Rancang Bangun Aplikasi Enkripsi dan Dekripsi Citra Digital Menggunakan Algoritma Rijndael Berbasis Java SE

Yoga Aprianto[1], Rico Kurniawan[2], Renni Angreni[3]
[1,2,3]STMIK Global Informatika MDP Jl. Rajawali No.14 Palembang
[1,2,3]PS Teknik Informatika STMIK Global Informatika MDP
e-mail: *[1]yogalionel888168@gmail.com, [2]rico.kurniawan13@gmail.com, [3]renni@mdp.ac.id

Abstrak

Citra digital merupakan salah satu data atau informasi yang sering disalahgunakan, oleh karena itu untuk menjaga keamanan dan kerahasiaan suatu data maupun informasi salah satunya dapat dengan teknik enkripsi dan dekripsi. Teknik ini berguna untuk membuat pesan, data, maupun informasi tidak dapat dibaca atau dimengerti oleh orang lain, kecuali untuk penerima yang berhak dan mengetahui teknik dekripsinya. Melihat penting dan bermanfaatnya teknik enkripsi dan dekripsi maka akan sangat baik jika metode dalam pemrosesannya menggunakan algoritma dengan tingkat keamanan yang tinggi, salah satunya dengan algoritma Rijndael. Hasil dari aplikasi ini mampu mengenkripsi dan mendekripsi file citra tanpa mengubah integritas data dari file citra tersebut.

Kata kunci—Algoritma Rijndael, Citra Digital, Enkripsi, Dekripsi

Abstract

Digital image is one of data or information that is often misused, therefore, to maintain the security and confidentiality of the data as well as one can with the information encryption and decryption techniques. This technique is useful for creating messages, data and information can not be read or understood by others, unless the recipient is entitled to know the techniques and decryption. Viewing essential and beneficial technique of encryption and decryption, it will be very good if the processing method using an algorithm with a high level of security, one of the Rijndael algorithm. The results of this application is able to encrypt and decrypt files without changing the integrity of the image data from the image file.

Keywords—Rijndael algorithm, Digital Image, Encryption, Decryption

1. PENDAHULUAN

Keamanan merupakan salah satu aspek penting dalam pengiriman data maupun komunikasi melalui jaringan. Salah satu cara untuk menjaga keamanan dan kerahasiaan suatu data maupun informasi adalah dengan teknik enkripsi dan dekripsi. Teknik ini berguna untuk membuat pesan, data, maupun informasi tidak dapat dibaca atau dimengerti oleh orang lain, kecuali untuk penerima yang berhak dan mengetahui teknik dekripsinya. Teknik enkripsi dan dekripsi dikenal dan dipelajari dalam kriptografi.

Kriptografi merupakan ilmu yang mempelajari mengenai cara mengamankan suatu informasi. Pada tahun 1990-an, algoritma enkripsi yang banyak dipakai adalah algoritma DES (*Data Encryption Standard*). Namun, seiring dengan makin canggihnya teknologi dan berkembangnya dunia *cryptanalysis*, maka keamanan data dengan algoritma DES yang menggunakan kunci sepanjang 56 bit dianggap tidak memadai lagi, karena itu pada tahun 2000 terpilihlah algoritma Rijndael sebagai standar algoritma kriptografi baru pengganti algoritma DES, yang juga dinamakan sebagai algortima AES. Informasi berupa citra digital telah digunakan secara luas dalam berbagai macam bidang seperti pemerintahan, militer, badan keuangan, rumah sakit dan perusahaan perdagangan untuk menyimpan informasi penting, misalnya hasil pemeriksaan pasien dalam bidang rumah sakit, area geografi dalam bidang penelitian, posisi musuh dalam bidang militer, produk baru dalam perusahaan dan lain sebagainya.

Melihat penting dan bermanfaatnya teknik enkripsi dan dekripsi, maka akan sangat baik pula jika metode dalam pemrosesannya menggunakan algoritma dengan tingkat keamanan yang tinggi, salah satunya dengan algoritma Rijndael.

Aplikasi yang akan dibangun ini nantinya merupakan aplikasi berbasis desktop dengan dukungan bahasa pemrograman Java SE, dimana adalah bahasa pemrograman yang *cross platform.*

2. METODE PENELITIAN

2.1 Studi Literatur

2.1.1 Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat kontinyu seperti gambar pada monitor televisi, foto sinar X, hasil CT Scan dll. Sedangkan pada citra digital adalah citra yang dapat diolah oleh computer [7].

Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari *M* kolom *N* baris, dimana perpotongan antara kolom dan baris disebut piksel (piksel = *picture element*), yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat *(x,y)* adalah *f(x,y)*, yaitu besar intensitas atau warna dari piksel di titik itu. Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks berikut.

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

Berdasarkan gambaran tersebut, secara matematis citra digital dapat dituliskan sebagai fungsi intensitas *f (x,y),* dimana harga *x* (baris) dan *y* (kolom) merupakan koordinat posisi dan *f(x,y)* adalah nilai fungsi pada setiap titik *(x,y)* yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel di titik tersebut. Pada proses digitalisasi (*sampling* dan kuantitas)

diperoleh besar baris *M* dan kolom *N* hingga citra membentuk matriks *M x N* dan jumlah tingkat keabuan piksel G [7].

Ada dua jenis format file citra yang sering digunakan dalam pengolahan citra, yaitu citra *bitmap* dan citra vector. Citra *bitmap* ini menyimpan data kode citra secara digital dan lengkap (cara penyimpanannya adalah per piksel). Sedangkan pada format file citra vektor merupakan citra vektor yang dihasilkan dari perhitungan matematis dan tidak terdapat piksel, yaitu data yang tersimpan dalam bentuk vektor posisi, dimana yang tersimpan hanya informasi vektor posisi dengan bentuk sebuah fungsi [7]. Contoh format file citra antara lain adalah BMP, GIFF, TIF, JPG, IMG, dll.

2.1.2 Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya [1]. Definisi ini mungkin cocok pada masa lalu di mana kriptografi digunakan untuk keamanan komunikasi penting seperti komunikasi di kalangan militer, diplomat dan mata-mata. Namun saat ini kriptografi lebih dari sekedar *privacy*, tetapi juga tujuan data *integrity*, *authentication* dan *non repudiation* [4].

Ada 4 (empat) tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu [1] :

1. Kerahasiaan (*confidentiality*)**.**
2. Integritas data (*data integrity*).
3. Otentikasi (*authentication*).
4. Nirpenyangkal (*non-repudiation*).

2.1.3 Algoritma Kriptografi

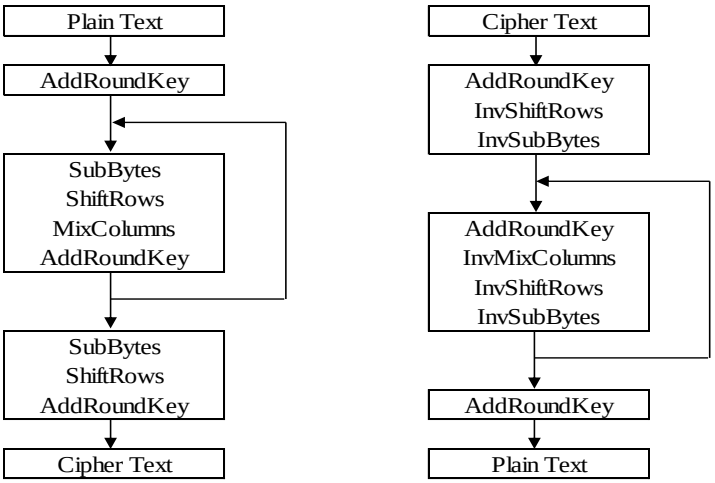
Algoritma kriptografi disebut juga *cipher* yaitu aturan untuk *enchipering* dan *dechipering*, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk *enciphering* dan *dechiphering* [4]. Algoritma kriptografi terdiri dari fungsi dasar yaitu:

a. Teknik Enkripsi
   Teknik enkripsi, merupakan hal yang sangat penting dalam kriptografi yang merupakan pengamanan data yang dikirimkan terjaga rahasianya, pesan asli disebut *plaintext* yang dirubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan *cipher* atau kode.
b. Teknik Dekripsi
   Teknik dekripsi, merupakan kebalikan dari enkripsi, pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (*plaintext*) disebut dengan dekripsi pesan. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan yang digunakan untuk enkripsi.
c. Kunci
   Kunci, yang dimaksud di sini adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi, kunci terbagi jadi 2 (dua) bagian yaitu kunci pribadi (*private key*) dan kunci umum (*public key*).

