INFORMATICS AND DIGITAL EXPERT (INDEX) - VOL. 4 NO. 1 (2022) 25-32
Terbit *online* pada laman web jurnal : https://e-journal.unper.ac.id/index.php/informatics
| ISSN (Print) 2775-2208 | ISSN (Online) 2715-0453 |

# Kriptografi Untuk Enkripsi Ganda Pada Gambar Menggunakan Algoritma AES (Advanced Encryption Standard) Dan RC5 (Rivest Code 5)

*Galih Yuga Pangestu[1], Asep Id Hadiana[2], Puspita Nurul Sabrina[3]*

[1]*Informatika,Fakultas Sains dan Informatika, Universitas Jenderal Achmad Yani, Jl. Terusan Sudirman, Cimahi, Indonesia*
e-mail: galihyuga123@gmail.com[1] , asep.hadiana@lecture.unjani.ac.id[2], puspita.sabrina@lecture.unjani.ac.id[3]

**INFORMASI ARTIKEL**

*Sejarah Artikel:*
Diterima Redaksi: 01 Februari 2022
Revisi Akhir: 04 Juni 2022
Diterbitkan *Online*: 02 Juli 2022

*Kata Kunci* :
Kriptografi, Gambar, AES, RC5, Enkripsi, Dekripsi. Kunci Simetris

*Korespondensi :*
Telepon / Hp : +62 8953 5180 0608
E-mail : **email@galihyuga123@gmail.com**

**A B S T R A K**

Pada masa kini, informasi berupa gambar sangatlah penting, terutama pada bidang kemiliteran. Gambar yang diproses melalui channel komunikasi militer, harus dirahasiakan sehingga data gambar menjadi aman dan tidak dapat dilihat oleh penyusup.maka, penerapan enkripsi gambar perlu diterapkan untuk menjaga kerahasiaan dan keamanan gambar tersebut. saat ini telah banyak algoritma algoritma untuk mengenkripsi gambar. Salah satu algoritma yang cukup populer adalah AES (Advanced Encryption Standard). AES merupakan algoritma kriptografi berjenis cipher blok yang terkenal luas dalam pengenkripsian sebuah data karena algoritma ini lebih baik untuk mencegah serangan brute force dibanding algoritma pendahulunya yaitu DES (Data Encryption Standard). AES ini akan diterapkan pada sebuah program yang berfungsi untuk mengamankan gambar kemiliteran agar tidak terjadi pencurian gambar oleh pihak ketiga. Gambar harus dienkripsi dahulu menggunakan sebuah kunci simetris sebelum dikirim ke penerima agar aman, dan penerima harus memiliki kunci dari pengirim agar dapat melakukan dekripsi terhadap gambar yang telah dienkripsi tersebut. Namun, hanya dengan menggunakan AES saja belum cukup untuk memberikan keamanan ekstra pada data gambar tersebut. diperlukan algoritma tambahan untuk melakukan enkripsi terhadap gambar yang telah dienkripsi menggunakan AES, sehingga gambar hasil enkripsi AES tidak dapat diakses juga. Salah satu algoritma yang cocok untuk diterapkan karena memiliki kunci simetris juga dan proses enkripsinya cukup cepat. Salah satu algoritma tambahan yang cukup cepat untuk melakukan enkripsi adalah RC5 (Rivest Code 5), yang dikembangkan oleh ron rivest untuk mengenkripsi file dengan cepat dan dengan kunci simetris. Dari hasil kedua algoritma di atas maka akan terbentuk suatu enkripsi ganda yang memberikan keamanan lebih terhadap data gambar militer. Tujuan dari peneliti menggunakan algoritma AES dan RC5 adalah agar data gambar lebih sulit untuk dipenetrasi serta hasilnya juga data yang terenkripsi lebih aman namun tetap mudah untuk dilakukan dekripsi.

## 1. PENDAHULUAN

Dikarenakan meningkatnya permintaan akan keamanan informasi, *enkripsi* dan *dekripsi* gambar telah menjadi penelitian yang penting dan memiliki prospek aplikasi yang luas. Maka dari itu bidang *enkripsi* menjadi sangat penting di era sekarang. Keamanan gambar sering sekali memiliki ancaman yang cukup serius dan perlu diperhatikan. *Enkripsi* dan *dekripsi* pada gambar telah banyak diterapkan pada bidang komunikasi kemiliteran, dll sebagainya. Agar data aman dari berbagai serangan dan untuk *integritas data* kita harus *mengenkripsi* data tersebut sebelum dikirim atau disimpan. Pada bidang kemiliteran, masing masing pasti tentunya memiliki gambar rahasia yang harus diamankan yang tidak boleh dilihat pihak ketiga atau penyusup, seperti misalnya gambar peta markas militer, gedung atau bangunan militer, peralatan perang, dll sebaginya. Sebagian besar informasi ini sekarang dikumpulkan dan disimpan di komputer elektronik dan dikirim melalui jaringan ke komputer lain. Jika gambar rahasia tersebut jatuh ke tangan yang salah, maka akan terjadi pelanggaran keamanan data yang dapat menyebabkan terjadinya deklarasi perang, peniruan benda benda kemiliteran, ancaman serangan militer, dll sebagainya. [1]

