Symbol*	*Dec*	*Hex*	*Bin*	*Symbol*
60	3C	0011 1100	<	100	64	0110 0100	d
61	3D	0011 1101	=	101	65	0110 0101	e
62	3E	0011 1110	>	102	66	0110 0110	f
63	3F	0011 1111	?	103	67	0110 0111	g
64	40	0100 0000	@	104	68	0110 1000	h
65	41	0100 0001	A	105	69	0110 1001	i
66	42	0100 0010	B	106	6A	0110 1010	j
67	43	0100 0011	C	107	6B	0110 1011	k
68	44	0100 0100	D	108	6C	0110 1100	l
69	45	0100 0101	E	98	62	0110 0010	b
71	47	0100 0111	G	99	63	0110 0011	c
72	48	0100 1000	H	100	64	0110 0100	d
73	49	0100 1001	I	101	65	0110 0101	e
74	4A	0100 1010	J	102	66	0110 0110	f
75	4B	0100 1011	K	103	67	0110 0111	g
76	4C	0100 1100	L	104	68	0110 1000	h
77	4D	0100 1101	M	105	69	0110 1001	i
78	4E	0100 1110	N	106	6A	0110 1010	j
79	4F	0100 1111	O	107	6B	0110 1011	k
80	50	0101 0000	P	108	6C	0110 1100	l
81	51	0101 0001	Q	109	6D	0110 1101	m
82	52	0101 0010	R	110	6E	0110 1110	n
83	53	0101 0011	S	111	6F	0110 1111	o
84	54	0101 0100	T	112	70	0111 0000	p
85	55	0101 0101	U	113	71	0111 0001	q
86	56	0101 0110	V	114	72	0111 0010	r
87	57	0101 0111	W	115	73	0111 0011	s
88	58	0101 1000	X	116	74	0111 0100	t
89	59	0101 1001	Y	117	75	0111 0101	u
90	5A	0101 1010	Z	118	76	0111 0110	v
91	5B	0101 1011	[	119	77	0111 0111	w
92	5C	0101 1100	\	120	78	0111 1000	X
93	5D	0101 1101	]	121	79	0111 1001	Y
94	5E	0101 1110	^	122	7A	0111 1010	Z
95	5F	0101 1111	_	123	7B	0111 1011	{
96	60	0110 0000	~	124	7C	0111 1100	|
97	61	0110 0001	a	125	7D	0111 1101	}
98	62	0110 0010	b	126	7E	0111 1110	~
99	63	0110 0011	c	127	7F	0111 1111	Delete

UPT Perpustakaan Universitas Jember

A3. *The Extended* ASCII *Codes* (Character code 128-255)

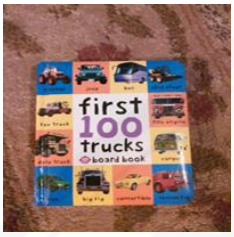
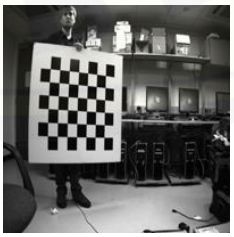
Dec	*Hex*	*Bin*	*Symbol*	*Dec*	*Hex*	*Bin*	*Symbol*
128	80	1000 0000	€	167	A7	1010 0111	§
129	81	1000 0001		168	A8	1010 1000	¨
130	82	1000 0010	‚	169	A9	1010 1001	©
131	83	1000 0011	Ƒ	170	AA	1010 1010	ª
132	84	1000 0100	„	171	AB	1010 1011	«
133	85	1000 0101	…	172	AC	1010 1100	¬
134	86	1000 0110	†	173	AD	1010 1101	
135	87	1000 0111	‡	174	AE	1010 1110	®
136	88	1000 1000	ˆ	175	AF	1010 1111	¯
137	89	1000 1001	‰	176	B0	1011 0000	°
138	8A	1000 1010	Š	177	B1	1011 0001	±
139	8B	1000 1011	‹	178	B2	1011 0010	²
140	8C	1000 1100	Œ	179	B3	1011 0011	³
141	8D	1000 1101		180	B4	1011 0100	´
142	8E	1000 1110	Ž	181	B5	1011 0101	µ
143	8F	1000 1111		182	B6	1011 0110	¶
144	90	1001 0000		183	B7	1011 0111	·
145	91	1001 0001	‘	184	B8	1011 1000	¸
146	92	1001 0010	’	185	B9	1011 1001	¹
147	93	1001 0011	“	186	BA	1011 1010	º
148	94	1001 0100	”	187	BB	1011 1011	»
149	95	1001 0101	•	188	BC	1011 1100	¼
150	96	1001 0110	–	189	BD	1011 1101	½
151	97	1001 0111	—	190	BE	1011 1110	¾
152	98	1001 1000	˜	191	BF	1011 1111	¿
153	99	1001 1001	™	192	C0	1100 0000	À
154	9A	1001 1010	Š	193	C1	1100 0001	Á
155	9B	1001 1011	›	194	C2	1100 0010	Â
156	9C	1001 1100	Œ	195	C3	1100 0011	Ã
157	9D	1001 1101		196	C4	1100 0100	Ä
158	9E	1001 1110	Ž	197	C5	1100 0101	Å
159	9F	1001 1111	Ÿ	198	C6	1100 0110	Æ
160	A0	1010 0000		199	C7	1100 0111	Ç
161	A1	1010 0001	¡	200	C8	1100 1000	È
162	A2	1010 0010	¢	201	C9	1100 1001	É
163	A3	1010 0011	£	202	CA	1100 1010	Ê
164	A4	1010 0100	¤	203	CB	1100 1011	Ë
165	A5	1010 0101	¥	204	CC	1100 1100	Ì
166	A6	1010 0110	¦				

