ISSN 2540 – 7902

KRIPTOGRAFI AES MODE CBC PADA CITRA DIGITAL BERBASIS ANDROID

Henry[*,1], Awang Harsa Kridalaksana[2], Zainal Arifin[3]

[1,2,3]Program Studi Ilmu Komputer, FKTI Universitas Mulawarman
Kampus Gunung Kelua, Samarinda, 75123 Kalimantan Timur
Email : sancini2@gmail.com[1], awanghk@unmul.ac.id[2], zainal.arifin@unmul.ac.id[3]

ABSTRAK

Keamanan merupakan salah satu aspek penting dalam pengiriman data maupun komunikasi melalui jaringan, citra digital merupakan salah satu data yang mengandung banyak informasi, oleh karena itu untuk menjaga keamanan dan kerahasiaan suatu data maupun informasi salah satunya dengan menggunakan teknik kriptografi. Kriptografi adalah suatu ilmu dan seni untuk menjaga keamanan data atau pesan. Salah satu algoritma yang sering di gunakan saat ini adalah AES (Advanced Encryption Standard). Penerapan kriptografi pada penelitian ini menggunakan AES dengan panjang kunci 128 bit yang beroperasi pada mode operasi blok CBC(Cipher Block Chaining) dan aturan padding PKCS#7 (Public – Key Cryptography Standards) agar dapat menjaga ukuran masukan plainteks tetap berada ukuran bloknya. Android merupakan operasi sistem mobile terbaru dari google yang sedang populer saat ini, karena itu banyak serangan menargetkan smartphone berbasis Android. Penelitian ini bertujuan untuk membuat suatu perangkat lunak pada smartphone android menggunakan algoritma kriptografi AES yang bertujuan untuk mengamankan data citra digital dengan mengubah isi element penyusun pada tiap piksel agar informasi yang terkandung tidak diketahui oleh orang lain. Hasil dari penelitian yaitu menghasilkan sebuah perangkat lunak pada smartphone berbasis android yang dapat digunakan untuk mengamankan informasi citra digital dan mengembalikan informasi citra digital yang teracak menjadi informasi yang dapat dikenali.

Kata Kunci : Kriptografi, AES, CBC, PKCS, Citra Digital, Android

1. PENDAHULUAN

Pentingnya menjaga kerahasiaan suatu informasi membuat ilmu kriptografi digunakan untuk mengamankan berbagai data, baik data informasi secara umum maupun data multimedia seperti data citra pada khususnya. Tidak semua citra dibuat untuk konsumsi masyarakat umum. Banyak dari citra tersebut bersifat pribadi hanya ditujukan untuk kelompok atau masyarakat tertentu. Oleh karena itu berbagai cara dilakukan masyarakat untuk mendapatkan informasi yang terdapat pada citra tersebut.

Teknologi baru smartphone merupakan media yang sangat populer pada saat ini. Smartphone memiliki banyak kegunaan yang dapat memudahkan aktifitas dewasa ini, kegunaan smartphone diantaranya dapat melakukan pertukaran data dengan mudah melalui jaringan dimana dan kapan saja melalui fitur kirim data yang dapat dilakukan melalui smartphone sangat bervariasi diantaranya melalui email, media sosial , dan masih banyak lainnya.

Berdasarkan penelitian sebelumnya yang telash dipublikasikan melalui jurnal yang diterbitkan pada tahun 2012 oleh R. Kristoforus and Stefanus Aditya dalam Seminar Nasional Aplikasi Teknologi Informasi (SNATI) dengan judul "Implementasi Algoritma Rijndael Untuk Enkripsi dan Dekripsi pada Citra Digital". Dalam jurnal yang diterbitkan oleh R. Kristoforus and Stefanus Aditya kesimpulan yang dihasilkan yaitu, perancangan program aplikasi kriptografi dengan menggunakan algoritma AES dapat memberikan keamanan citra digital atau informasi baik pada saat disimpan pada komputer maupun saat ditransmisikan dalam jaringan komputer [1].

Berdasarkan uraian sebelumnya penulis bermaksud untuk membangun sebuah sistem berbasis Android yang dapat melakukan pengamanan citra digital menggunakan metode enkripsi AES pada mode operasi blok CBC (Cipher Block Chaining). Dimana kriptografi yang digunakan mampu mengacak nilai piksel penyusun citra, sehingga keamanan data citra yang bersifat pribadi akan tetap aman saat disimpan maupun ditransmisikan melalui jaringan dan perangkat lunak yang berbasis android pada smartphone akan sangat membantu mengamanan data citra kapanpun dibutuhkan.

2. TINJAUAN PUSTAKA

2.1. Citra Digital

Citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda[2]. Citra dapat dikelompokan menjadi citra tampak (foto, lukisan dll) dan citra tak tampak (citra digital). Diantara jenis-jenis citra tersebut, hanya citra digital yang dapat diolah menggunakan komputer. Setiap citra mempunyai beberapa karakteristik, antara lain ukuran citra, resolusi, dan format nilainya. Umumnya citra berbentuk persegi yang memiliki lebar dan tinggi tertentu.