2.1.4 Algoritma Rijndael

Algoritma Rijndael menggunakan substitusi, permutasi dan sejumlah putaran yang dikenakan pada tiap blok yang akan dienkripsi dekripsi [2]. Untuk setiap putarannya, Rijndael menggunakan kunci yang berbeda. Kunci setiap putaran disebut *round key*. Ukuran blok untuk algoritma Rijndael adalah 128 *bit* (16 *byte*). Algoritma Rijndael mempunyai 3 (tiga) parameter [2] :

a. *Plaintext* adalah array yang berukuran 16 *byte*, yang berisi data masukan.
b. *Ciphertext* adalah array yang berukuran 16 *byte*, yang berisi hasil enkripsi.
c. Kunci adalah array yang berukuran 16 *byte*, yang berisi kunci *cipher* (disebut juga *chiper key*).



Gambar 1 Diagram Proses Enkripsi dan Proses Dekripsi

Gambar 1 merupakan diagram blok yang menunjukkan garis besar proses enkripsi dan dekripsi.

2.1.5 Java

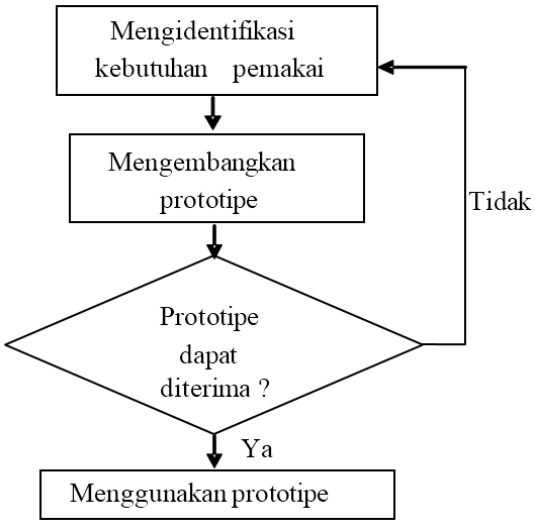
Java adalah nama sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer yang berdiri sendiri (*stand alone*) ataupun pada lingkungan jaringan [6].

Untuk beragam aplikasi yang dibuat dengan bahasa Java, java dipaketkan dalam edisi-edisi berikut :

1. Java 2 *Standard Edition* (J2SE).
2. Java 2 Enterprise Edition (J2EE).
3. Java 2 Micro Edition (J2ME).

2.2 Metode Prototyping

Prototyping merupakan metodologi pengembangan perangkat lunak yang menitik beratkan pada pendekatan aspek rancangan, fungsi dan antarmuka pengguna [5]. Metodologi *prototyping* membagi tahapan pengembangan perangkat lunaknya menjadi empat tahap yang disesuaikan dengan proses identifikasi kebutuhan seperti pada Gambar 2 [5].



Gambar 2 Proses dalam Metodologi *Prototyping*

a. Tahap Analisis (*Analysis*)
Pada tahapan ini menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Tahapan ini meliputi identifikasi objek kebutuhan, resiko kesalahan aplikasi dan merumuskan hipotesa *prototype*.

b. Tahap Rancangan (*Design*)
Setelah tahap analisis dilakukan, maka pemrogram mendesain secara terperinci sebuah rancangan *prototype* aplikasi yang menggambarkan keseluruhan aplikasi dan resiko-resiko yang mungkin berpengaruh pada aplikasi.

c. Tahap Pengujian Sistem
Pengujian sistem bertujuan untuk menemukan kesalahan-kesalahan yang terjadi pada sistem dan melakukan revisi sistem. Tahap ini penting untuk memastikan bahwa sistem bebas dari kesalahan. Setelah pengujian sistem, maka dari itu dilakukan tahap evaluasi. Evaluasi yang penulis lakukan terhadap hasil penelitian sebagai berikut :

1). Mengukur Nilai Akurasi
Parameter utama yang dapat digunakan untuk mengukur keefektifan citra yang berhasil dilakukan pada saat proses enkripsi dan proses deskripsi

$$\text{Nilai akurasi} = \frac{\text{Jumlah foto yang berhasil}}{\text{Jumlah foto uji coba}} \text{ x } 100\%$$

2). Pengujian dengan *Black Box Testing*
Metode ujicoba *black box* memfokuskan pada keperluan fungsional dari *software*. Karena itu ujicoba *black box* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program.

d. Tahap Implementasi (*Implementation*)
Prototype harus dicoba-coba untuk menentukan perilakunya dan mengumpulkan keluaran dari hasil eksekusi sistem sehingga didapat aplikasi yang sesuai dengan keinginan. Hasil dari implementasi akan dievaluasi untuk menilai kebenaran dan efisiensi aplikasi.