UPT Perpustakaan Universitas Jember

Dec	*Hex*	*Bin*	*Symbol*	*Dec*	*Hex*	*Bin*	*Symbol*
205	CD	1100 1101	Í	231	E7	1110 0111	Ç
206	CE	1100 1110	Î	232	E8	1110 1000	È
207	CF	1100 1111	Ï	233	E9	1110 1001	É
208	D0	1101 0000	Đ	234	EA	1110 1010	Ê
209	D1	1101 0001	Ñ	235	EB	1110 1011	Ë
210	D2	1101 0010	Ò	236	EC	1110 1100	Ì
211	D3	1101 0011	Ó	237	ED	1110 1101	Í
212	D4	1101 0100	Ô	238	EE	1110 1110	Î
213	D5	1101 0101	Õ	239	EF	1110 1111	Ï
214	D6	1101 0110	Ö	240	F0	1111 0000	Đ
215	D7	1101 0111	×	241	F1	1111 0001	Ñ
216	D8	1101 1000	Ø	242	F2	1111 0010	Ò
217	D9	1101 1001	Ù	243	F3	1111 0011	Ó
218	DA	1101 1010	Ú	244	F4	1111 0100	Ô
219	DB	1101 1011	Û	245	F5	1111 0101	Õ
220	DC	1101 1100	Ü	246	F6	1111 0110	Ö
221	DD	1101 1101	Ý	247	F7	1111 0111	÷
222	DE	1101 1110	Þ	248	F8	1111 1000	Ø
223	DF	1101 1111	ß	249	F9	1111 1001	Ù
224	E0	1110 0000	À	250	FA	1111 1010	Ú
225	E1	1110 0001	Á	251	FB	1111 1011	Û
226	E2	1110 0010	Â	252	FC	1111 1100	Ü
227	E3	1110 0011	Ã	253	FD	1111 1101	Ý
228	E4	1110 0100	Ä	254	FE	1111 1110	Þ
229	E5	1110 0101	Å	255	FF	1111 1111	Ÿ
230	E6	1110 0110	Æ				

UPT Perpustakaan Universitas Jember

LAMPIRAN B. Data Penelitian (*Plainimage*, *Public Key*, dan *Private Key*)

No.	*Plainimage*	*Public Key* 1	*Public Key* 2	*Private Key* 1 (*Sender*)	*Private Key* 2 (*Receiver*)
1.	 (*trucks*.png)	 (*building*.png)	 (*holdingCup*.png)	5	8
2.	 (*city*.png)	 (*image*05.png)	 (*onion*.png)	3	4
3.	 (*text*.png)	 (*paper*.png)	 (*peppers*.png)	2	6
4.	 (*textgray*.png)	 (*image*0150.png)	 (*left*05.png)	5	4

UPT Perpustakaan Universitas Jember

No.	*Plainimage*	*Public Key* 1	*Public Key* 2	*Private Key* 1 (*Sender*)	*Private Key* 2 (*Receiver*)
5.	 (*cameraman*.png)	 (*coins*.png)	 (*image*1.png)	6	7

UPT Perpustakaan Universitas Jember

LAMPIRAN C. Matriks Derajat Keabuan *Plainimage* (*cameraman*.png)

Ukuran matriks 200x200 *pixel*

Pixel	1	2	3	4	5	6	7	8	9	10	11	12	...	200
1	**158**	**158**	**157**	157	158	159	157	158	157	159	160	159	…	152
2	**158**	**157**	**157**	157	158	158	158	159	156	159	159	158	...	153
3	**158**	**157**	**157**	157	158	158	158	159	156	159	159	158	...	153
4	156	153	158	159	156	156	156	157	154	155	158	160	...	152
5	155	155	156	156	155	154	157	157	153	156	157	158	...	152
6	157	157	157	155	156	157	160	156	157	158	158	160	...	148
7	158	157	156	157	161	159	159	159	158	161	157	160	...	148
8	155	156	158	159	160	159	158	161	160	159	160	158	...	154
9	153	155	156	154	156	156	157	157	159	157	158	160	...	152
10	154	153	155	155	159	157	158	157	160	159	157	159	...	151
11	152	156	157	157	159	157	157	160	158	159	161	159	...	149
12	156	158	159	158	156	156	159	160	161	160	161	162	...	152
13	156	157	158	158	155	158	159	156	158	158	160	159	...	154
14	152	155	155	156	157	156	157	157	155	158	157	158	...	152
15	153	154	153	156	154	154	155	155	155	157	155	159	...	154
16	154	154	156	157	155	156	159	157	159	161	159	161	...	152
17	155	155	155	158	158	160	159	159	161	161	159	161	...	152
18	151	152	155	155	157	156	157	157	157	158	160	159	...	156
19	151	153	154	154	154	154	156	157	158	156	159	159	...	157
20	153	153	154	154	154	156	158	156	158	159	159	159	...	154
21	159	156	155	156	156	156	156	158	159	162	161	161	...	152
22	158	155	156	159	156	159	159	160	160	163	159	161	...	153
23	160	159	158	158	157	158	159	160	162	161	160	164	...	154
24	156	158	157	158	156	160	158	159	160	159	163	164	...	153
25	153	154	156	157	155	158	160	159	160	159	161	162	...	147
26	154	157	155	155	156	157	157	160	159	162	161	163	...	152
27	157	159	157	159	160	157	160	164	162	164	165	165	...	153
28	157	157	157	160	158	157	158	160	162	161	161	161	...	152
29	156	157	156	155	155	159	159	159	161	159	159	161	...	152
30	153	153	151	156	157	157	157	160	159	159	159	158	...	149
31	158	157	155	158	157	158	158	160	159	160	162	161	...	148
32	157	158	159	158	159	161	162	163	164	162	160	163	...	150
33	156	157	160	160	160	160	162	163	164	163	164	164		151
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
200	122	125	156	118	167	154	146	164	135	117	138	145	...	116

UPT Perpustakaan Universitas Jember

LAMPIRAN D. Matriks Derajat Keabuan *Public Key* (*cameraman*.png)

D1. Matriks Derajat Keabuan *Public Key* 1 (*coins*.png)