*Corresponding Authors
Email : sancini2@gmail.com

Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (Red, Green, Blue - RGB) [2].

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang-nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak 256 x 256 x 256 = 1677726 jenis warna. Dalam komputer, nilai-nilai komponen sering disimpan sebagai angka integer antara 0 sampai 255, kisaran yang dapat ditampung sebuah byte (8-bit) nilai ini dapat dituliskan dalam angka desimal maupun heksadesimal seperti yang di tunjukan pada gambar 1.

Gambar 1. HexTriplet

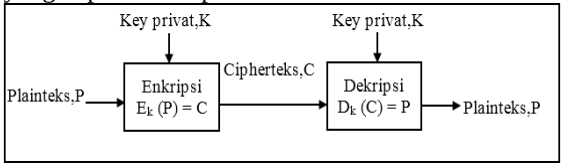
2.2. Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan [3]. Sedangkan Menezes menyatakan bahwa kriptografi adalah ilmu yang mempelajari teknik – teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, dan otentikasi [4]. Jadi dapat disimpulkan bahwa kriptografi adalah ilmu dan seni yang mempelajari teknik – teknik matematika untuk menjaga keamanan data.

Sistem kriptografi (cryptosistem) adalah kumpulan yang terdiri dari algoritma kriptografi dengan semua plaintext, ciphertext, dan kunci yang mungkin [3]. Sistem kriptografi dapat diklasifikasikan ke dalam 3 dimensi yang berbeda [5], yaitu :

a. Tipe operasi yang digunakan untuk mentransformasikan plaintext ke ciphertext.
b. Jumlah kunci yang digunakan.
c. Cara memproses plaintext.

Dalam perkembangannya ada dua jenis algoritma kriptografi, yaitu algoritma enkripsi kunci simetris (symmetric-key encryption algorthm) dan algoritma enkripsi kunci public (public-key encryption algorthm)[5]. Algoritma kunci simetris yang dapat dilihat pada Gambar 2.



Gambar 2. Enkripsi Kunci Simestris

2.3. AES (Advanced Encryption Standard)

AES merupakan algoritma Rijndael yang ditemukan oleh Dr.Vincent Rijmen dan Dr.Joan Daemen merupakan algoritma simetri dan cipher blok. Dengan demikian algoritma ini menggunakan kunci yang sama pada saat enkripsi dan deskripsi serta input dan outputnya berupa blok dengan jumlah bit tertentu. Algoritma Rijndael ditetapkan oleh NIST (National Institute of Standards and Technology) sebagai AES (Advanced Encryption Standard) pada bulan Oktober 2000.

Algoritma AES mendukung berbagai variasi kunci yang digunakan, namun algoritma AES mempunyai ukuran kunci yang tetap sebesar 128, 192, 256 bit [6]. Ukuran blok untuk algoritma AES adalah 128 bit atau 16 byte. Jumlah iterasi dalam proses enkripsi dan dekripsi dipengaruhi oleh ukuran kunci yang akan dipakai.

Tabel 1. Tiga versi AES

Versi AES	Panjang kunci (Nk words)	Ukuran Blok (Nb words)	Jumlah Putaran (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Algoritma AES menggunakan substitusi, permutasi, dan sejumlah putaran yang dikenakan pada tiap blok yang akan dienkripsi/dekripsi. Untuk setiap putarannya, AES menggunakan kunci yang berbeda. Kunci setiap putaran disebut round key. Tetapi tidak seperti DES yang berorientasi bit, AES beroperasi dalam orientasi byte sehingga memungkinkan untuk implementasi algoritma yang efisien ke dalam software dan hardware. Ukuran blok untuk algoritma AES adalah 128 bit (16 byte).

CBC (Cipher Block Chaining)

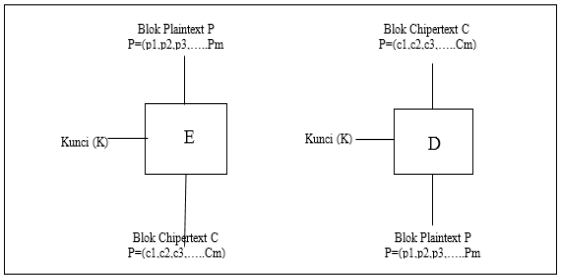
Pada cipher blok, rangkaian bit-bit plainteks dibagi menjadi blok-blok bit dengan panjang sama. Enkripsi dilakukan terhadap blok bit plaintext menggunakan bit-bit kunci (yang ukurannya sama dengan blok plaintext). Algoritma enkripsi menghasilkan blok ciphertext yang berukuran sama dengan blok plaintext. Dekripsi dilakukan dengan cara yang serupa seperti enkripsi. Misalkan blok plaintext (P) yang berukuran m bit dinyatakan sebagai vector [6].