2.3 Perancangan dan Penerapan Algoritma

Dalam pengembangan aplikasi ini menerapkan metode enkripsi dan dekripsi dengan menggunakan algoritma Rijndael. Algoritma Rijndael merupakan jenis algoritma kriptografi yang sifatnya simetri dan *chipper block*. Dengan demikian algoritma ini menggunakan kunci yang sama pada saat enkripsi dan dekripsi serta *input* dan *output*nya berupa blok dengan jumlah bit tertentu. Algoritma Rijndael mendukung berbagai variasi ukuran kunci yang akan digunakan. Namun algoritma Rijndael mempunyai ukuran kunci yang tetap sebesar 128, 192 dan 256 bit. Penelitian ini akan dibahas algoritma Rijndael menggunakan kunci sebesar 128 bit. Ukuran blok untuk algoritma Rijndael adalah 128 bit atau 16 *byte*. Jumlah iterasi dalam proses enkripsi dan dekripsi dipengaruhi oleh ukuran kunci yang akan dipakai. Misalkan N_b adalah panjang blok dibagi 32 dan N_k adalah panjang kunci dibagi 32, maka jumlah iterasinya : $N_r = \max(N_k, N_b) + 6$

Tabel 1 Perbandingan Jumlah Iterasi Algoritma Rijndael

	Panjang Kunci (Nk)	Ukuran Blok (Nb)	Jumlah Round (Nr)
AES - 128	4	4	10
AES - 192	6	4	12
AES - 256	8	4	14

Tabel 1 menunjukkan dengan menggunakan blok *cipher* sebesar 128 bit dan kunci sebesar 128 bit, maka dalam proses enkripsi dan dekripsinya dengan algoritma Rijndael melakukan 10 iterasi.

A. *Key Schedule*
Tahap *key schedule* bertujuan membangun 10 sub-kunci yang akan digunakan pada tiap iterasi dalam tahap enkripsi dan dekripsi. Di awal akan dimasukkan suatu kunci yang disebut *cipher key*, yang seterusnya akan dilakukan ekspansi terhadap *cipher key* tersebut [3].

B. Proses Enkripsi
Proses enkripsi Rijndael diawali dengan proses *AddRoundKey* diikuti sembilan iterasi dengan struktur yang tersusun atas empat proses yaitu *SubBytes, ShiftRows, MixColumns dan AddRoundKey* [3]. Akhir proses enkripsi yaitu iterasi kesepuluh yang tersusun atas tiga proses terurut *SubBytes, ShiftRows*, dan *AddRoundKey* yang keseluruhan proses tersebut diiringi proses *key schedule* bagi setiap iterasi. Seluruh fungsi operasi (penjumlahan dan perkalian) yang tercakup dalam AES merupakan operasi-operasi yang didefinisikan dalam ruang lingkup GF(2^8) dengan polynomial *irreducible* pembangkit $f(x) = x^8 + x^4 + x^3 + x + 1$ [3]. Secara garis besar alur proses enkripsi dari Algoritma Rijndael ditunjukkan dalam Gambar 3.

C. Transformasi *AddRoundKey* [3]
Transformasi ini melakukan proses XOR antara tabel *state* pada *plaintext* dengan 128 bit kunci yang sudah dibangkitkan sebelumnya.