Ukuran matriks 200x200 *pixel*

Pixel	1	2	3	4	5	6	7	8	9	10	11	12	...	200
1	**49**	**49**	**49**	48	49	50	50	49	49	49	50	50	…	55
2	**47**	**48**	**49**	49	49	49	49	49	49	49	48	50	...	57
3	**49**	**49**	**48**	49	49	50	50	50	50	49	50	49	...	57
4	48	47	49	49	49	49	48	49	49	48	49	50	...	55
5	48	49	50	51	50	51	49	48	49	49	50	49	...	58
6	49	50	50	50	50	50	49	49	49	50	49	50	...	59
7	51	48	49	49	49	49	50	49	50	50	50	52	...	56
8	51	50	49	50	50	49	49	50	49	50	49	49	...	57
9	49	48	50	50	50	47	48	49	50	50	50	49	...	59
10	47	48	49	49	49	48	49	49	49	50	51	51	...	57
11	50	50	49	48	48	49	50	49	50	52	50	49	...	58
12	50	49	50	49	49	49	50	49	49	50	50	51	...	59
13	50	50	51	50	50	51	49	50	51	49	50	51	...	58
14	49	50	50	50	50	50	49	49	50	50	50	50	...	59
15	49	49	49	48	48	49	49	50	49	48	49	51	...	59
16	51	51	50	50	50	50	50	51	51	50	50	50	...	60
17	50	50	52	50	50	48	49	49	50	50	50	50	...	60
18	51	51	53	51	51	49	49	49	50	50	51	51	...	61
19	51	51	52	50	50	50	51	51	50	49	50	50	...	60
20	49	50	50	51	51	51	50	51	51	50	50	50	...	60
21	50	50	50	50	50	50	50	49	50	50	51	51	...	62
22	51	51	50	50	51	51	51	50	50	51	51	50	...	62
23	50	51	51	50	51	51	51	51	51	51	51	52	...	61
24	51	51	51	51	51	52	50	50	51	51	50	51	...	60
25	51	50	50	51	51	50	51	51	51	50	50	52	...	61
26	51	51	52	51	51	51	50	50	50	50	50	50	...	62
27	51	52	52	50	51	51	52	52	52	51	52	51	...	62
28	52	51	51	51	51	52	53	50	52	51	52	52	...	60
29	51	52	52	52	52	51	52	51	51	52	51	53	...	61
30	52	52	52	52	51	53	53	52	52	52	52	53	...	64
31	51	51	51	52	52	52	52	53	52	52	53	51	...	62
32	52	52	52	54	53	52	51	53	53	52	52	53	...	60
33	51	51	52	52	52	52	53	53	53	54	54	55		60
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
200	69	69	70	69	70	71	70	70	70	71	71	71	...	71

UPT Perpustakaan Universitas Jember

D2. Matriks Derajat Keabuan *Public Key* 2 (*image*1.png)
Ukuran matriks 200x200 *pixel*

Pixel	1	2	3	4	5	6	7	8	9	10	11	12	...	200
1	**24**	**23**	**18**	17	17	17	14	12	11	9	9	8	…	100
2	**33**	**23**	**18**	12	12	13	11	12	10	9	9	8	...	99
3	**46**	**22**	**13**	12	7	8	7	10	8	11	8	8	...	99
4	44	31	12	11	6	5	6	10	8	9	10	7	...	103
5	23	39	20	6	7	6	6	7	8	7	7	8	...	105
6	11	9	18	8	6	7	6	6	7	6	7	7	...	104
7	18	6	6	9	7	5	6	5	6	6	6	6	...	107
8	22	7	5	5	5	6	7	7	7	8	6	6	...	104
9	23	9	5	5	5	6	6	7	8	8	8	8	...	108
10	23	12	6	5	5	7	6	6	6	8	5	3	...	111
11	26	19	8	5	5	7	6	3	5	5	5	4	...	112
12	28	25	8	4	5	7	6	6	5	5	5	5	...	112
13	27	27	22	7	6	6	5	5	4	7	5	5	...	113
14	30	29	29	11	4	5	4	5	4	6	4	4	...	115
15	33	32	33	26	8	4	6	7	7	5	5	4	...	111
16	35	36	37	32	21	9	6	7	6	4	5	6	...	115
17	35	38	37	33	28	15	7	8	6	5	6	6	...	119
18	34	39	37	32	27	21	8	6	6	7	6	5	...	120
19	38	37	35	33	29	25	12	4	3	6	6	3	...	122
20	38	32	34	34	24	18	9	4	3	4	6	3	...	123
21	33	32	22	16	12	11	7	5	7	7	7	7	...	122
22	30	15	6	6	7	4	4	6	6	7	7	6	...	123
23	21	7	6	7	6	6	6	8	6	7	6	6	...	124
24	16	5	6	7	5	6	7	7	7	5	6	6	...	126
25	7	6	5	5	6	9	8	7	5	6	7	7	...	125
26	6	8	5	7	8	10	9	7	5	8	7	7	...	125
27	6	8	6	7	7	7	8	6	6	6	7	6	...	125
28	6	7	7	5	7	6	6	6	7	6	8	6	...	125
29	6	6	7	7	8	6	8	7	7	7	8	7	...	128
30	5	8	7	9	8	6	7	7	7	8	8	8	...	127
31	6	8	7	9	7	6	7	8	7	8	8	8	...	125
32	7	8	9	9	7	7	6	8	8	6	8	8	...	128
33	7	8	8	8	7	8	6	8	7	8	9	7		129
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
200	2	3	2	4	4	2	2	2	2	2	3	1	...	4