P = (p1, p2, ..., pm) (1)

Bila plaintext dibagi menjadi n buah blok, barisan blok-blok plaintext dinyatakan sebagai

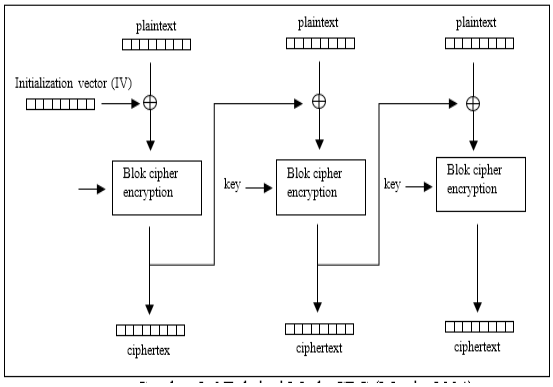
(P1, P2, …, Pn) (2)

Skema enkripsi dan dekripsi dengan cipher blok dapat dilihat pada gambar 3.



Gambar 3. Enkripsi Dekripsi cipher block

Mode operasi CBC ditemukan oleh IBM pada tahun 1976. Pada mode ini, tiap blok dari plainteks dilakukan XOR dengan hasil cipherteks dari blok sebelumnya yang kemudian dilakukan enkripsi. Dengan cara ini, tiap ciphertext dari masing-masing blok akan tergantung pada seluruh hasil cipherteks dari blok-blok sebelumnya. Selain itu, untuk membuat tiap pesan menjadi unik, digunakan IV (Initialization Vector) untuk dilakukan XOR dengan blok pertama.

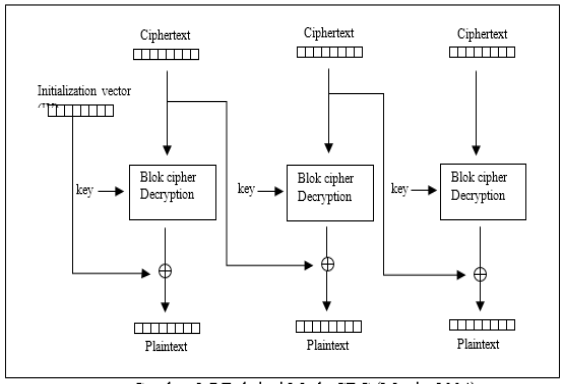


Gambar 4 Enkripsi Mode CBC

Gambar 4 merupakan proses enkripsi pada mode CBC, dimana pada proses enkripsi dilakukan secara sekuensial dari blok data pertama hingga blok data terakhir. Dimana pada awal operasi enkrips blok data akan di XOR dengan IV (Initialization Vector).

Jika blok pertama memiliki indeks 1, maka rumus matematis untuk enkripsi pada mode CBC adalah :

$$C_i = E_k(P_1 \oplus C_{i-1}), \quad C_0=\text{IV} \qquad (3)$$



Gambar 5 Dekripsi Mode CBC

sedangkan rumus matematis untuk dekripsi pada mode CBC adalah :

$$C_i = D_k(C_i) \oplus C_{i-1}, \; C_0=\text{IV} \qquad (4)$$

Dimana :

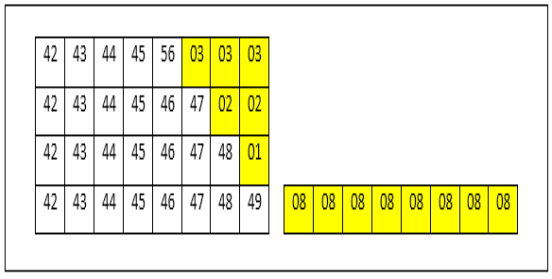
C_i : Ciphertext pada blok i
P_i : Plaintext pada blok i
E_k (...) : Fungsi enkripsi yang digunakan
D_k (...) : Fungsi dekripsi yang digunakan
IV : Initialization Vector

Yang dalam hal ini, C_0 = IV (initialization vector). IV dapat diberikan oleh pengguna atau dibangkitkan secara acak oleh program. Jadi, untuk menghasilkan blok ciphertext pertama (C_1), IV digunakan untuk menggantikan blok ciphertext sebelumnya, C_0. Sebaliknya pada dekripsi, blok plainteks diperoleh dengan cara meng-XOR-kan IV dengan hasil dekripsi terhadap blok ciphertext pertama. Pada mode CBC, blok plaintext yang sama menghasilkan blok ciphertext yang berbeda hanya jika blok-blok plaintext sebelumnya berbeda.

Padding

Dalam block-cipher plaintext dan ciphertext harus dipotong-potong dan disusun dalam blok-blok data berukuran sama[5]. Sebagai contoh, DES dan Blowfist menggunakan blok berukuran 64 bit, AES menggunakan blok berukuran 128 bit. Karena data harus masuk dalam blok berukuran sama, maka dibutuhkan padding byte sebagai pengganjal untuk menggenapi data agar pas dengan ukuran blok.