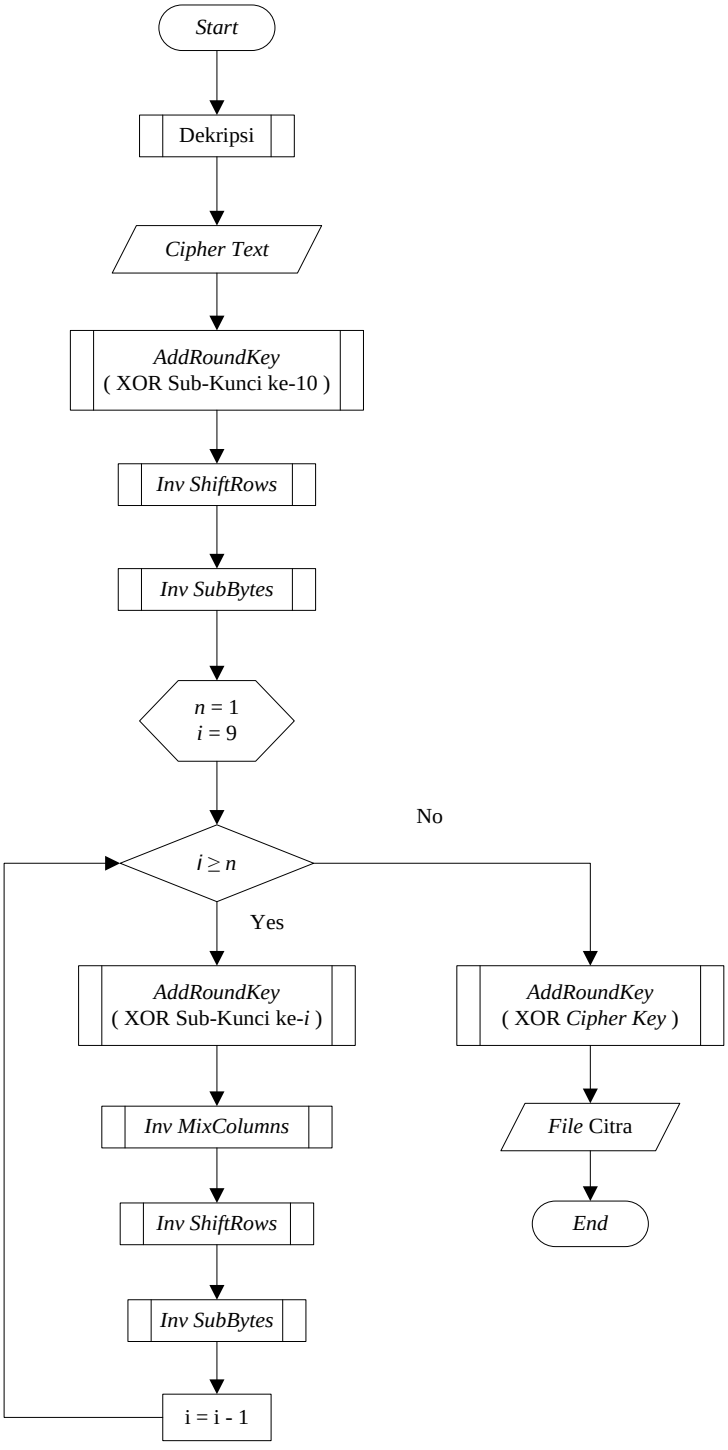
D. Transformasi *SubBytes* [3]
Transformasi *SubBytes* memetakan setiap *array state* dengan menggunakan tabel substitusi S-Box. Sebuah tabel S-Box terdiri dari 16x16 baris dan kolom dengan masing-masing berukuran 1 *byte*. Rijndael memiliki satu buah S-Box yang akan dipakai pada setiap iterasi. Cara subsitusinya adalah sebagai berikut : Setiap *byte* pada table *state* $\alpha(r,c) = xy$ denan xy adalah digit heksadesimal dari α (r,c), nilai substitusinya adalah elemen dalam S-Box yang merupakan perpotongan baris ke x dengan kolom ke y dan menghasilkan nilai substitusi baru yaitu b(r,c) [3].

E. Transformasi *ShiftRows* [3]
Transformasi *ShiftRows* akan beroperasi pada tiap baris dalam tabel *state.* Proses ini akan bekerja dengan cara memutar elemen matriks hasil proses transformasi Sub*Byte*, pada 3 baris terakhir (baris 1, 2, dan 3) ke kiri dengan jumlah perputaran yang berbeda-beda. Baris 1 akan diputar sebanyak 1 kali, baris 2 akan diputar sebanyak 2 kali dan baris 3 akan diputar sebanyak 3 kali. Sedangkan baris 0 tidak akan diputar.

F. Transformasi *MixColumns* [3]
Transformasi ini mengalikan setiap kolom dari table *state* dengan polinom $\alpha(x)$ mod $(x^4 + 1)$. Setiap kolom diperlukan sebagai polinom 4 suku pada GF (2^8). Sementara itu, polinom $\alpha(x)$ yang ditetapkan yaitu $\alpha(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$ dengan tiap konstantanya merupakan bilangan heksadesimal.