UPT Perpustakaan Universitas Jember

LAMPIRAN E. Matriks Derajat Keabuan *Shared Key*

E1. Matriks Derajat Keabuan *Shared Key* Pengirim Pesan Tersandi

Ukuran matriks 200x200 *pixel*

Pixel	1	2	3	4	5	6	7	8	9	10	11	12	...	200
1	28	52	4	44	44	44	38	17	36	10	10	6	…	77
2	42	52	4	17	17	46	36	17	49	10	10	6	...	65
3	42	25	46	17	42	6	42	49	6	36	6	6	...	65
4	10	2	17	36	16	43	16	49	6	10	49	42	...	76
5	52	38	9	16	42	16	16	42	6	42	42	6	...	33
6	36	10	4	6	16	42	16	16	42	16	42	42	...	83
7	4	16	16	10	42	43	16	43	16	16	16	16	...	92
8	25	42	43	43	43	16	42	42	42	6	16	16	...	41
9	52	10	43	43	43	32	16	42	6	6	6	6	...	93
10	36	17	16	43	43	42	16	16	16	6	43	40	...	83
11	29	7	6	43	43	42	16	40	43	43	43	15	...	52
12	47	47	6	15	43	42	16	16	43	43	43	43	...	87
13	29	29	25	42	16	16	43	43	15	42	43	43	...	52
14	52	28	28	36	15	43	15	43	15	16	15	15	...	19
15	9	37	9	29	6	15	16	42	42	43	43	15	...	87
16	4	44	13	37	37	10	16	42	16	15	43	16	...	94
17	4	24	13	9	47	24	42	6	16	43	16	16	...	94
18	7	38	13	37	29	37	6	16	16	42	16	43	...	8
19	24	13	4	9	28	47	17	15	40	16	16	40	...	94
20	24	37	7	7	28	4	10	15	40	15	16	40	...	94
21	9	37	25	13	17	36	42	43	42	42	42	42	...	61
22	52	24	16	16	42	15	15	16	16	42	42	16	...	61
23	37	42	16	42	16	16	16	6	16	42	16	16	...	8
24	13	43	16	42	43	16	42	42	42	43	16	16	...	94
25	42	16	43	43	16	10	6	42	43	16	42	42	...	8
26	16	6	43	42	6	49	10	42	43	6	42	42	...	61
27	16	6	16	42	42	42	6	16	16	16	42	16	...	61
28	16	42	42	43	42	16	16	16	42	16	6	16	...	94
29	16	16	42	42	6	16	6	42	42	42	6	42	...	61
30	43	6	42	10	6	16	42	42	42	6	6	6	...	2
31	16	6	42	10	42	16	42	6	42	6	6	6	...	61
32	42	6	10	28	42	42	16	6	6	16	6	6	...	99
33	42	6	6	6	42	6	16	6	42	7	28	3		99
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
200	64	19	64	49	49	64	64	64	64	64	19	1	...	49

UPT Perpustakaan Universitas Jember

E2. Matriks Derajat Keabuan *Shared Key* Penerima Pesan Tersandi
Ukuran matriks 200x200 *pixel*

Pixel	1	2	3	4	5	6	7	8	9	10	11	12	...	200
1	36	30	19	6	6	6	2	45	25	37	37	48	…	94
2	23	30	19	45	45	15	25	45	13	37	37	48	...	69
3	24	20	15	45	29	48	29	13	48	25	48	48	...	69
4	16	15	45	25	43	3	43	13	48	37	13	29	...	60
5	30	51	21	43	29	43	43	29	48	29	29	48	...	95
6	25	37	19	48	43	29	43	43	29	43	29	29	...	82
7	19	43	43	37	29	3	43	3	43	43	43	43	...	16
8	20	29	3	3	3	43	29	29	29	48	43	43	...	90
9	30	37	3	3	3	4	43	29	48	48	48	48	...	37
10	29	45	43	3	3	29	43	43	43	48	3	14	...	98
11	12	27	48	3	3	29	43	14	3	3	3	7	...	78
12	44	9	48	7	3	29	43	43	3	3	3	3	...	48
13	41	41	20	29	43	43	3	3	7	29	3	3	...	78
14	23	17	17	25	7	3	7	3	7	43	7	7	...	105
15	32	18	32	12	48	7	43	29	29	3	3	7	...	48
16	34	47	4	18	35	37	43	29	43	7	3	43	...	52
17	34	11	4	32	44	42	29	48	43	3	43	43	...	52
18	26	51	4	18	41	35	48	43	43	29	43	3	...	107
19	11	4	34	32	17	9	45	7	14	43	43	14	...	52
20	11	18	26	26	36	19	37	7	14	7	43	14	...	52
21	32	18	20	49	45	25	29	3	29	29	29	29	...	99
22	23	42	43	43	29	7	7	43	43	29	29	43	...	99
23	35	29	43	29	43	43	43	48	43	29	43	43	...	107
24	49	3	43	29	3	43	29	29	29	3	43	43	...	52
25	29	43	3	3	43	37	48	29	3	43	29	29	...	107
26	43	48	3	29	48	13	37	29	3	48	29	29	...	99
27	43	48	43	29	29	29	48	43	43	43	29	43	...	99
28	43	29	29	3	29	43	43	43	29	43	48	43	...	52
29	43	43	29	29	48	43	48	29	29	29	48	29	...	53
30	3	48	29	37	48	43	29	29	29	48	48	48	...	1
31	43	48	29	37	29	43	29	48	29	48	48	48	...	99
32	29	48	37	16	29	29	43	48	48	43	48	48	...	45
33	29	48	48	48	29	48	43	48	29	56	16	21		45
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
200	57	57	57	54	54	57	57	57	57	57	57	1	...	54

UPT Perpustakaan Universitas Jember

LAMPIRAN F. Matriks Derajat Keabuan *Secret Key* (Data ke lima)