Aturan mengenai padding dijelaskan dalam standar PKCS#7 dan PKCS#5 (Public Key Cryptographic Standard). Padding dilakukan dengan mengisi byte bernilai N bila dibutuhkan padding sebanyak N byte. Sebagai contoh, bila dibutuhkan padding 3 byte, maka padding berisi ’03 03 03′, bila dibutuhkan padding 5 byte, maka padding berisi ’05 05 05 05 05′.



Gambar 6 Contoh Padding

Dalam standar PKCS memang sudah diatur bahwa padding harus ditambahkan pada semua data, walaupun data tersebut sudah genap seukuran blok yang diperlukan.

2.4. Algoritma SHA(Secure Hash Algorithm)

Fungsi hash SHA (Secure Hash Algorithm), antara lain SHA-1, SHA-224, SHA-256, SHA-384 dan SHA-512 adalah lima fungsi hash kriptografis yang dibuat oleh National Security Agency (NSA) dan dinyatakan sebagai standar keamananan pemerintah USA [7]. SHA-1 digunakan dalam

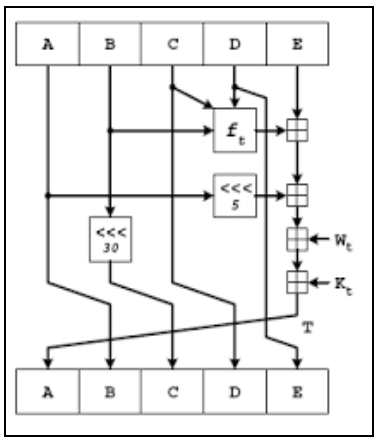
proses sekuriti banyak program, seperti TLS and SSL, PGP, SSH, S/MIME dan IPsec. Fungsi ini dianggap sebagai pengganti fungsi hash MD5 yang lebih sering digunakan publik. Tetapi pada kenyataannya kedua fungsi ini tetap digunakan sebagai proses enkripsi.

Sedangkan 4 varian lainnya (SHA-224, SHA-256, SHA-384 dan SHA- 512) biasanya disebut sebagai SHA-2. Hingga saat ini belum ada serangan terhadap SHA-2, tetapi karena kemiripannya dengan SHA-1, para peneliti khawatir dan mengembangkan kandidat baru penggantinya. Kelima algoritma SHA ini memiliki ukuran pesan, blok, kata, dan pesan digit yang berbeda dimana ukuran-ukuran dari masing-masing algoritma SHA tersebut dapat dilihat pada tabel berikut.

Tabel 2 Massage Diggest Pada SHA

Algorithm	Massage Size (bits)	Block Size (bits)	Word Size (bits)	Massege Diggest Size (bits)
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$<2^{128}$	1024	64	384
SHA-512	$<2^{128}$	1024	64	512

SHA-1 adalah pengembangan dari SHA-0 dimana SHA-1 memperbaiki kelemahan yang ada di SHA-0. SHA-1 merupakan fungsi hash yang paling popular dibandingkan dengan fungsi hash SHA lainnya. SHA-1 memproduksi 160 bit digest berdasarkan prinsip yang sama dengan algoritma MD4 dan MD5 namun dengan design yang berbeda. SHA-1 mempunyai kapasitas input message 2^{64}-1, dengan hasil hash 160 bits dan evaluasi kekuatan hash 2^{80} misal SHA-1 digunakan untuk meng-hash sebuah pesan M, yang mempunyai panjang maksimum 2^{64}-1 bits. Algoritma ini menggunakan urutan dari 80 kali 32-bit kata, dengan menggunakan 5 variabel yang menampung 32 bits per variabel, dan hasil hashnya.



Gambar 7 Analogi Hash Pada SHA – 1

3. HASIL DAN PEMBAHASAN

3.1 Rancangan Sistem

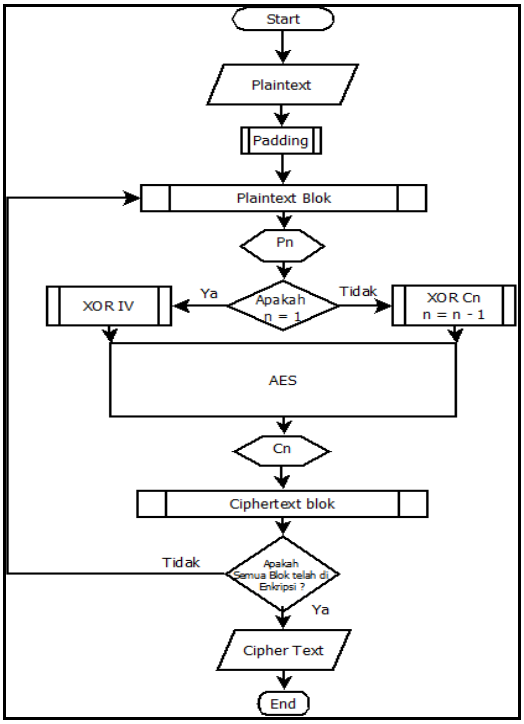
Sistem kriptografi yang dibangun bersifat sangat fleksible dalam menerima kunci masukan dari pengguna, dimana sistem akan memproses kunci masukan dari pengguna sebelum diteruskan ke algoritma AES. Proses ini meliputi proses pengacakan kunci masukan pengguna menggunakan algoritma SHA-1 yang selanjutnya akan dilanjutkan pada pemotongan panjang kunci yang dihasilkan algoritma SHA-1 sebnayak 32bit pada bagian akhir hasil karena hasil dari fungsi SHA-1 akan selalu bernilai 160bit. Algoritma AES yang digunakan hanya menerima kunci masukan sepanjang 128 bit sehingga hasil dari fungsi hash hanya akan diambil sebanyak 128 bit, yang kemudian akan digunakan sebagai kunci utama unutk membangun sub kunci pada proses enkripsi algoritma AES.