G. Proses Dekripsi [3]
Struktur proses dekripsi Rijndael secara umum sama dengan proses enkripsi, tetapi pada proses dekripsi Rijndael memiliki urutan 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. Meskipun proses pembentukan *key schedule* pada proses dekripsi dan enkripsi identik, akan tetapi proses penjadwalan penggunaan kunci pada setiap iterasi pada dekripsi berkebalikan dengan proses enkripsi. Penjadwalan kunci pada proses dekripsi pada tiap iterasi dimulai dari word ke-43 sampai word ke-0 atau dimulai dari sub-kunci ke 10 sampai *cipher key*. Secara garis besar alur proses dekripsi dari Algoritma Rijndael ditunjukkan dalam gambar 4.

H. Transformasi *AddRoundKey* [3]
Invers atau kebalikan dari transformasi *AddRoundKey* adalah proses XOR antara *state* 128 bit pada *ciphertext* dengan 128 bit *round-key* yang dibangkitkan sebelumnya dengan menggunakan kunci tiap iterasi yang berkebalikan dari proses enkripsi.

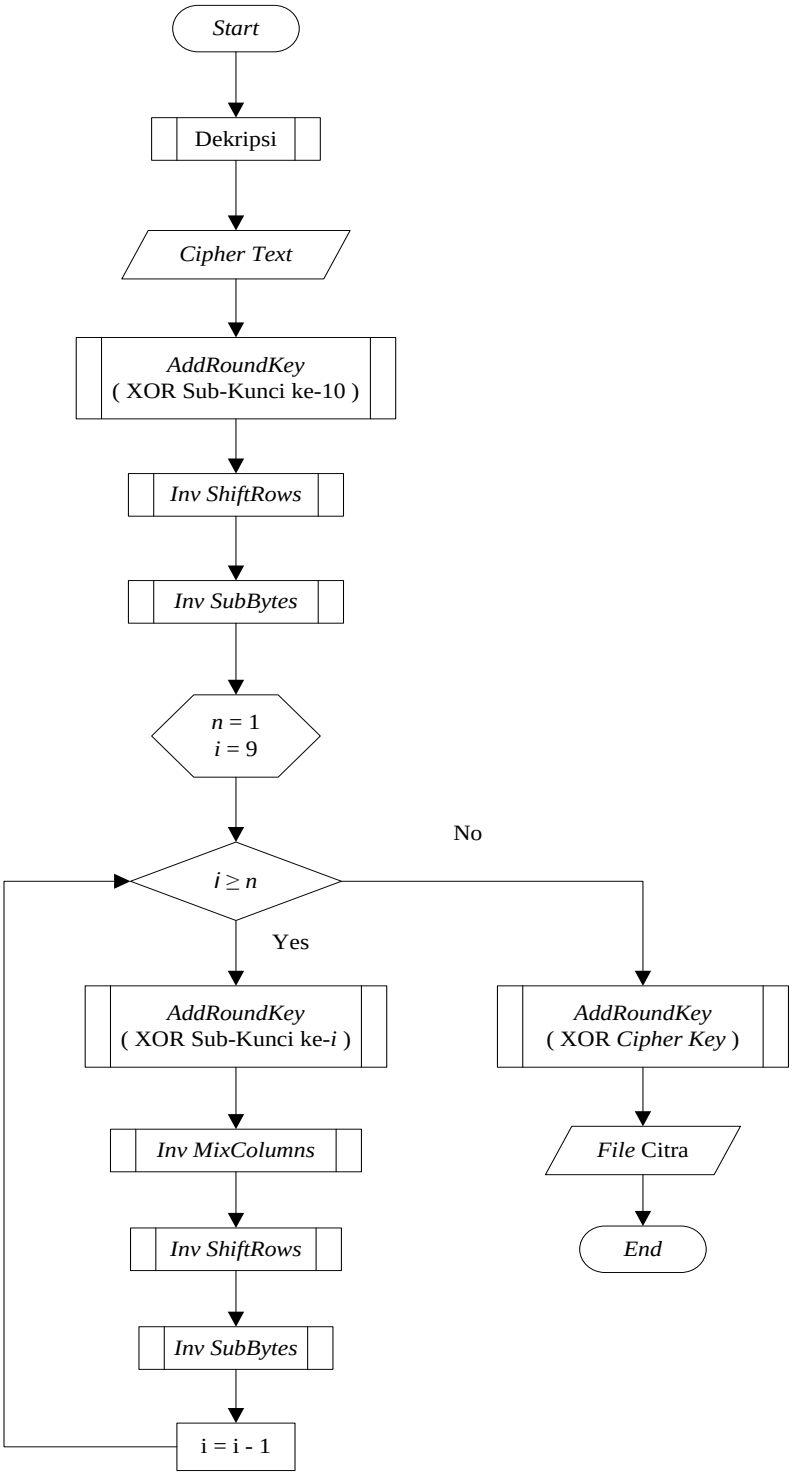


Gambar 3 *Flowchart* Proses Enkripsi Algoritma Rijndael

I. Transformasi *InvShiftRows* [3]
Invers dari transformasi *ShiftRows* juga memutar *byte-byte* pada 3 baris terakhir dengan jumlah putaran yang sama hanya saja dengan arah kebalikannya yaitu ke kanan.

J. Transformasi *InvSubBytes* [3]
Invers dari transformasi *SubByte* adalah substitusi yang menggunakan tabel *invers* S-Box.

K. Transformasi *InvMixColumns* [3]
Invers dari transformasi *MixColumns* adalah mengalikan setiap kolom dari tabel *state* pada 16 *byte ciphertext* dengan polinom b(x) mod ($x^4 + 1$). Setiap kolom diperlakukan sebagai polinom 4 suku pada GF(2^8). Sementara itu, Polinom b(x) yang ditetapkan yaitu b(x) = {0B}x^3 + {0D}x^2 + {09}x + {0E} dengan tiap konstantanya merupakan bilangan heksadesimal.



Gambar 4 *Flowchart* Proses Dekripsi Algoritma Rijndael

3. HASIL DAN PEMBAHASAN

Prosedur uji coba program menjelaskan tentang cara pengoperasian program serta tahap-tahap yang perlu dilakukan *user* untuk menjalani aplikasi enkripsi dan dekripsi *file* citra ini.

