PENGAMANAN CITRA *GRAYSCALE* MENGGUNAKAN ALGORITMA AES 128 DENGAN KUNCI CITRA *GRAYSCALE*

SKRIPSI

Oleh

AHMAD KHOIRUL UMAM
NIM 131810101036

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018

PENGAMANAN CITRA *GRAYSCALE* MENGGUNAKAN ALGORITMA AES 128 DENGAN KUNCI CITRA *GRAYSCALE*

SKRIPSI

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

Oleh

AHMAD KHOIRUL UMAM
NIM 131810101036

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2018

PERSEMBAHAN

Bismillahirrohmanirrohim

Dengan rahmat Allah SWT yang Maha Pengasih lagi Maha Penyayang, sholawat serta salam terhadap baginda Rasulullah Muhammad SAW. Saya persembahkan skripsi ini sebagai wujud syukur dan rasa terima kasih untuk:

1. Orang tua saya, Bapak Suyanto dan Ibu Harmiati, yang selalu memberikan segala bentuk dukungan, motivasi, dan do'anya untuk kemudahan, kelancaran, dan kesuksesan saya.
2. Seluruh Keluarga besar yang juga memberikan segenap dukungan dan doanya untuk saya.
3. Teman-teman ATLAS 13 yang sudah berjuang bersama 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.

MOTTO

"*Perbedaan orang bodoh dan jenius adalah orang jenius punya batasannya* ". [1]

"*Jika sebuah jendela kesempatan muncul, jangan turunkan tirainya*". [2]

[1] Albert Einstein
[2] Tom Peters

PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama: Ahmad Khoirul Umam

NIM : 131810101036

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Pengamanan citra *grayscale* menggunakan algoritma *AES* 128 dengan kunci citra *grayscale*" adalah benar-benar hasil karya sendiri, kecuali kutipan yang telah disebutkan sumbernya, belum pernah diajukan di institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

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

Jember, November 2018

Yang menyatakan,

Ahmad Khoirul Umam

NIM 131810101036

SKRIPSI

PENGAMANAN CITRA *GRAYSCALE* MENGGUNAKAN ALGORITMA AES 128 DENGAN KUNCI CITRA *GRAYSCALE*

Oleh

Ahmad Khoirul Umam
NIM 131810101036

Pembimbing:

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

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

PENGESAHAN

Skripsi berjudul "Pengamanan citra *grayscale* menggunakan algoritma *AES* 128 dengan kunci citra *grayscale*" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Tim Penguji:

Ketua,	Anggota I,
Ahmad Kamsyakawuni, S.Si., M.Kom.	Abduh Riski, S.Si., M.Si.
NIP. 197211291998021001	NIP. 199004062015041001
Anggota II,	Anggota III,
Kosala Dwidja Purnomo, S.Si., M.Si.	Ikhsanul Halikin, S.Si., M.Si.
NIP. 196908281998021001.	NIP. 198610142014041001

Mengesahkan
Dekan,

Drs. Sujito, Ph.D.

NIP. 196102041987111001

RINGKASAN

Pengamanan Citra *Grayscale* menggunakan Algoritma *AES* 128 dengan kunci Citra *Grayscale*; Ahmad Khoirul Umam, 131810101036; 2018; 75 Halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Perkembangan era globalisasi yang sangat pesat memberikan dampak yang signifikan terhadap perkembangan teknologi dan informasi saat ini. Perubahan yang terjadi tidak hanya berdampak postif namun juga ada yang berdampak negatif. Salah satunya adalah penyalahgunaan data privasi seseorang oleh pihak yang tidak berwenang. Perlindungan data dan informasi diperlukan untuk menjamin data privasi seseorang aman dan tidak disalahgunakan oleh orang lain.

Kriptografi adalah ilmu untuk menjaga atau mengamankan suatu informasi dengan cara mengacak atau menyembunyikan suatu informasi agar sulit untuk dianalisa. Algoritma yang digunakan dalam penelitian ini adalah algoritma *AES* 128 yang diubah kuncinya menggunakan citra *grayscale*. Algoritma *AES* merupakan Algoritma modern pengganti algoritma *DES* , dan merupakan algoritma yang banyak dipilih untuk mengamankan suatu data/pesan karena efisien dan pengamanannya yang kuat. Pada prosesnya, kunci citra *grayscale* dibagi pikselnya menjadi 16 blok, pada setiap blok tersebut dioperasikan pikselnya dengan operasi XOR sehingga akan didapatkan 1 karakter tiap bloknya. Dari proses XOR piksel-piksel yang ada pada 16 blok tersebut nantinya akan didapatkan 16 karakter atau kunci dengan panjang 128 bit.

Data yang digunakan dalam penelitian ini adalah *plain image* berupa citra *grayscale* dan kunci berupa citra *grayscale* juga. Citra *graysaclae* (*plain* image) akan dienkripsi dengan kunci citra yang juga citra grayscale. Mulanya, kunci akan diambil dari sebuah citra *grayscale* dengan membagi 16 blok agar mendapatkan kunci dengan panjang 128 bit, lalu dibangkitkan kunci tersebut dengan pembangkit kunci pada algoritma *AES* sehingga akan mendapatkan 10 sub-kunci. Kemudian dilakukan enkripsi dengan 10 sub-kunci yang didapat tadi menggunakan algoritma *AES*. Hasil

dari proses enkripsi akan menghasilkan sebuah citra tersandi atau *cipher image* yang sudah tidak mengandung informasi dari *plain image*-nya.

Analisis keamanan digunakan untuk mengetahui keamanan dari algoritma atau metode yang diajukan. Analisis keamanan yang digunakan yaitu analisis analisis diferensial, analisis dan analisis sensitivitas kunci. Hasil dari kedua analisis tersebut menunjukkan bahwa algoritma yang diajukan aman terhadap serangan karena memiliki kunci yang variatif, sensitif.

PRAKATA

Puji syukur kehadirat Allah SWT atas rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Pengamanan citra *grayscale* menggunakan algoritma *AES* 128 dengan kunci citra *grayscale*". Skripsi ini disusun untuk memenuhi salah satu syarat pada program pendidikan strata satu (S1) Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Skripsi ini disusun melalui beberapa tahap, baik dalam bentuk seminar maupun bimbingan intensif. 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, Kosala Dwidja Purnomo, S.Si., M.Si, selaku Dosen Penguji I, Ikhsanul Halikin, 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. Kedua orang tua dan keluarga besar yang selalu mendukung dan memberikan seluruh bantuannya dari awal sampai terselesaikannya skripsi ini;
6. Teman-teman dan semua pihak yang juga telah membantu terselesaikannya skripsi ini.