Algoritma enkripsi yang digunakan dalam sistem ini adalah algoritma AES dan beroperasi pada mode operasi CBC (Cipher Block Chaining), Dengan menggunakan mode CBC, setiap blok cipherteks bergantung tidak hanya pada kunci dan blok plainteksnya tetapi juga pada seluruh blok plainteks sebelumnya , sehingga pengacakan data terhadap plainteks akan menghasilkan cipherteks yang benar-benar berbeda .

Algoritma AES hanya dapat menerima masukan blok plainteks sebesar 128 bit atau kelipatan nya, sehingga untuk dapat menerima masukan blok plainteks yang kurang dari 128 bit penulis menggunakan algoritma padding PKCS7 untuk mengisi blok plainteks terakhir yang tidak genap 128 bit , namun dengan menggunakan padding ini walaupun jumlah data pada blok plainteks sudah lengkap, blok plainteks akan tetap ditambah padding sebesar 1 blok plainteks atau bernilai 128 bit.

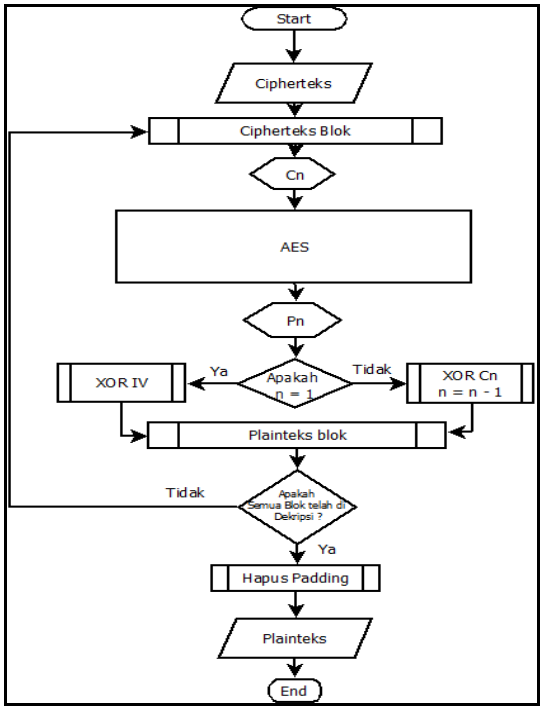
Plainteks masukan yang berupa nilai red green blue dari citra digital akan di susun dalam sebuah array yang kemudian sistem akan menghitung panjang data dan melanjutkannya ke proses padding dan akan menghasilkan blok – blok plainteks sebesar 128 bit atau kelipatan nya, kemudian blok plainteks ini akan di XOR dengan blok cipherteks sebelumnya apabila plainteks yang akan dienkripsi merupakan blok pertama , blok plainteks tersebut akan di XOR dengan IV (initialization vector), yang selanjutnya akan dilanjutkan ke proses enkripsi menggunakan proses AES.

Proses enkripsi akan mengalami proses perulangan sampai semua blok plainteks selesai di enkripsi, yang kemudian blok – blok cipherteks hasil dari enkripsi yang telah diproses akan disatukan kembali menjadi cipherteks. Agar lebih memahami proses dan tahapan yang terjadi pada sistem ini dapat dilihat pada flowchart yang digambarkan pada gambar 8 dberikut.



Gambar 8 Flowchart Enkripsi Mode CBC

Flowchart proses dekripsi mode CBC dapat di lihat pada gambar 9, pada gambar ini akan dijelaskan bahwa proses dekripsi merupakan arah balik dari proses enkripsinya dimana proses pelepasan padding terjadi pada tahap akhir, agar padding dapat dikenali oleh sistem. Kemudian menyusun blok-blok plainteks yang dihasilkan.