Ukuran matriks 200x200 *pixel*

Pixel	1	2	3	4	5	6	7	8	9	10	11	12	...	200
1	**44**	**52**	**7**	16	16	16	11	6	47	13	13	44	…	85
2	**36**	**52**	**7**	6	6	24	47	6	46	13	13	36	...	17
3	**28**	**9**	**24**	6	28	43	28	46	43	47	43	28	...	17
4	13	34	6	47	49	40	49	46	43	13	46	13	...	81
5	52	11	37	49	28	49	49	28	43	28	28	52	...	42
6	47	13	7	43	49	28	49	49	28	49	28	47	...	9
7	7	49	49	13	28	40	49	40	49	49	49	7	...	44
8	9	28	40	40	40	49	28	28	28	43	49	9	...	81
9	52	13	40	40	40	7	49	28	43	43	43	52	...	43
10	16	6	49	40	40	28	49	49	49	43	40	16	...	112
11	17	29	43	40	40	28	49	38	40	40	40	17	...	44
12	10	10	43	42	40	28	49	49	40	40	40	10	...	95
13	17	17	9	28	49	49	40	40	42	28	40	17	...	44
14	52	44	44	47	42	40	42	40	42	49	42	52	...	19
15	37	4	37	17	43	42	49	28	28	40	40	37	...	95
16	7	16	15	4	4	13	49	28	49	42	40	7	...	107
17	7	36	15	37	10	36	28	43	49	40	49	7	...	107
18	29	11	15	4	17	4	43	49	49	28	49	29	...	1
19	36	15	7	37	44	10	6	42	38	49	49	36	...	107
20	36	4	29	29	44	7	13	42	38	42	49	36	...	107
21	37	4	9	15	6	47	28	40	28	28	28	37	...	107
22	52	36	49	49	28	42	42	49	49	28	28	52	...	107
23	4	28	49	28	49	49	49	43	49	28	49	4	...	1
24	15	40	49	28	40	49	28	28	28	40	49	15	...	107
25	28	49	40	40	49	13	43	28	40	49	28	28	...	1
26	49	43	40	28	43	46	13	28	40	43	28	49	...	107
27	49	43	49	28	28	28	43	49	49	49	28	49	...	107
28	49	28	28	40	28	49	49	49	28	49	43	49	...	107
29	49	49	28	28	43	49	43	28	28	28	43	49	...	53
30	40	43	28	13	43	49	28	28	28	43	43	40	...	1
31	49	43	28	13	28	49	28	43	28	43	43	49	...	107
32	28	43	13	35	28	28	49	43	43	49	43	28	...	112
33	28	43	43	43	28	43	49	43	28	21	35	28		112
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
200	57	57	57	54	54	57	57	57	57	57	57	1	...	54

UPT Perpustakaan Universitas Jember

LAMPIRAN G. Matriks Derajat Keabuan *Cipherimage* (Data ke lima)

Ukuran matriks 200x200 *pixel*

Pixel	1	2	3	4	5	6	7	8	9	10	11	12	...	200
1	**188**	**17**	**62**	226	252	129	24	12	24	50	216	245	…	144
2	**188**	**47**	**62**	150	12	252	211	24	33	50	50	18	...	230
3	**145**	**168**	**100**	150	145	18	145	152	40	48	245	18	...	230
4	184	150	12	48	158	16	158	180	83	119	62	240	...	61
5	233	48	158	158	155	120	163	169	217	144	169	18	...	225
6	24	168	62	176	158	169	254	158	169	61	145	240	...	147
7	101	163	158	168	132	154	64	154	61	43	163	254	...	255
8	178	144	60	154	220	64	145	132	240	245	254	61	...	94
9	74	119	16	121	16	174	163	169	245	152	18	45	...	77
10	185	215	37	27	154	169	61	163	254	245	193	152	...	33
11	159	17	152	193	154	169	163	167	60	154	161	24	...	79
12	36	62	245	190	16	144	64	254	161	220	161	127	...	70
13	222	99	16	145	37	61	154	16	190	145	220	154	...	57
14	55	91	91	104	113	16	113	193	89	61	113	190	...	139
15	75	121	75	222	83	59	37	155	155	193	27	24	...	229
16	224	185	168	34	169	184	64	169	64	17	154	43	...	203
17	53	105	53	81	62	23	218	245	43	161	64	43	...	203
18	197	205	53	169	99	247	152	163	163	145	254	154	...	158
19	117	217	224	16	57	123	71	113	204	158	64	152	...	62
20	231	103	145	145	57	174	17	164	204	24	64	152	...	101
21	0	247	178	168	71	104	144	60	218	255	132	132	...	203
22	17	105	158	64	144	24	24	254	254	24	218	43	...	187
23	151	218	61	145	163	61	64	45	194	132	254	127	...	81
24	168	60	163	145	16	254	145	218	240	154	32	127	...	187
25	12	120	16	193	37	17	45	218	220	64	132	255	...	156
26	120	152	27	155	40	180	168	240	154	124	132	24	...	203
27	163	245	163	218	240	169	45	127	194	127	33	142	...	187
28	163	169	169	220	145	163	61	254	255	43	248	43	...	203
29	158	163	144	155	176	64	245	218	132	218	245	132	...	118
30	164	217	68	184	152	163	169	240	218	245	245	18	...	32
31	61	152	155	17	169	61	145	45	218	45	124	248	...	197
32	169	18	50	32	218	132	194	55	72	194	45	55	...	240
33	144	152	45	45	240	45	194	55	113	32	14	210		33
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
200	162	27	248	143	203	120	130	221	89	153	209	69	...	114

UPT Perpustakaan Universitas Jember

LAMPIRAN H. Matriks Derajat Keabuan Hasil Dekripsi *Cipherimage* (Data ke lima)