Penulis menyadari masih banyak kekurangan dalam penulisan skripsi ini. Namun, suatu usaha tidak akan berakhir dan berhasil jika tidak dimulai. Oleh karena

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

Jember,November 2018

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman

BAB 1. PENDAHULUAN

1.1. Latar Belakang

Melihat dari perkembangan zaman, teknologi menjadi semakin pesat dari tahun ketahun. Berkat perkembangan teknologi yang begitu pesat memungkinkan manusia dapat berkomunikasi dan saling bertukar informasi/data secara jarak jauh. Antar kota antar wilayah antar negara bahkan antar benua bukan merupakan suatu kendala lagi dalam melakukan komunikasi dan pertukaran data. Seiring dengan itu tuntutan akan sekuritas (keamanan) terhadap kerahasiaan informasi yang saling dipertukarkan tersebut semakin meningkat. Begitu banyak pengguna seperti departemen pertahanan, suatu perusahaan atau bahkan individu-individu tidak ingin informasi yang disampaikannya diketahui oleh orang lain atau kompetitornya atau negara lain. Teknologi berupa komputer saat ini sangat dibutuhkan manusia untuk menyimpan suatu data yang bersifat rahasia dan hanya dapat diakses oleh orang-orang tertentu. Oleh karena itu dikembangkanlah cabang ilmu yang mempelajari tentang cara-cara pengamanan data atau dikenal dengan istilah Kriptografi.

Kriptografi adalah salah satu ilmu untuk melindungi data dengan teknik pengubahan maupun pengacakan data dalam bentuk kode tertentu sehingga data tersebut tidak terlihat seperti aslinya. Kriptografi terdapat dua konsep utama yakni enkripsi dan dekripsi. Enkripsi adalah proses dimana informasi/data yang hendak dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya dengan menggunakan algoritma tertentu. Dekripsi adalah kebalikan dari enkripsi yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi awal.

Rijmen dan Daemen (1999), mengajukan sebuah proposal standard algoritma kriptografi baru yang dinamakan sebagai Algoritma *Rijndael*, sebagai pengganti algoritma kriptografi lama (*Data Encryption Standard* / DES). Algoritma DES dianggap sudah tidak aman lagi karena dengan perangkat keras khusus kuncinya mampu ditemukan dalam beberapa hari. Algoritma *Rijndael* merupakan algoritma

yang banyak dipilih untuk mengamankan suatu data/pesan karena efisien dan pengamanannya yang kuat. Pada tahun 2001, algoritma yang diajukan oleh Rijmen dan Daemen ini kemudian dipilih oleh *National Institute of Standards and Technologi* (NIST) dengan sebutan *Advanced Encryption Standart* (AES) sebagai pengganti algoritma lama (DES). Algoritma AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128, 192, dan 256.

Fadhillah (2012), melakukan penelitian tentang algoritma AES dalam pengamanan citra digital. Penelitian tersebut menggunakan gambar sebagai plain image dan mencoba beberapa model kunci seperti huruf, angka dan symbol. Hasil dari penelitian tersebut yaitu keamanan pengamanan citra dengan algoritma AES bisa lebih optimal jika menggukanan jenis kombinasi dari angka, huruf dan symbol.

Pada penelitian ini, penulis memanfaatkan berbagai teknik pengamanan suatu informasi yang sudah dipaparkan. Penulis akan meneliti dan membuat suatu aplikasi untuk mengamankan suatu data berupa citra *graysacle*, dengan kunci citra *grayscale* menggunakan algoritma *AES* dengan panjang kunci 128 bit. Citra tersebut akan dibagi menjadi 16 blok dan di XOR-kan pikselnya pada tiap blok tersebut, sehingga menghasilkan 16 karakter untuk dijadikan kunci pada algoritma AES 128.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah tertera diatas, maka rumusan masalah ditekankan pada :

a. Bagaimana cara membangkitkan kunci *AES* 128 dari sebuah citra *grayscale*.
b. Bagaimana proses enkripsi dan dekripsi citra *grayscale* menggunakan algoritma *AES* dengan kunci citra *grayscale*.
c. Bagaimana Analisis keamanan dari hasil enkripsi algoritma AES 128 dengan metode yang diajukan.

1.3. Tujuan

Tujuan dari penelitian ini adalah sebagai berikut :

a. Membangkitkan kunci *AES* 128 dari citra *grayscale*.
b. mengenkripsi dan mendekripsi citra *grayscale* menggunakan algoritma *AES* dengan kunci citra *grayscale*.
c. Menganalisis hasil dari enkripsi algoritma AES dengan metode yang telah diajukan.

1.4. Manfaat

Manfaat yang ingin diperoleh dengan adanya penelitian ini sebagai berikut :

a. Mengetahui cara membangkitkan kunci algoritma *AES* 128 dari citra *grayscale*.
b. Mengetahui hasil enkripsi dan deskripsi dari algoritma *AES* 128 dengan kunci citra *grayscale*.
c. Mengetahui hasil analisis dari algoritma *AES* 128 dengan metode yang sudah diajukan.

BAB II. TINJAUAN PUSTAKA

2.1. Kriptografi

Kriptografi berasal dari Bahasa Yunani, yaitu "Kripto" yang berarti rahasia (secret) dan "Graphia" yang berarti tulisan (writing). Kriptografi adalah ilmu yang digunakan untuk menjaga keamanan dari pesan ketika pesan tersebut dikirim dari suatu tempat ke tempat yang lain. Sedangkan menurut Munir (2006), kriptografi merupakan ilmu yang mempelajari tentang tulisan rahasia atau pesan tersembunyi. Secara garis besar, ilmu kriptografi mempelajari teknik untuk menyembunyikan, melindungi dan mengamankan suatu informasi dengan cara membuat suatu bentuk baru yang susah dipahami maknanya. Kriptografi juga bisa diartikan sebagai ilmu tentang pengacakan pesan dengan fungsi matematika agar pihak berwenang tidak bisa membaca pesan tersebut. Pengacakan pesan dalam ilmu kriptografi biasa disebut dengan enkripsi (penyandian data). Sedangkan untuk mengembalikan suatu pesan tersebut yang sudah dienkripsi ke pesan semula dinamakan dengan deskripsi (pengembalian data).

Gambar 2.1 Proses enkripsi dan dekripsi

Gambar 2.1 merupakan proses enkripsi dan dekripsi dalam kriptografi. Istilah penting dalam kriptografi yang harus diketahui adalah *plaintext*, *ciphertext*, enkripsi, dekripsi, kunci (*key*), dan algoritma. *Plaintext* merupakan informasi awal atau pesan yang bisa dibaca dan dimengerti maknanya. *Ciphertext* merupakan informasi hasil pesan *plaintext* yang sudah disandikan. Enkripsi adalah teknik untuk menyandikan *plaintext*. Dekripsi adalah teknik untuk mengembalikan ciphertext menjadi *plaintext* kembali. Kunci (key*)* berfungsi untuk mengatur dan menjalankan suatu algoritma. Sedangkan, algoritma adalah suatu metode untuk melalukan proses enkripsi dan dekripsi tersebut (Prayudi, 2005).