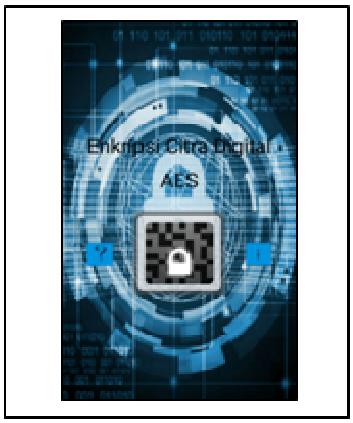


Gambar 9 Flowchart Dekripsi Mode CBC

3.2 Implementasi Sistem

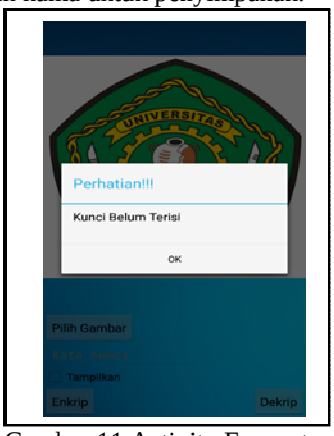
Tahapan realisasi yang dilakukan setelah rancangan aplikasi.Implementasi dilakukan untuk mengetahui hasil dari rancangan sistem yang telah dibangun. Desain sistem perangkat lunak enkripsi Citra Digital dengan menggunakan algoritma AES mode operasi CBC dengan padding PKCS7 diimplementasikan menggunakan bahasa pemrograman java berbasis Android.

Pada perangkat lunak ini terdapat 4 halamn yang dapat di akses oleh pengguna yaitu halaman Home, Help, Info dan Encryptor. Halaman awal yang akan di hadapkan pada pengguna merupakan Activity Home, activity Home merupakan halaman utama yang menghubungkan dengan menu-menu lainnya melalui image button yang dapat dilihat pada gambar 10.



.Gambar 10 Activity Home

Melalui tombol tengah dengan gambar gembok user dapat mengakses activity encryptor, pada activity encryptor ini user dapat melakukan proses pengacakan dan pengembalian data menggunakan proses enkripsi dan dekripsi, serta menyimpan data gambar menjadi file citra digital kedalam penyimpanan.Dalam activity ini setiap proses yang di lakukan akan langsung di tampilkan di image view. Activity ini memiliki beberapa alert dialog yang akan memberikan informasi tentang kesalahan pada kegiatan yg dilakukan user seperti user belum memasukan kunci, gagal dekripsi ,belum ada gambar yang terpilih, dan belum memasukan nama untuk penyimpanan.



Gambar 11 Activity Encryptor

3.3 Pengujian Sistem

Pengujian sistem pada perangkat lunak enkripsi citra digital dengan menggunakan metode AES ini dilakukan dengan menggunakn Black Box

Testing, perubahan ukuran file, uji lama waktu dan menghitung perbedaan antara citra asli dan citra hasil rekonstruksi dengan menghitung nilai PSNR dan MSE. Dengan ujicoba Black Box memungkinkan untuk melihat fungsional suatu perangkat lunak.

Black Box Testing

Pengujian Black Box Testing ini ditujukan untuk melatih keseluruhan unit fungsional dari perangkat lunak agar perangkat lunak dapat bekerja dengan baik tanpa mengalami kegagalan sistem.

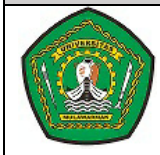
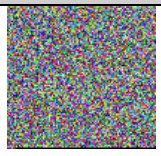
Tabel 3 Uji Black Box Testing

No	Kasus Uji	Skenario	Hasil yang Diharapkan	Keterangan
1	Memasukan gambar	User memilih gambar dari gallery sebagai masukan	System dapat menerima masukan gambar	Valid
2	Berganti tampilan image view	Melakukan proses pilih gambar/enkripsi/dekripsi	Image view dapat berganti setiap proses	Valid
3	Memasukan kunci	User memasukan kunci untuk proses enkripsi	Kunci dapat digunakan untuk enkripsi / dekripsi gambar	Valid
4	Melakukan enkripsi	Enkripsi dengan kunci yang dimasukan User	sistem dapat melakukan enkripsi dengan baik.	Valid
5	Melakukan Dekripsi	Dekripsi dengan kunci yang dimasukan User	sistem dapat melakukan dekripsi dengan baik.	Valid
6	Menyimpan Gambar	Minyimpan hasil gambar	System dapat menyimpan gambar	valid

Size Difference

Tahap size difference dilakukan agar dapat mengetahui besar nya perubahan ukuran yang terjadi setelah perangkat lunak melakukan proses enkripsi dan dekripsi. Sample uji yang digunakan dapat dilihat pada tabel 4.

Tabel 4 Sample Uji

Gambar Asli	Gambar Enkripsi	Gambar Dekripsi
		

Pengujian menggunakan satu sample gambar dengan ukuran dimensi yang berbeda kemudian mencatat ukuran tiap sample yang dihasilkan perangkat lunak setelah melewati proses enkripsi dan dekripsi. Hasil pengujian dapat dilihat pada tabel 5.