Ukuran matriks 200x200 *pixel*

Pixel	1	2	3	4	5	6	7	8	9	10	11	12	...	200
1	**158**	**158**	**157**	157	158	159	157	158	157	159	160	159	…	152
2	**158**	**157**	**157**	157	158	158	158	159	156	159	159	158	...	153
3	**158**	**157**	**157**	157	158	158	158	159	156	159	159	158	...	153
4	156	153	158	159	156	156	156	157	154	155	158	160	...	152
5	155	155	156	156	155	154	157	157	153	156	157	158	...	152
6	157	157	157	155	156	157	160	156	157	158	158	160	...	148
7	158	157	156	157	161	159	159	159	158	161	157	160	...	148
8	155	156	158	159	160	159	158	161	160	159	160	158	...	154
9	153	155	156	154	156	156	157	157	159	157	158	160	...	152
10	154	153	155	155	159	157	158	157	160	159	157	159	...	151
11	152	156	157	157	159	157	157	160	158	159	161	159	...	149
12	156	158	159	158	156	156	159	160	161	160	161	162	...	152
13	156	157	158	158	155	158	159	156	158	158	160	159	...	154
14	152	155	155	156	157	156	157	157	155	158	157	158	...	152
15	153	154	153	156	154	154	155	155	155	157	155	159	...	154
16	154	154	156	157	155	156	159	157	159	161	159	161	...	152
17	155	155	155	158	158	160	159	159	161	161	159	161	...	152
18	151	152	155	155	157	156	157	157	157	158	160	159	...	156
19	151	153	154	154	154	154	156	157	158	156	159	159	...	157
20	153	153	154	154	154	156	158	156	158	159	159	159	...	154
21	159	156	155	156	156	156	156	158	159	162	161	161	...	152
22	158	155	156	159	156	159	159	160	160	163	159	161	...	153
23	160	159	158	158	157	158	159	160	162	161	160	164	...	154
24	156	158	157	158	156	160	158	159	160	159	163	164	...	153
25	153	154	156	157	155	158	160	159	160	159	161	162	...	147
26	154	157	155	155	156	157	157	160	159	162	161	163	...	152
27	157	159	157	159	160	157	160	164	162	164	165	165	...	153
28	157	157	157	160	158	157	158	160	162	161	161	161	...	152
29	156	157	156	155	155	159	159	159	161	159	159	161	...	152
30	153	153	151	156	157	157	157	160	159	159	159	158	...	149
31	158	157	155	158	157	158	158	160	159	160	162	161	...	148
32	157	158	159	158	159	161	162	163	164	162	160	163	...	150
33	156	157	160	160	160	160	162	163	164	163	164	164		151
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
200	122	125	156	118	167	154	146	164	135	117	138	145	...	116

UPT Perpustakaan Universitas Jember

LAMPIRAN I. Skrip Program Analisis Differensial

NPCR.m

```
clear all;
clc;

a=imread('plainimage5.png'); %Plainimage
b=imread('cipherimage5.png'); %Cipherimage
[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
imshow(mat2gray(beda,[0 1]));
npcr=sum(sum(sum(beda)))/(c*d*e)*100
```

UPT Perpustakaan Universitas Jember

LAMPIRAN J. Skrip Program Pembentukan *Shared Key*

Diffie_Hellman_SharedKey.m

```
clear all;
clc;

Pubkey1=imread('publickey1.png');
Pubkey2=imread('publickey2.png');
pk==str2num(input('Masukkan private key : ','s'));
[br1, kl1, kn1]=size(Pubkey1);
for kanal=1:kn1
array1(:,:,kanal)=uint64(reshape(transpose(Pubkey1(:,:,kanal)),1,br1*kl1));
array2(:,:,kanal)=uint64(reshape(transpose(Pubkey2(:,:,kanal)),1,br1*kl1));
    %%
    %ALGORITMA DIFFIE-HELLMAN
    %=========================================================================
    %Pembangkitan dan Penentuan Nilai P dan G
    %-------------------------------------------------------------------------
    for i=1:br1*kl1
        %Pembangkitan Nilai P dan G
        %%-----------------------------------------------------------------
        %Perlakuan a
        %Jika Nilai Pixel A & Pixel B = 0
        if array1(:,i,kanal)==0 && array2(:,i,kanal)==0
           array3(:,i,kanal)=array1(:,i,kanal)+2;
           array4(:,i,kanal)=array2(:,i,kanal)+2;
        %Jika Nilai Pixel A = 0
        elseif array1(:,i,kanal)==0
           array3(:,i,kanal)=array1(:,i,kanal)+2;
           array4(:,i,kanal)=array2(:,i,kanal);
        %Jika Nilai Pixel B = 0
        elseif array2(:,i,kanal)==0
           array4(:,i,kanal)=array2(:,i,kanal)+2;
           array3(:,i,kanal)=array1(:,i,kanal);
        else
           array3(:,i,kanal)=array1(:,i,kanal);
           array4(:,i,kanal)=array2(:,i,kanal);
        end
        %Perlakuan b
        %Jika Nilai Pixel A = Pixel B
        if array3(:,i,kanal)==array4(:,i,kanal)
           array3(:,i,kanal)=array3(:,i,kanal)+2;
        end
        %Penentuan Nilai P dan G
        %%-----------------------------------------------------------------
        %Perlakuan c
        %Jika Nilai pixel A < Nilai pixel B; p = Pixel B; g = Pixel A
        if array3(:,i,kanal)<array4(:,i,kanal)
           p(:,i,kanal)=array4(:,i,kanal);
           g(:,i,kanal)=array3(:,i,kanal);
        %Jika Nilai pixel A > Nilai pixel B; p = Pixel A; g = Pixel B
        else
           p(:,i,kanal)=array3(:,i,kanal);
           g(:,i,kanal)=array4(:,i,kanal);
        end
        %Perlakuan d
```

UPT Perpustakaan Universitas Jember


```
        %Nilai p harus Prima
        if isprime(p(:,i,kanal))==0
           p(:,i,kanal)=nextprime(sym(p(:,i,kanal)));
        end
        %Operasi Modulo pada Difie-hellman
        %%-------------------------------------------------------------
        Sk(:,i,kanal)=mod(g(:,i,kanal)^pk,p(:,i,kanal));

        if Sk(:,i,kanal)>255
           Sk(:,i,kanal)=mod(Sk(:,i,kanal),256);
        end
    end
    Seckeynew(:,:,kanal)=uint8(transpose(reshape(Sk(:,:,kanal),br1,kl1)));
end
abc=imshow(mat2gray(Seckeynew));
imsave(abc);
```