2.2. Citra

Citra (*image*) atau istilah lain untuk gambar adalah salah satu media yang memegang peranan sangat penting sebagai bentuk informasi visual. Citra dapat membentuk dua dimensi dan tiga dimensi untuk mempresentasikan bentuk suatu objek.

Menurut Murni (1992) bahwa citra sebagai keluaran dari suatu sistem perekam 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.

Suatu citra dapat didefinisikan sebagai fungsi $f(x, y)$ dengan x dan y adalah suatu koordinat dan f dari (x, y) menyatakan intensitas atau tingkat keabuan dari citra pada suatu titik. Derajat keabuan memiliki rentang nilai dari l_{min} sampai l_{max} atau $l_{min} < f < l_{max}$. Selang (l_{min}, l_{max}) disebut sebagai skala keabuan seperti yang ditunjukkan pada Gambar 2.2.

Gambar 2.2 Koordinat titik pada suatu citra

(Sumber: Gonzalez, 1977)

Agar suatu citra dapat dilakukan proses komputasi pada komputer, citra harus didigitalisasi terlebih dahulu dimana proses digitalisasi merupakan representasi citra dari fungsi kontinu menjadi nilai-nilai diskrit. Hasil digitalisasi ini disebut sebagai citra digital. Citra digital dapat ditulis dalam bentuk matrik sebagai berikut:

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

Indeks baris (i) dan indeks kolom (j) menyatakan suatu koordinat titik pada citra, dan $f(i,j)$ merupakan intensitas (derajat keabuan) pada titik (i,j). Masing-masing elemen pada citra digital (elemen matriks) disebut dengan pixel atau pel. Jadi citra yang berukuran N x M mempunyai NM buah pixel (Dulimarta, 1997).

Citra grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixel nya, artinya nilai Red = Green = Blue. Nilai-nilai tersebut digunakan untuk menunjukkan intensitas warna. Citra yang ditampilkan terdiri atas warna abu-abu, bervariasi pada warna hitam sebagai bagian intensitas terlemah dan putih sebagai intensitas terkuat seperti yang ditunjukkan pada Gambar 2.3.

Gambar 2.3 Citra *grayscale* 256x256 piksel dengan kedalaman 8-bit
(Sumber : MATLAB library)

Citra *grayscale* berbeda dengan citra hitam-putih, dimana pada konteks komputer, citra hitam-putih hanya terdiri atas dua warna saja yaitu, hitam dan putih. Pada citra grayscale, warna bervariasi antara hitam dan putih menyebabkan terdapat warna keabuan dengan berbagai tingkat dari hitam hingga mendekati putih. Umumnya citra *grayscale* direpresentasikan dalam 8-bit yang berarti terdapat 2^8 atau 256 derajat keabuan dengan rentang nilai 0-255, dimana 0 menunjukkan level intensitas paling gelap dan 255 menunjukkan intensitas paling terang.

2.3 Sistem Basis pada Bilangan

Sistem bilangan berdasarkan basisnya adalah bilangan desimal dan bilangan biner. Bilangan desimal merupakan bilangan yang memiliki basis 10, yaitu : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Bilangan biner merupakan sistem bilangan yang memiliki 2 basis, yakni 0 dan 1. Bilangan oktal merupakan sistem bilangan yang memiliki 8 basis, yakni 0, 1, 2, 3, 4, 5, 6, dan 7. Sedangkan bilangan hexadesimal merupakan sitem bilangan yang memiliki 16 basis, yakni 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, dan F (Feoh, 2011).

Bilangan-bilangan tersebut dapat dikonversi menjadi satu lainnya. Berikut merupakan beberapa langkah untuk melakukan konversi bilanganbilangan tersebut.

a. Konversi Bilangan Desimal ke Bilangan Biner

Salah satu cara dalam mengkonversi bilangan desimal menjadi bilangan biner adalah dengan cara membagi bilangan desimal dengan dua kemudian diambil sisa pembagiannya. Sisa-sisa pembagian membentuk jawaban, yaitu sisa yang pertama akan menjadi *least significant bit* (LSB) dan sisa yang terakhir menjadi *Most Significant Bit* (MSB).

Contoh :

Bilangan desimal 36

36 : 2 = 18 sisa 0 (*LSB*)
18 : 2 = 9 sisa 0
9 : 2 = 4 sisa 1
4 : 2 = 2 sisa 0
2 : 2 = 1 sisa 0
1 : 2 = 0 sisa 1
0 : 2 = 0 (*MBS*) sisa 0

Bilangan biner ditulis dari bawah ke atas, maka bilangan biner dari 36 adalah 0100100

b. Konversi Bilangan Biner ke Bilangan Desimal

Sistem bilangan biner adalah susunan bilangan yang mempunyai basis 2 sebab sistem bilangan ini menggunakan dua nilai koefisien yang mungkin yaitu 0 dan 1. Konversi dilakukan dengan menggunakan persamaan (2.1).

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

dimana

r = basis bilangan biner yaitu 2

i = posisi nilai biner, dimulai dari 0

d = nilai biner

n = banyaknya angka biner

Contoh :

$0100100 = 0x2^0 + 0x2^1 + 1x2^2 + 0x2^3 + 0x2^4 + 1x2^5 + 0x2^6$

$= 0 + 0 + 4 + 0 + 0 + 32 + 0$

$= 36$

2.4 ASCII (*American Standard Code for Information Interchange*)

Kode ASCII (*American Standart Code for Information Interchange*) merupakan suatu kumpulan karakter, baik huruf maupun simbol seperti Hex dan Unicode sesuai dengan *standart* yang sudah ditentukan. Kode ASCII bersifat universal, dan digunakan oleh komputer untuk menunjukkan teks. Kode ASCII masih dibagi lagi menjadi beberapa bagian, seperti kode ASCII *Control Characters*, ASCII *Printable Character*, dan *The Extended* ASCII *Codes*. Ketiga macam kode ASCII tersebut dapat dilihat pada Lampiran A.

2.5 Kriptografi AES

AES (*Advanced Encryption Standard*) merupakan algoritma *cippher* yang cukup aman untuk melindungi data atau informasi yang bersifat rahasia. Pada tahun 2001, AES digunakan sebagai standar algoritma kriptografi terbaru yang dipublikasikan oleh *NIST* (*National Institute of Standard and Technology*) sebagai pengganti algoritma *DES* (*Data Encryption Standard*) yang sudah berakhir masa penggunaannya. Algoritma AES adalah algoritma kriptografi yang dapat mengenkripsi dan mendekripsi data dengan panjang kunci yang bervariasi, yaitu 128, 192, dan 256.