Tabel 5 Uji Size Difference

No	Size Asli	Size Enkrip	Dimensi	Size Dekrip	Dimensi
1	9,33 kb	34.6 kb	100x101	16 kb	100x100
2	28,8 kb	137 kb	200x201	51 kb	200x200
3	53,2 kb	309 kb	300x301	96,6 kb	300x300
4	79,5 kb	547 kb	400x401	146 kb	400x400
5	107 kb	855 kb	500x501	202 kb	500x500
6	135 kb	1200 kb	600x601	259 kb	600x600
7	162 kb	1630 kb	700x701	315 kb	700x700
8	188 kb	2120 kb	800x801	357 kb	800x800
9	217 kb	2680 kb	900x901	413 kb	900x900
10	246 kb	3320 kb	1000x1001	470 kb	1000x1000

Berdasarkan data hasil uji perubahan ukuran diatas dapat di ambil kesimpulan bahwa proses enkripsi akan menghasilkan :

1. Ukuran gambar enkripsi lebih besar dari gambar asli.
2. Ukuran gambar dekripsi lebih kecil jika di bandingkan gambar enkripsi.
3. Ukuran gambar dekripsi lebih besar jika di bandingkan gambar asli.
4. Jika dinilai berdasarkan ukuran gambar hasil rekonstruksi akan berbeda dengan gambar asli

Running Time

Tahap running time proses dilakukan agar dapat mengetahui lama waktu proses yang di butuhkan perangkat lunak untuk melakukan proses enkripsi dan dekripsi. Hasil uji dapat dilihat pada tabel berikut.

Table 6 Uji Waktu Emulator

No	Dimensi	Ukuran Gambar	Waktu Enkripsi	Waktu Dekripsi
1	100 x 100	9,33 kb	0,723 detik	0,619 detik
2	200 x 200	28,8 kb	1,225 detik	1,757 detik
3	300 x 300	53,2 kb	2,713 detik	2,821 detik
4	400 x 400	79,5 kb	4,717 detik	5,131 detik
5	500 x 500	107 kb	7,259 detik	7,701 detik
6	600 x 600	135 kb	10,807 detik	11,605 detik
7	700 x 700	162 kb	14,57 detik	15,368 detik
8	800 x 800	188 kb	18,928 detik	19,255 detik
9	900 x 900	217 kb	26,061 detik	25,305 detik
10	1000 x 1000	246 kb	34,238 detik	32,038 detik

Tabel di atas merupakan hasil yang didapat setelah melakukan uji coba dengan mengunnakan emulator Android Virtual Device (AVD) dengan spesifikasi CPU ARM (armeabi-v7a), sistem operasi Android 4.1.2 (Api level 16), dan kapasitas RAM 512 Mb. Berdasarkan hasil uji yang di dapat perbedaan waktu enkripsi dan dekripsi yang di butuhkan untuk memproses tidak jauh berbeda dengan nilai maksimal perbedaan 2,200 detik yang

terjadi pada uji coba dengan sample dimensi 1000 x 1000.

Tabel 7 Uji Waktu Smartphone

No	Dimensi	Ukuran Gambar	Waktu Enkripsi	Waktu Dekripsi
1	100 x 100	9,33 kb	0,018 detik	0,022 detik
2	200 x 200	28,8 kb	0,064 detik	0,075 detik
3	300 x 300	53,2 kb	0,141 detik	0,162 detik
4	400 x 400	79,5 kb	0,254 detik	0,291 detik
5	500 x 500	107 kb	0,393 detik	0,442 detik
6	600 x 600	135 kb	0,571 detik	0,649 detik
7	700 x 700	162 kb	0,772 detik	0,879 detik
8	800 x 800	188 kb	1,002 detik	1,135 detik
9	900 x 900	217 kb	1,256 detik	1,465 detik
10	1000 x 1000	246 kb	1,570 detik	1,779 detik

Tabel di atas merupakan hasil yang didapat setelah melakukan uji coba dengan mengunnakan Smartphone dengan spesifikasi CPU Octa-core 2.0 Ghz, sistem operasi Android 5.0.2 (Api level 21). Berdasarkan hasil uji yang di dapat perbedaan waktu enkripsi dan dekripsi yang di butuhkan untuk memproses tidak jauh berbeda dengan nilai maksimal perbedaan 0,209 detik yang terjadi pada uji coba dengan sample dimensi 1000 x 1000.

Berdasarkan kedua pengujian di atas dapat di ambil kesimpulan :

1. Proses waktu proses yang di butuhkan untuk enkripsi dan dekripsi tidak berbeda terlalu jauh.
2. Proses waktu proses enkripsi dan dekripsi dipengaruhi mesin yang menjalankannya.
3. Ukuran objek mempengaruhi lama waktu proses enkripsi dan dekripsi.