Tahap awal yang harus dilakukan *user* untuk menjalankan aplikasi ini adalah dengan cara mengeksekusi aplikasi sehingga tampil menu utama yang terdiri dari beberapa menu pilihan. Halaman tampilan menu utama aplikasi merupakan tampilan dimana untuk mulai menggunakan aplikasi. Pada saat *user* mengklik tombol menu enkripsi, maka akan masuk pada tampilan menu enkripsi. Kemudian *user* memilih *browse* maka akan muncul *open dialog file* gambar yang digunakan untuk memilih *file* gambar pada *drive* penyimpanan dan kemudian menampilkan gambar asli dalam *picture box* pada aplikasi yang telah disediakan. Setelah gambar asli dipilih maka akan tampil *original image*. Kemudian masukan *key* dan klik tombol enkripsi maka citra asli akan menjadi *file .enk* yang tidak bisa dibuka atau dilihat. *File* inilah yang merupakan *file* hasil enkripsi citra asli yang akan tersimpan secara otomatis pada *drive* penyimpanan selokasi dengan citra asli. Setelah itu *user* dapat mengklik tombol hasil enkripsi untuk melihat *file* hasil enkripsi dengan memilih nama *file* hasil enkripsi tersebut. Pada saat *user* mengklik menu dekripsi, maka akan masuk ke menu dekripsi, menu ini dipilih untuk mengubah *file* enkripsi menjadi citra asli.

User memilih *browse* maka akan muncul *open dialog file* enkripsi yang digunakan untuk memilih *file* enkripsi pada *file drive* penyimpanan komputer dan hanya menampilkan nama *file* yang ber*extention .enk* saja. Setelah *file* enkripsi dipilih maka akan tampil nama *file* enkripsi tersebut dan tombol dekripsi akan aktif. Kemudian masukan *key* yang sama dengan proses enkripsi dan klik tombol dekripsi maka *file* enkripsi akan menjadi *file .jpg*. *User* dapat mengklik tombol hasil dekripsi untuk melihat *file* hasil dekripsi dengan memilih nama *file* gambar, dimana file hasil dekripsi diberi penambahan nama di belakang nama file yaitu 01, 02 dan seterusnya.

4. UJI COBA DAN ANALISIS PENGUJIAN

Pada implementasi algoritma Rijndael dalam mengamankan data citra digital, dilakukan beberapa uji coba dengan menggunakan beberapa variasi jenis kunci input. Ukuran kunci sepanjang 128 bit (16 karakter) dan divariasikan dengan menggunakan berbagai jenis karakter baik angka, huruf, simbol, maupun gabungan dari ketiganya.