Input dan *output* dari algoritma AES terdiri dari urutan data sebesar 128 bit. Urutan data yang sudah terbentuk dalam satu kelompok 128 bit tersebut disebut juga sebagai blok data atau *plaintext* yang nantinya akan dienkripsi menjadi *ciphertext*. *Cipher key* dari AES terdiri dari *key* dengan panjang 128 bit, 192 bit, atau 256 bit. Perbedaan panjang kunci akan mempengaruhi jumlah *round* yang akan diimplementasikan pada algoritma *AES* ini. Berikut ini adalah Tabel 2.1 yang memperlihatkan jumlah *round* / putaran (Nr) yang harus diimplementasikan pada masing-masing panjang kunci.

Tabel 2.1 Banyaknya iterasi Algoritma Rijndael

(Sumber : Rijmen dan Daemen, 1999)

AES-bit	**Panjang *Key* (Nk)**	**Panjang *Block* (Nb)**	**Banyak Iterasi (Nr)**
128 bit	4 words	4 words	10
192 bit	6 words	4 words	12
256 bit	8 words	4 words	14

Patel dan Padate (2015), menjelaskan tahapan proses enkripsi dan dekripsi algoritma AES pada pesan berupa citra dengan menggunakan kunci AES-128 bit. Hal pertama sebelum melakukan proses enkripsi dan dekripsi adalah membangi *plaintext* menjadi partisi matriks 128 bit kemudian direpresentasikan dalam hexadesimal. Dilanjutkan dengan membangun 10 sub kunci yang akan dipakai pada setiap iterasi. Kunci yang pertama kali di *input* oleh *user* disebut sebagai *cipherkey*. *Cipherkey* dari *AES*-128 tersebut dibagi menjadi 4 partisi 32 bit dan direpresentasikan ke bentuk heksadesimal dalam matriks 4x4. Representasi ini akan menghasilkan 4 *words* yakni *words*-0 (W_0), *words*-1 (W_1), *words*-2 (W_2), dan *words*-3 (W_3).

Pada kasus ini akan dibangkitkan 40 *word* atau 10 sub-*cipherkey* baru, yang selanjutnya akan dipakai pada tiap iterasi saat proses enkripsi. Untuk menghasilkan *word* ke-4 atau kolom pertama pada sub kunci pertama (W_i), maka prosesnya terdiri dari beberapa operasi yang berurutan yaitu sebagai berikut :

a. Operasi *RotWord*

Operasi ini melakukan perputaran 8 bit pada 32 bit W_{i-1}, dengan menggeser kolom secara siklis ke atas satu kali. *Words* (k_0,k_1,k_2,k_3) menjadi *words* (k_1,k_2,k_3,k_0).

b. Operasi *SubWord*

Hasil *RotWord* kemudian disubstitusi dengan nilai dari tabel S-*Box*. Tabel S-*Box* algoritma *AES* direpresentasikan dalam tabel dberikut:

Tabel 2.2 S-*Box AES Algoritm*
(Sumber: Patel dan Padate, 2015)

		y															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
x	**0**	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	Fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	Ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	Fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	Ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	Dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	A	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	B	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	C	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	D	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	E	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	F	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

c. Operasi XOR

Hasil operasi *SubWord* kemudian di XOR dengan nilai konstan R-*Con* yang bersesuaian tiap round. Matriks dan formula dari R-*Con* seperti pada persamaan 2.4

$$R\mathit{Con}\,[j] = (\mathrm{RC}\,[j], 0, 0, 0), \tag{2.4}$$

dimana,

$$\mathrm{RC}\,[J] = 01_{16} \tag{2.5}$$

$$\mathrm{RC}\,[j] = 2 * \mathrm{RC}\,[j - 1] \tag{2.6}$$

dengan sifat operasi * merupakan perkalian yang didefinisakan pada GF(2^8). Pada GF(2^8), nilai dari maksimal dari RC adalah $255_{(10)}$, dan jika didapatkan nilai RC yang lebih dari $255_{(10)}$ saat proses perkalian, maka hasilnya akan dimodulo dengan polinomial *irreducible*. Polinomial *irreducible* pada algoritma AES, yaitu:

$$f(x) = x^8 + x^4 + x^3 + x + 1 \qquad (2.7)$$

nilai $f(x)$ tersebut setara dengan $283_{(10)}$ ($100011011_{(2)}$). Sedangkan nilai R-Con dalam hexadesimal yang didapat yaitu,

01	03	04	08	10	20	40	80	1*B*	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

(Satria, 2009)

d. Operasi XOR dengan W_{i-4}

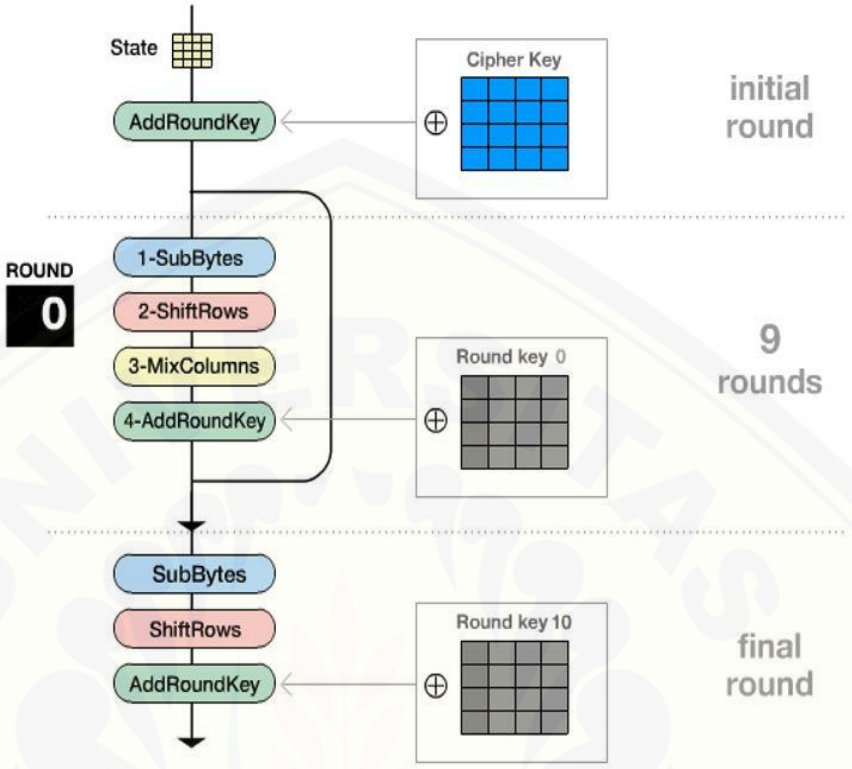
Operasi ini merupakan operasi terakhir untuk mendapatkan W_i. Pada tahap ini, hasil dari proses sebelumnya di XOR dengan W_{i-4}, dan nilai dari operasi ini adalah nilai dari W_i.

Proses selanjutnya, untuk mendapatkan nilai dari W_{i+1} dan seterusnya, yaitu cukup dengan melakukan operasi XOR antara W_i dengan W_{i-3}. Lakukan langkah ini sampai mendapatkan seluruh nilai untuk sub-*cipherkey*.