UPT Perpustakaan Universitas Jember

LAMPIRAN K. Skrip Program Pembentukan *Secret Key*

Diffie_Hellman_SecretKey.m

```
clear all;
clc;

Pubkey1=imread('publickey1.png');
Pubkey2=imread('publickey2.png');
Shakey=imread('shared key.png');
pk=str2num(input('Masukkan private key : ','s'));
y=str2num(input('Masukkan y : ','s'));
[br1, kl1, kn1]=size(Pubkey1);
for kanal=1:kn1
%Mengubah nilai pixel matrik berbentuk array
array1(:,:,kanal)=uint64(reshape(transpose(Pubkey1(:,:,kanal)),1,br1*kl1));
array2(:,:,kanal)=uint64(reshape(transpose(Pubkey2(:,:,kanal)),1,br1*kl1));
    %Pembangkitan dan Penentuan Nilai P dan G
    %Pembangkitan
    %-------------------------------------------------------------------
    for i=1:br1*kl1
        %Perlakuan a
        %Jika Nilai Pixel A & Pixel B = 0
        if array1(:,i,kanal)==0 && array2(:,i,kanal)==0
           array3(:,i,kanal)=array1(:,i,kanal)+2;
           array4(:,i,kanal)=array2(:,i,kanal)+2;
        %Jika Nilai Pixel A = 0
        elseif array1(:,i,kanal)==0
           array3(:,i,kanal)=array1(:,i,kanal)+2;
           array4(:,i,kanal)=array2(:,i,kanal);
        %Jika Nilai Pixel B = 0
        elseif array2(:,i,kanal)==0
           array4(:,i,kanal)=array2(:,i,kanal)+2;
           array3(:,i,kanal)=array1(:,i,kanal);
        else
           array3(:,i,kanal)=array1(:,i,kanal);
           array4(:,i,kanal)=array2(:,i,kanal);
        end
        %Perlakuan b
        %Jika Nilai Pixel A = Pixel B
        if array3(:,i,kanal)==array4(:,i,kanal)
           array3(:,i,kanal)=array3(:,i,kanal)+2;
        end
    %-------------------------------------------------------------------
    %Penentuan Nilai P dan G
    %-------------------------------------------------------------------
        %Perlakuan c
        %Jika Nilai pixel A < Nilai pixel B; p = Pixel B; g = Pixel A
        if array3(:,i,kanal)<array4(:,i,kanal)
           p(:,i,kanal)=array4(:,i,kanal);
           g(:,i,kanal)=array3(:,i,kanal);
        %Jika Nilai pixel A > Nilai pixel B; p = Pixel A; g = Pixel B
        else
           p(:,i,kanal)=array3(:,i,kanal);
           g(:,i,kanal)=array4(:,i,kanal);
        end
```

UPT Perpustakaan Universitas Jember


```
        %Perlakuan d
        %Nilai p harus Prima
        if isprime(p(:,i,kanal))==0
           p(:,i,kanal)=nextprime(sym(p(:,i,kanal)));
           g(:,i,kanal)=g(:,i,kanal);
        end
    end
    %-----------------------------------------------------------------
    %Operasi Modulo pada Difie-hellman
    %=================================================================
   Kunc(:,i,kanal)=mod(Sk(:,i,kanal)^pk,p(:,i,kanal));%Kunci Enk/Dek
        %Jika Nilai pixel melebihi ukuran 8 bit (255)
        if Kunc(:,i,kanal)>255
           Kunc(:,i,kanal)=mod(Kunc(:,i,kanal),256);
        end

    end
    %Pembentukan citra
    Seckeynew(:,:,kanal)=uint8(transpose(reshape(Kunc(:,:,kanal),br1,kl1)));
end
abc=imshow(mat2gray(Seckeynew));
imsave(abc);
```

UPT Perpustakaan Universitas Jember

LAMPIRAN L. Skrip Program Enkripsi Algoritma S-DES

S_DES_Enkripsi.m

```
clear all;
clc;

Image=imread('plainimage1.png);
Seckeynew=imread('secretkey1.png);
[br1, kl1, kn1]=size(Image);
% Fungsi Pembangkitan Kunci S-DES
P10 = [3 5 2 7 4 10 1 9 8 6]; % Tabel permutasi P10
P8 = [6 3 7 4 8 5 10 9]; %Tabel permutasi P8
%-------------------------------------------------------------------------
% Fungsi Enkripsi/Dekripsi S-DES
IP = [2 6 3 1 4 8 5 7]; %Tabel initial permutasi
EP = [4 1 2 3 2 3 4 1]; %Tabel ekspansi/permutasi
s0 = [1 0 3 2; 3 2 1 0; 0 2 1 3; 3 1 3 2]; %Tabel s-boxes(0)
s1 = [0 1 2 3; 2 0 1 3; 3 0 1 0; 2 1 0 3]; %Tabel s-boxes(1)
P4 = [2 4 3 1]; %Tabel permutasi P4
IPI = [4 1 3 5 7 2 8 6]; %Tabel initial permutasi invers
%-------------------------------------------------------------------------
for kanal=1:kn1
    Key(:,:,kanal)=reshape(transpose(Seckeynew(:,:,kanal)),1,br1*kl1);
    Gambar(:,:,kanal)=reshape(transpose(Image(:,:,kanal)),1,br1*kl1);
    for i=1:br1*kl1
        %%
        %PEMBANGKITAN KUNCI S-DES
        %=======================================================================
        Kunci=de2bi(Key(:,i,kanal),10,'left-msb'); %Konversi biner
        %Operasi Permutasi P10
        Kunci=Kunci(P10);
        A=Kunci(1:5); %5 bit awal
        B=Kunci(6:10); %5 bit akhir
        %Operasi LS-1
        LS_1A=[A(2:5) A(1)]; %LS-1 5 bit awal
        LS_1B=[B(2:5) B(1)]; %LS-1 5 bit akhir
        K1=[LS_1A,LS_1B];
        %Operasi Permutasi P8
        K1=K1(P8); %Sub-kunci K1
        %Operasi LS-2
        LS_2A=[LS_1A(3:5) LS_1A(1:2)]; %LS-2 5 bit awal
        LS_2B=[LS_1B(3:5) LS_1B(1:2)]; %LS-2 5 bit akhir
        K2=[LS_2A,LS_2B];
        %Operasi Permutasi P8
        K2 = K2(P8); %Sub-kunci K2
        K=[K1,K2];

        %ENKRIPSI S-DES
        %=======================================================================
        Citra=de2bi(Gambar(:,i,kanal),8,'left-msb'); %Konversi biner
        %Fungsi Permutasi Awal
        Citra=Citra(IP);
        %Fungsi FK
        L=Citra(1:4);
        R=Citra(5:8);
        z=0;
```