Pada tahap uji coba ini akan dilakukan dengan 2 skenario yaitu skenario pertama adalah dilakukan pengujian dengan berbagai macam ukuran gambar atau citra dimana hasil dari pengujiannya ditunjukkan dari 100 buah citra yang diuji untuk dienkripsikan dan didekripsikan kembali dapat tetap menunjukkan keutuhan datanya berupa ukuran piksel dan memory citra yang dienkripsi, tidak mengubah citra asli, didapatkan hasil bahwa

$$\text{Nilai akurasi} = \frac{100}{100} \text{ x } 100\% = 100\%$$

sedangkan skenario kedua adalah dengan berbagai macam warna yaitu citra *black white,* citra *grayscale* dan citra RGB dimana hasil dari pengujiannya ditunjukkan

$$\text{Nilai akurasi} = \frac{100}{100} \text{ x } 100\% = 100\%$$

Dari nilai akurasi yang diperlihatkan menunjukkan secara keseluruhan bahwa teknik enkripsi dan dekripsi menggunakan algoritma Rijndael lebih baik, dimana proses enkripsi dan dekripsi yang dilakukan terhadap *file* citra tersebut tidak mempengaruhi ukuran piksel dan ukuran memori dari *file* citra yang asli. Hal ini dikarenakan *file* citra hasil enkripsi memiliki ukuran memori yang sama atau sesuai dengan *file* citra asli sehingga setelah dilakukan proses dekripsi *file* citra bisa kembali seperti semula tanpa mengubah ukuran piksel dan ukuran memori dari *file* citra asli, tetapi

pengujian menunjukkan perbedaan yang cukup signifikan untuk *running time* dengan ukuran piksel yang berbeda, makin besar ukuran piksel, maka makin besar pula *running time* yang dibutuhkan. Dalam kasus variasi jenis kunci *input* juga tidak mempengaruhi ukuran piksel dan memori serta tidak ditemukan perbedaan *running time* yang signifikan, karena semua angka, huruf, simbol dan gabungan ketiganya akan diubah menjadi bilangan ASCII, sehingga tidak terjadi perbedaan yang besar dari segi *running time* ketika menggunakan variasi jenis kunci.

5. KESIMPULAN

Algoritma Rijndael dapat diimplementasikan atau diterapkan dalam teknik enkripsi dan dekripsi pada citra digital berbasis java SE. Variasi jenis kunci input (angka saja, huruf saja, simbol saja dan gabungan ketiganya) yang digunakan untuk proses enkripsi dan dekripsi tidak mempengaruhi ukuran ukuran piksel dan ukuran memori *file* citra asli atau *file* tidak mengalami perubahan ukuran piksel dan ukuran memori, sehingga integritas citra terpenuhi. Algoritma Rijndael dapat melakukan proses enkripsi dan dekripsi pada bermacam macam citra seperti citra RGB, citra *grayscale* dan citra *black white.*

6. SARAN

Ukuran kunci yang digunakan dalam teknik enkripsi dan dekripsi untuk selanjutnya dapat dicoba dengan ukuran yang beragam yakni 192 bit dan 256 bit. Untuk pengembangan selanjutnya, dapat mengaplikasikan algoritma Rijndael dalam mengamankan data selain citra digital, seperti audio dan video.

DAFTAR PUSTAKA

[1] Ariyus, Dony.2008. *Pengantar Ilmu KRIPTOGRAFI.* Yogyakarta, Penerbit: Andi.

[2] Federal Information Processing Standards Publication 197, 2001. "Anouncing the ADVANCED ENCRYPTION STANDARD (AES)" hal. 1-51.

[3] Joan Daemen & Vincent Rijmen. A Specification for Rijndael, the AES Algorithm.

[4] Munir, Rinaldi. 2006. *Kriptografi*. Bandung: Informatika Bandung.

[5] *Perangkat Lunak (Terstruktur dan Berorientasi Objek)*. Modula, Bandung.

[6] Shalahuddin, Muhammad dan Rosa A.S. 2011. *Modul Pembelajaran Rekayasa.*

[10] Sutoyo. T. et al. 2009. *Teori Pengolahan Citra Digital*, Yogyakarta. Penerbit: Andi