Begitu seluruh sub-*cipherkey* sudah didapatkan, maka akan dilanjutkan dengan proses enkripsi ataupun proses dekripsi:

a. Proses Enkripsi AES

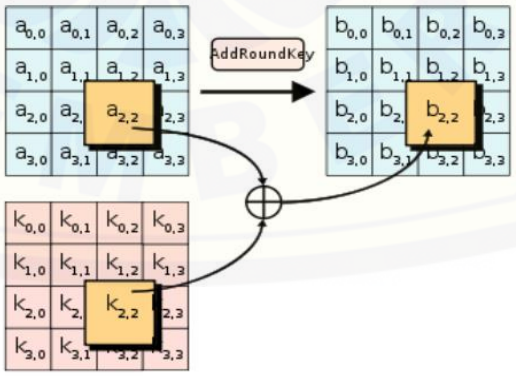
Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi *bytes*, yaitu *SubBytes*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pada awal proses enkripsi, *input* yang telah dicopykan ke dalam *state* akan mengalami transformasi *byte AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulang-ulang sebanyak *Nr*. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns*. Ilustrasinya dapat dilihat pada Gambar 2.4 di bawah ini :



Gambar 2.4 Ilustrasi Proses Enkripsi AES
(Sumber : Rinaldi Munir 2004)

1. *AddRoundKey*

Pada proses ini yaitu melakukan operasi XOR antara 128 bit partisi plaintext pertama dengan 128 bit *cipherkey* dan sub-*cipherkey* yang sudah dibangkitkan sebelumnya. Prosesnya dapat dilihat pada Gambar 2.5 :

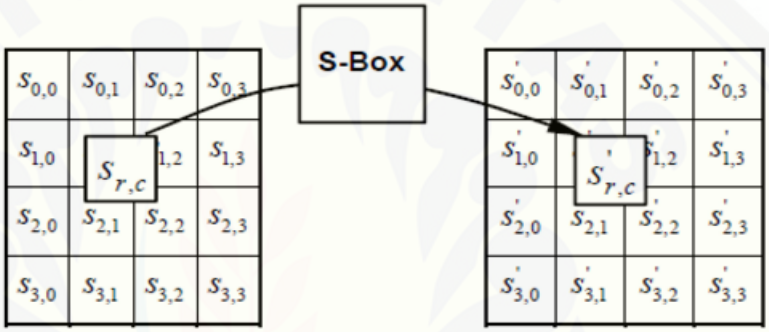


Gambar 2.5 Transformasi *AddRoundKey*

(Sumber : Munir, 2004)

2. *SubBytes*

SubBytes merupakan transformasi *byte* dimana setiap elemen pada *state* akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Untuk setiap *byte* pada *array* state, misalkan *S*[*r*, *c*] = *xy*, yang dalam hal ini *xy* adalah digit heksadesimal dari nilai *S*[*r*, *c*], maka nilai substitusinya, dinyatakan dengan *S'*[*r*, *c*], adalah elemen di dalam tabel substitusi yang merupakan perpotongan baris *x* dengan kolom *y*. Gambar 2.6 mengilustrasikan pemetaan *byte* pada setiap *byte* dalam *state*.

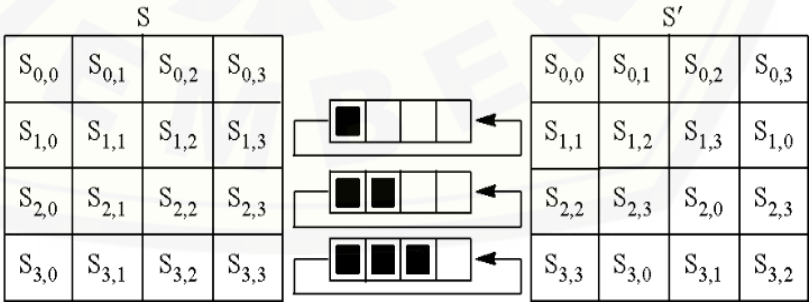


Gambar 2.6 Pengaruh pemetaan pada setiap *byte* dalam *state*

(Sumber : Munir, 2004)

3. *Shiftrows*

Transformasi *Shiftrows* pada dasarnya adalah proses pergeseran *byte* dimana byte paling kiri akan dipindahkan menjadi *byte* paling kanan (rotasi *byte*). Proses pergeseran *Shiftrow* ditunjukkan dalam Gambar 2.7.



Gambar 2.7 Transformasi *Shiftrows*

(Sumber : Munir, 2004)

4. *MixColumns*

MixColumns mengoperasikan setiap elemen yang berada dalam satu kolom pada state. Secara lebih jelas, transformasi *mixcolumns* dapat dilihat pada Persamaan 2.8.

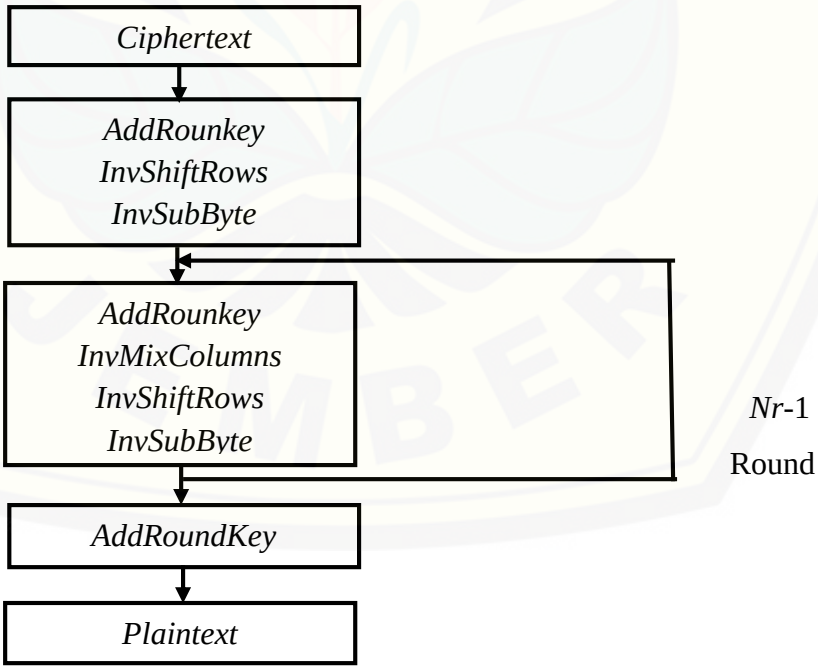
$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad (2.8)$$

Hasil dari perkalian matriks diatas dapat dianggap seperti perkalian yang ada pada Persamaan 2.9.