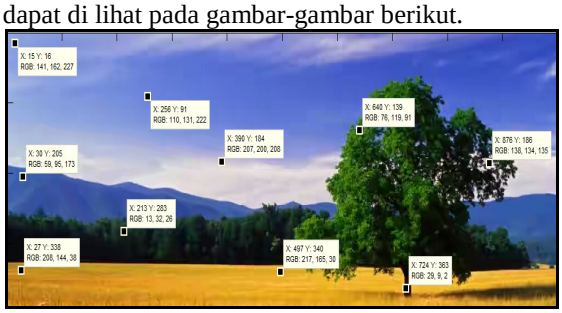
Noise Ratio

Dalam pengujian noise ratio antara gambar asli dan gambar hasil rekonstruksi penulis menggunakan 4 gambar yang berbeda, kemudian akan membanding kan nya dengan gambar hasil rekonstruksi dari masing- masing gambar tersebut. Nilai MSE akan bernilai baik jika hasil semakin mendekati angka 0, untuk penilaian MSE semakin besar nilai akan semakin baik dan akan menghasilkan nilai *infinite* (∞) jika tidak ada perbedaan antara kedua gambar.

Tabel 8 Noise Ratio

No	Gambar Asli	Gambar Rekonstruk	MSE	PSNR
1			R : 4.99 G : 3.25 B : 5.84	R : 39.33 dB G : 43.61 dB B : 37.74 dB
2			R : 2.37 G : 1.21 B : 2.30	R : 46.76 dB G : 53.75 dB B : 47.06 dB
3			R : 2.98 G : 1.50 B : 3.43	R : 44.47 dB G : 51.34 dB B : 43.06 dB
4			R : 3.87 G : 2.14 B : 5.02	R : 41.86 dB G : 47.80 dB B : 39.28 dB

Berdasarkan hasil uji di atas dapat di ambil kesimpulan bahwa terjadi perbedaan kualitas pada gambar, pada tiap unsur warna memiliki perbedaan tingkat noise yang di hasilkan. Jika di lihat pada hasil percobaa nilai rasio perbedaan yang tertinggi terdapat pada unsur warna biru (blue) namun rasio ini masih berada pada tingkat yang baik karena untuk tipe kompresi gambar lossy seperti jpeg rentang nilai PSNR yang berada pada nilai 30 – 50 dB masih pada batas tingkat noise yang tidak bisa di lihat mata manusia. Selanjutnya contoh hasil verifikasi perbedaan nilai pixel pada gambar akan di ambil 10 titik yang sama pada gambar asli dan gambar hasil rekonstruksi dan melihat perbedaan nilai Antara kedua gambar tersebut. Hasil nilai dapat di lihat pada gambar-gambar berikut.



Gambar 12 Gambar Asli

Gambar 13 Gambar Rekonstruksi

Setelah membandingkan nilai pada titik-titik pixel antara kedua gambar diatas dapat di ambil kesimpulan bahwa perubahan yang terjadi pada nilai penyusun pixel gambar rekonstruksi tidak jauh berbeda dengan gambar asli sehingga perubahan nilai tidak menyebabkan perubahan besar pada gambar.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Kesimpulan yang dapat diambil setelah menyelesaikan penulisan tentang enkripsi citra digital dengan menggunakan metode algoritma AES mode CBC serta dilakukan pengujian dari sistem yang telah dibuat adalah :

1. Penerapan kriptografi pada pengacakan data citra digital dapat mengamankan informasi.
2. Hasil dari enkripsi akan menghasilkan ukuran file yang lebih besar dari file asli.
3. Ukuran file berpengaruh pada lama waktu proses enkripsi dan dekripsi.
4. Waktu operasi yang di butuhkan juga di pengaruhi mesin yang menjalankan nya.

4.2 Saran

Berdasarkan kesimpulan yang diperoleh, untuk pengembangan lebih lanjut dari perangkat lunak enkripsi citra digital dengan menggunakan metode algoritma AES mode CBC dan, maka diberikan beberapa saran yaitu :

1. Perangkat lunak ini dapat di implementasi kan ke platform lain.
2. Perangkat lunak dapat mendukung tipe tipe kompresi gambar yang lain seperti bmp, tiff , dan lain lain.

5. DAFTAR PUSTAKA

[1] R, Kristoforus & Stefanus, A. 2012. Implementasi Algoritma Rijndael Untuk Enkripsi dan Dekripsi Pada Citra Digital. Seminar Nasional Aplikasi Teknologi Informasi.

[2] Sutoyo. T. et al. 2009. Teori Pengolahan Citra Digital. Yogyakarta: Andi

[3] Schneier, B. 1996. Applied Cryptography. John wiley & Sons.inc.

[4] Menezes, A., Van Orschot , P., & Vanstone, S. (1996). Handbook of Applied Cryptography. CRC Press.Inc.

[5] Stallings, w. 2005. Cryptography and Network Security Principles and Practices. Prentice HAll.

[6] Munir, R. 2006. Kriptografi. Bandung: Informatika.

[7] Huda, W. 2003. Perkembangan Enkripsi Fungsi Hash pada SHA (Secure Hash Algorithm). Jurnal Ilmu Komputer dan Teknologi Informasi.