UPT Perpustakaan Universitas Jember


```
            for fk=1:2 %perulangan fungsi FK
                Rnew=R(EP); %Operasi enkspansi/permutasi
                Rnew=xor(Rnew,K(1+z*2:8+z*2)); %Operasi XOR dengan K1
                 %Operasi S-box (0)
                LRnew=s0(bi2de([Rnew(4) Rnew(1)])+1,bi2de([Rnew(3) Rnew(2)])+1);
                 %Operasi S-box (1)
                RRnew=s1(bi2de([Rnew(8) Rnew(5)])+1,bi2de([Rnew(7) Rnew(6)])+1);
                 %Konversi biner
                w=[floor(LRnew/2) rem(LRnew,2) floor(RRnew/2) rem(RRnew,2)];
                FRSK=w(P4); %Operasi permutasi P4
                R1(1+z:4+z) = xor(L,FRSK); %Operasi XOR (L XOR FRSK)
                %Fungsi Permutasi Sedrhana (SW)
                L=R;
                R=R1;
                z=z+4; %Untuk fungsi FK ke-2
            end
            R2L2=[R1(5:8) R1(1:4)];
            %Fungsi Permutasi Akhir
            PA=R2L2(IPI); %Operasi IP invers
            PA=bi2de(PA,'left-msb'); %Konversi desimal

            abc(:,i,kanal)=PA;
    end
end
Cipherimage(:,:,kanal)=transpose(reshape(abc(:,:,kanal),br1,kl1));
imshow(mat2gray(Cipherimage));
imsave
```

UPT Perpustakaan Universitas Jember

LAMPIRAN M. Skrip Program Dekripsi Algoritma S-DES

S_DES_Dekripsi.m

```
clear all;
clc;

Image=imread('cipherimage1.png');
Seckeynew=imread('secretkey1.png');
[br1, kl1, kn1]=size(Image);
% Fungsi Pembangkitan Kunci S-DES
P10 = [3 5 2 7 4 10 1 9 8 6]; % Tabel permutasi P10
P8 = [6 3 7 4 8 5 10 9]; %Tabel permutasi P8
%-------------------------------------------------------------------------
% Fungsi Enkripsi/Dekripsi S-DES
IP = [2 6 3 1 4 8 5 7]; %Tabel initial permutasi
EP = [4 1 2 3 2 3 4 1]; %Tabel ekspansi/permutasi
s0 = [1 0 3 2; 3 2 1 0; 0 2 1 3; 3 1 3 2]; %Tabel s-boxes(0)
s1 = [0 1 2 3; 2 0 1 3; 3 0 1 0; 2 1 0 3]; %Tabel s-boxes(1)
P4 = [2 4 3 1]; %Tabel permutasi P4
IPI = [4 1 3 5 7 2 8 6]; %Tabel initial permutasi invers
%-------------------------------------------------------------------------
for kanal=1:kn1
    Key(:,:,kanal)=reshape(transpose(Seckeynew(:,:,kanal)),1,br1*kl1);
    Gambar(:,:,kanal)=reshape(transpose(Image(:,:,kanal)),1,br1*kl1);
    for i=1:br1*kl1
        %%
        %PEMBANGKITAN KUNCI S-DES
        %=====================================================================
        Kunci=de2bi(Key(:,i,kanal),10,'left-msb'); %Konversi biner
        %Operasi Permutasi P10
        Kunci=Kunci(P10);
        A=Kunci(1:5); %5 bit awal
        B=Kunci(6:10); %5 bit akhir
        %Operasi LS-1
        LS_1A=[A(2:5) A(1)]; %LS-1 5 bit awal
        LS_1B=[B(2:5) B(1)]; %LS-1 5 bit akhir
        K1=[LS_1A,LS_1B];
        %Operasi Permutasi P8
        K1=K1(P8); %Sub-kunci K1
        %Operasi LS-2
        LS_2A=[LS_1A(3:5) LS_1A(1:2)]; %LS-2 5 bit awal
        LS_2B=[LS_1B(3:5) LS_1B(1:2)]; %LS-2 5 bit akhir
        K2=[LS_2A,LS_2B];
        %Operasi Permutasi P8
        K2 = K2(P8); %Sub-kunci K2
        K=[K2,K1];

        %DEKRIPSI S-DES
        %=====================================================================
        Citra=de2bi(Gambar(:,i,kanal),8,'left-msb'); %Konversi biner
        %Fungsi Permutasi Awal
        Citra=Citra(IP);
        %Fungsi FK
        L=Citra(1:4);
        R=Citra(5:8);
        z=0;
```

UPT Perpustakaan Universitas Jember


```
        for fk=1:2 %perulangan fungsi FK
            Rnew=R(EP); %Operasi enkspansi/permutasi
            Rnew=xor(Rnew,K(1+z*2:8+z*2)); %Operasi XOR dengan K1
             %Operasi S-box (0)
            LRnew=s0(bi2de([Rnew(4) Rnew(1)])+1,bi2de([Rnew(3) Rnew(2)])+1);
             %Operasi S-box (1)
            RRnew=s1(bi2de([Rnew(8) Rnew(5)])+1,bi2de([Rnew(7) Rnew(6)])+1);
             %Konversi biner
            w=[floor(LRnew/2) rem(LRnew,2) floor(RRnew/2) rem(RRnew,2)];
            FRSK=w(P4); %Operasi permutasi P4
            R1(1+z:4+z) = xor(L,FRSK); %Operasi XOR (L XOR FRSK)
            %Fungsi Permutasi Sedrhana (SW)
            L=R;
            R=R1;
            z=z+4; %Untuk fungsi FK ke-2
        end
        R2L2=[R1(5:8) R1(1:4)];
        %Fungsi Permutasi Akhir
        PA=R2L2(IPI); %Operasi IP invers
        PA=bi2de(PA,'left-msb'); %Konversi desimal

        abc(:,i,kanal)=PA;
    end
end
Plainimage(:,:,kanal)=transpose(reshape(abc(:,:,kanal),br1,kl1));
imshow(mat2gray(Plainimage));
imsave
```