$$\begin{aligned} S'_{0,c} &= (\{02\} \cdot S_{0,c}) \oplus (\{03\} \cdot S_{1,c}) \oplus S_{2,c} \oplus S_{3,c} \\ S'_{1,c} &= S_{0,c} \oplus (\{02\} \cdot S_{1,c}) \oplus (\{03\} \cdot S_{2,c}) \oplus S_{3,c} \\ S'_{2,c} &= S_{0,c} \oplus S_{1,c} \oplus (\{02\} \cdot S_{2,c}) \oplus (\{03\} \cdot S_{3,c}) \\ S'_{3,c} &= (\{03\} \cdot S_{0,c}) \oplus S_{1,c} \oplus S_{2,c} \oplus (\{02\} \cdot S_{3,c}) \end{aligned} \quad (2.9)$$

b. Proses Dekripsi AES

Proses dekripsi algoritma AES dapat dilihat pada Gambar 2.8 :



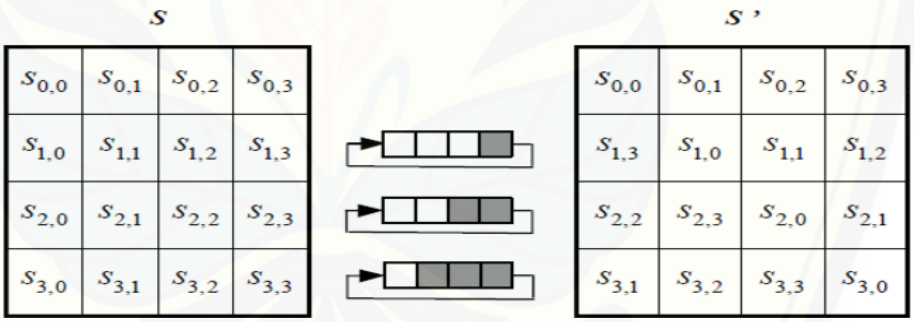
Gambar 2.8 Illustrasi Proses Dekripsi AES

Struktur proses dekripsi *AES* secara umum sama dengan proses enkripsi, tetapi pada proses dekripsi *AES* memilik proses transformasi penyusun tiap iterasi yang berbeda. Tidak hanya itu, transformasi yang digunakan pun merupakan transformasi kebalikan atau *invers* dari proses transformasi penyusun setiap iterasi pada proses enkripsi (Fadhillah, 2012).

Meskipun proses pembentukan kunci pada dekripsi identik dengan enkripsi, akan tetapi proses penjadwalan penggunaan kunci pada setiap iterasi berbeda. Penjadwalan kunci pada proses dekripsi dimulai dari word ke-43 sampai word ke-0 atau dimulai dari sub kunci ke 10 sampai *cipherkey*.

1. *InvShiftrows*

InvShiftRows adalah transformasi *byte* yang berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InvShiftRows,* dilakukan pergeseran bit ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran bit ke kiri. Ilustrasi transformasi *InvShiftRows* terdapat pada Gambar 2.9:



Gambar 2.9 Transformasi *InvShiftrows*
(Sumber : Rinaldi Munir 2004)

2. *InvSubBytes*

InvSubBytes juga merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*. Pada *InvSubBytes*, tiap elemen pada state dipetakan dengan menggunakan tabel *Inverse* S-Box. Tabel *Inverse* S-Box akan ditunjukkan dalam Gambar 2.10.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Gambar 2.10 Tabel *Invers S-Box*

(Sumber: Patel dan Padate, 2015)

3. InvMixColumns

Setiap kolom dalam *state* dikalikan dengan matrik perkalian dalam AES. Perkalian dalam matrik dapat dituliskan seperti pada Persamaan 2.10.

$$\begin{bmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{bmatrix} \quad (2.10)$$

Hasil dari perkalian dalam matrik ditunjukkan pada Persamaan 2.11.

$$\begin{aligned} S'_{0,c} &= (\{0E\} \cdot S_{0,c}) \oplus (\{0B\} \cdot S_{1,c}) \oplus (\{0D\} \cdot S_{2,c}) \oplus (\{09\} \cdot S_{3,c}) \\ S'_{1,c} &= (\{09\} \cdot S_{0,c}) \oplus (\{0E\} \cdot S_{1,c}) \oplus (\{0B\} \cdot S_{2,c}) \oplus (\{0D\} \cdot S_{3,c}) \\ S'_{2,c} &= (\{0D\} \cdot S_{0,c}) \oplus (\{09\} \cdot S_{1,c}) \oplus (\{0E\} \cdot S_{2,c}) \oplus (\{0B\} \cdot S_{3,c}) \\ S'_{3,c} &= (\{0B\} \cdot S_{0,c}) \oplus (\{0D\} \cdot S_{1,c}) \oplus (\{09\} \cdot S_{2,c}) \oplus (\{0E\} \cdot S_{3,c}) \end{aligned} \quad (2.11)$$

2.6 Analisis Keamanan

Proses Analisis kemanan suatu metode atau algoritma enkripsi sangat penting untuk dilakukan untuk mengetahui seberapa aman ketika algoritma enkripsi itu digunakan. Beberapa analisis keamanan suatu algoritma enkripsi adalah sebegai berikut:

a. Analisis Differensial

Analisis Diferensial digunakan untuk menentukan perbedaan dari dua citra. Langkahnya yaitu dengan menghitung nilai dari *number of pixels change rate* (*NPCR*). NPCR dengan nilai lebih besar dari 90% akan menyulitkan kriptanalisis dalam mencari hubungan statistik antara citra asli dengan citra tersandi (Dharmaadi, dkk., 2013).

$$NPCR = \frac{\sum_i \sum_j D(i,j)}{W \times H} \times 100\% \qquad (2.12)$$

dengan W dan H merepresentasikan lebar citra dan tinggi citra, dan bentuk dari $D(i,j)$ dapat ditentukan seperti berikut:

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

dengan $C(i,j)$ dan $C'(i,j)$ merepresentasikan nilai derajat keabuan dari baris *i*, dan kolom *j* dari citra *C* dan *C′* (Atani, dkk., 2013).

b. Analisis Sensitivitas Kunci

Analisis sensitivitas kunci dapat dilakukan dengan dasar dari tahapan berikut,

1) Ketika kunci yang digunakan untuk mengenkripsi citra tersebut sedikit berbeda maka akan menghasilkan *cipher image* yang sangat berbeda.
2) Jika ada perbedaan kunci antara proses enkripsi dan dekripsi maka tidak akan memperoleh *plain image* yang diinginkan.

Song dan Qiao (2015).

c. Analisis *Brute Force Attack*

Proses Analisis kemanan suatu metode atau algoritma sangat penting untuk dilakuakan untuk mengetahui seberapa aman ketika algoritma itu digunakan. *Brute force attack* adalah metode untuk menemukan skema kriptografi dengan mencoba

semua kemungkinan *password* atau kunci. *Brute force attack* adalah metode untuk menemukan skema kriptografi dengan jumlah besar kemungkinan kunci. *Brute force attack* memungkinkan dapat menyerang kunci privat dihampir semua skema kriptografi, tipe serangan ini bergantung pada ukuran kunci dan mekanisme pada enkripsi yang digunakan

Dalam bidang kriptografi, *brute force attack* merupakan teknik yang digunakan penyerang untuk menemukan kunci enkripsi dengan cara mencoba semua kemungkinan kunci.

Contoh :

Kunci yang digunakan pada algoritma AES yaitu kunci dengan panjang 128 bit. Maka jumlah kunci yang harus dievaluasi oleh pihak lawan adalah sebanyak $(2)(2)(2) \dots (2)(2) = 2^{128} = 3{,}4028e + 38$, artinya ada $3{,}4028\ x\ 10^{38}$ kemungkinan kunci yang harus dicoba. Jika 1 tahun ada 31.536.000 detik dan 1 komputer untuk menganalisis 1 kunci membutuhkan 1 detik, maka ada $1{,}0790\ x\ 10^{31}$ tahun untuk mencoba semua kunci.

Meskipun algoritma *brute force* tidak cocok karena membutuhkan waktu yang cukup lama, namun sebagaimana ciri algoritma *brute force* pada umumnya nilai plusnya terletak pada keberhasilannya yang selalu menemukan solusi (jika diberikan waktu yang cukup) (Wicaksono, 2013).

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang penulis gunakan dalam penelitian ini adalah citra *grayscale* yang terdiri dari tiga buah citra *plain image* dan tiga buah citra sebagai kunci. Percobaan dilakukan terhadap masing-masing *plain image* dengan kunci yang sudah disediakan. Gambar 3.1 dan Gambar 3.2 merupakan salah satu data yang digunakan pada penelitian ini, untuk dara selengkapnya dapat dilihat pada lampiran B.

Gambar 3.1 *plainimage* lena 256x256

Gambar 3.2 *Key* Cameraman.tif

(Sumber : MATLAB library)

3.2 Langkah-langkah Penelitian

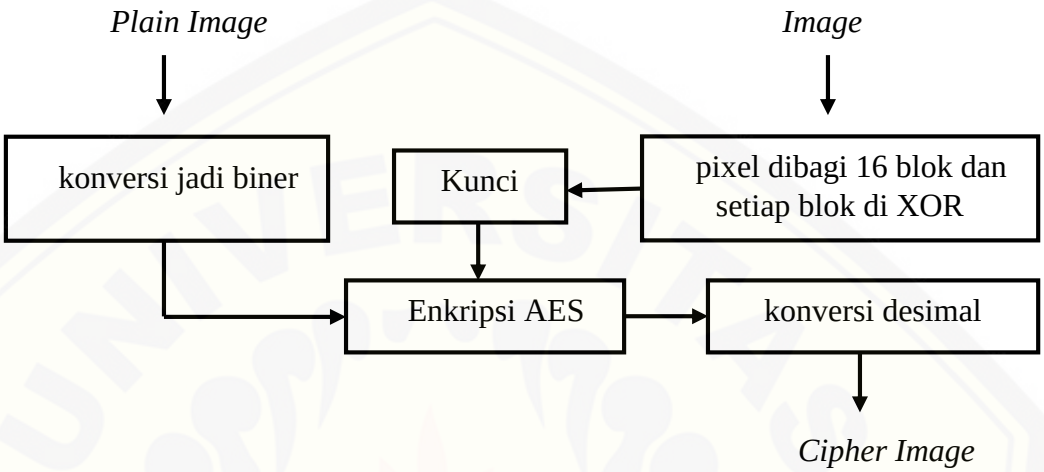
Secara sistematis, langkah-langkah penelitian yang dilakukan adalah sebagai berikut :

a. Studi Literatur

Pada tahap ini dilakukan pemahaman mengenai teori-teori terkait dengan penelitian yang dilakukan. Teori-teori tersebut adalah algoritma AES dengan panjang kunci 128 bit.

b. Analisa Data

Berikut ini merupakan langkah-langkah yang akan diterapkan pada seluruh data dan citra *grayscale* yang menjadi kunci pada algoritma AES 128.



Gambar 3.2 Proses Enkripsi

1. Proses Enkripsi

Seperti yang ditunjukkan pada Gambar 3.2 bahwa proses enkripsi bisa dilakukan dengan beberapa langkah sebagai berikut :

a) Konversi karakter *plain image* menjadi bilangan biner

Setiap karakter dalam *plain image* dikonversi menjadi bilagan desimal sesuai kode ASCII kemudian dikonversi menjadi bilangan biner.

b) Membagi pixel menjadi 16 blok dan di XOR setiap pixelnya

Membagi 16 blok dari jumlah pixel yang ada dengan mengurutkan baris pertama sampai baris terakhir. Kemudian mengoperasikan setiap pixel yang ada pada setiap blok dengan operasi XOR sehingga didapatkan 16 karakter atau 128 bit untuk dijadikan kunci pada algoritma AES 128.

c) Kunci

Kunci yang didapat hasil dari proses pembagian gambar menjadi 16 blok yang nantinya akan mendapat 16 karakter atau 128 bit untuk digunakan dalam enkripsi algoritma AES.

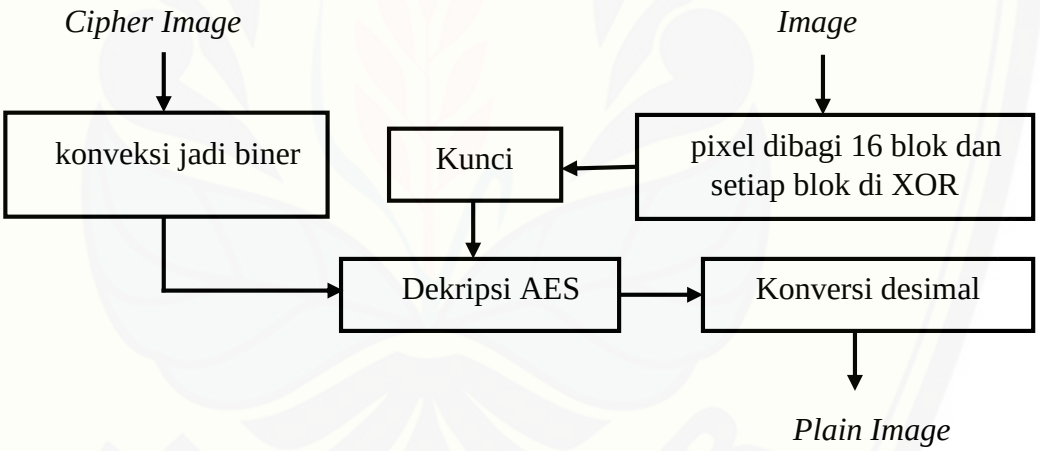
d) Enkripsi AES

Proses enkripsi dari *plain image* dan kunci yang sudah didapatkan dengan langkah sebagai beikut

1) *AddRoundKey*
2) *SubByte*
3) *Shiftrows*
4) *MixColumns*

e) Konversi desimal

Setiap bilangan biner dari hasil enkripsi tersebut dikembalikan lagi dalam bentuk desimal dan dirangkai menjadi sebuah citra. Hasil konversi ini akan menjadi *cipher image.*



Gambar 3.3 Proses Dekripsi

2. Proses Dekripsi

Seperti yang ditunjukkan pada Gambar 3.3 bahwa proses dekripsi bisa dilakukan dengan beberapa langkah sebagai berikut :

a) Konversi karakter *chipper image* menjadi bilangan biner

Setiap karakter dalam *chipper image* dikonversi menjadi bilagan desimal sesuai kode ASCII kemudian dikonversi menjadi bilangan biner.

b) Membagi pixel menjadi 16 blok dan di XOR setiap pixelnya

Membagi 16 blok dari jumlah pixel yang ada dengan mengurutkan baris pertama sampai baris terakhir. Kemudian menjumlahkan setiap pixel yang ada pada setiap blok dengan operasi XOR sehingga didapatkan 16 karakter untuk dijadikan kunci.

c) Kunci

Kunci yang didapat hasil dari proses pembagian gambar menjadi 16 blok yang nantinya akan mendapat 16 karakter atau 128 bit untuk digunakan dalam dekripsi algoritma AES

d) Dekripsi AES

Proses dekripsi dari *cipher image* dan kunci yang sudah didapatkan dengan langkah sebagai beikut :

1) *InvShiftrows*
2) *InvSubBytes*
3) *InvMixColumns*

e) Konversi desimal

Setiap bilangan biner dari hasil dekripsi tersebut dikembalikan lagi dalam bentuk desimal dan dirangkai menjadi sebuah gambar. Hasil konversi ini akan menjadi *plain image.*

c. Perancangan Program

Pada langkah ini dilakukan perancangan desain GUI (*Guide User Interface*) dengan menggunakan software MATLAB seperti tata-letak tombol-tombol serta pengaturan warna dan latar belakang agar tampilan menjadi menarik.

d. Pembuatan Program

Pembuatan program menggunakan software MATLAB, melakukan proses enkripsi dan dekripsi dengan kunci citra *grayscale* menggunakan algoritma *Advanced Encryption Standart* 128.

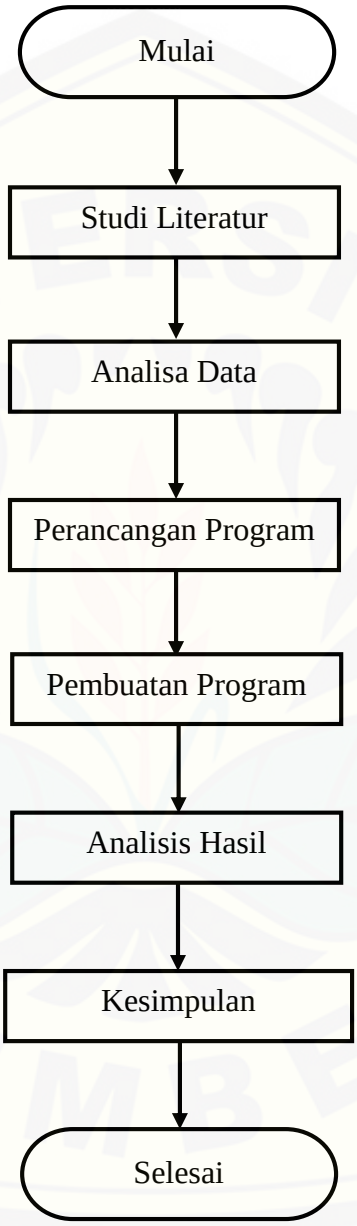
e. Analisis Hasil

Menguji jalannya program yang telah dibuat untuk menentukan apakah setiap proses telah berjalan dengan baik sesuai dengan hasil yang diinginkan atau tidak. Kemudian diuji keamanannya dari algoritma AES 128 dengan kunci citra *grayscale*.

f. Kesimpulan

Mengambil kesimpulan dari penelitian yang sudah dilakukan, yaitu menganalisis proses enkripsi dari algoritma AES 128 dengan kunci citra dalam mengubah *plain image* menjadi *cipherimage* dan sebaliknya. Serta menganalisis keamanan dari algoritma tersebut.

Flowchart dari langkah-langkah penelitian yang dilakukan dapat diamati pada Gambar 3.4.



Gambar 3.4 *Flowchart* Penelitian

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan pada penelitian ini, maka didapatkan beberapa kesimpulan sebagai berikut :

a. Proses pembangkitan kunci algoritma *AES* dari citra *grayscale* dengan mengkonversi menjadi 16 blok dan mengoperasikan setiap pikselnya pada setiap blok menggunakan operasi XOR. Hasil operasi tersebut akan mendapatkan 16 karatkter atau bilangan biner sepanjang 128 bit dan akan digunakan pada tahapan enkripsi maupun dekripsi terhadap algoritma AES.
b. Proses enkripsi dengan kunci citra *grayscale* yaitu dengan membangkitkan sebuah kunci dari citra *grayscale* tersebut menjadi 128 bit. Kemudian dilakukan tahapan-tahapan pada algoritma AES yakni pembangkitan 10 sub kunci, *AddRounKey, SubBytes, Shiftrows, MixColumns*. Sedangkan pada proses dekripsi yaitu caranya sama dengan mengkonversi kunci *grayscale* dan membangkitkan menjadi 10 sub kunci, kemudian melakukan tahapan algoritma AES dengan kunci kunci ke-10 sampai ke-1. Tahapan tersebut meliputi *AddRoundKey, InvShiftRows, InvSubBytes* dan *InvMixColumns*. Proses enkripsi maupun dekripsi memiliki putaran tahapan yang sama, yakni 10 putaran.
c. Analisis keamanan pada penelitian ini menunjukkan bahwa metode yang diajukan oleh penulis dengan menerapkan kunci citra *grayscale* untuk enkripsi algoritma AES 128 merupakan metode yang aman dalam penyandian informasi. Dibuktikan dengan *cipher image* yang memiliki nilai *NPCR* $\geq$ 90% dari semua data penelitian. Dilakukan juga percobaan enkripsi dengan kunci yang mirip dan semuanya mendapatkan nilai *NPCR* > 90% yang dapat diartikan bahwa algoritma ini memiliki kunci yang sensitif. Akan tetapi, algoritma ini tidak cocok pada data 2 (Textgray.png). Meskipun memiliki nilai *NPCR* > 90% dan kuncinya sensitif, *cipher image* terlihat masih berpola seperti *plain image*

karena pada data 2 memiliki *background* putih dan terlalu banyak piksel yang nilainya sama.

5.2 Saran

Saran yang dapat diberikan oleh penulis untuk penelitian selanjutnya yaitu memodifikasi kuncinya dengan citra *RGB* dan mengenkripsi citra *RGB*, sehingga tidak ada batasan dalam mengenkripsi citra pada algoritma AES.