Maka dari itu telah diusulkan beberapa tehnik untuk mengkripsi gambar, salah satu metode *pengenkripsian* data yang cukup terkenal adalah kriptografi [2]. Kriptografi adalah salah satu teknik paling signifikan dan populer untuk merahasiakan data dari penyerang

atau penyusup, dengan menggunakan dua proses penyandian yang dinamakan Encryption (*Enkrips)i* dan Decryption (*Dekripsi)*. *Enkripsi* merupakan proses pengacakan dan pengubahan struktur di dalam data agar data asli tidak dapat terbaca oleh penyusup atau penyerang. Teknik ini berisi sebuah cara untuk merahasiakan isi data yang asli (*Plaintext*) ke dalam format data yang strukturnya telah diubah dan diacak dan tidak dapat dibaca oleh penyusup, yang disebut sebagai *Cipher Text* [3]. Proses selanjutnya adalah *Dekripsi*. *Dekripsi* merupakan tehnik yang memiliki sifat berkebalikan dengan enkripsi, dimana dekripsi merupakan teknik untuk mengubah kembali file yang telah diacak dan dirubah strukturnya pada tahap enkripsi tadi menjadi teks biasa kembali (*Plain text*). Kedua teknik pengamanan data tersebut memiliki berbagai macam rumus serta jenis operasi kriptografinya serta kunci dan seberapa panjang kunci yang digunakan. [3].

Pada masa ini telah banyak penelitian penelitian terdahulu mengenai penerapan kriptografi untuk meng*enkripsi* gambar. Diantaranya adalah Simulasi *Enkripsi* gambar menggunakan algoritma *AES* [4], Pengenkripsian gambar digital berbasis pada algoritma AES [5], *Enkripsi* Gambar Menggunakan Algoritma *BlowFish* pada *MatLab* [6], *Pengenkripsian* gambar menggunakan metode sistem *chaos* dan *AES* [7], Analisis *Kriptografi* pada peng*enkripsi*an gambar menggunakan autoblocking dan elektrokardiografi [8], modulasi logistik untuk pengekripsian peta [9], pengamanan teks dan gambar dengan AES [10], Evaluasi Komprehensif dari algoritma AES, DES, RSA, 3DES, BLOWFISH [11], pengamanan citra medis menggunakan edge maps [12], pengenkripsian gambar menggunakan urutan operasi DNA berbasis novel chaos [13], pengamanan isi file berjenis word menggunakan algoritma RC5 [14], dan pengamanan data gambar menggunakan algoritma RC5 pada sistem berbasis androud.[15].

Algoritma diatas memang cukup populer pada dunia *Kriptografi* dan memiliki masing masing kekurangan tertentu. Namun yang paling baik dalam *pengenkripsian* gambar adalah algoritma *AES*, karena *AES* merupakan algoritma paling aman dari serangan brute force [5], walaupun memiliki waktu *enkripsi* yang cukup lama. Dan aplikasi untuk membuat program peng*enkripsi*an sering kali dibuat dalam Bahasa pemrograman populer seperti java, visual basic, dll sebagainya.

Namun dengan menggunakan *AES* saja tidak cukup untuk memberikan keamanan ekstra pada gambar militer tersebut. Gambar militer yang telah *dienkripsi* tersebut perlu dilakukan *enkripsi* lagi agar penyerang tidak dapat membuka file gambar yang telah di*enkripsi* menggunakan algoritma *AES.* Salah satu algoritma yang cukup baik untuk mengkripsi file adalah *RC5*. *RC5* adalah algoritma kriptografi yang cukup populer juga karena berjenis block cipher yang simetris yang dapat diimplementasikan pada perangkat lunak dan perangkat keras. [16]. Selain itu algoritma *RC5* digunakan karena proses enkripsinya yang cukup cepat dan tidak terlalu banyak memakan *memory* [17] Maka dari itu *Kriptografi* ganda akan memberikan keamanan ekstra yang lebih aman.

Penelitian ini mengusulkan untuk pembuatan sistem *Kriptografi enkripsi* gambar menggunakan algoritma *AES* dan *RC5* untuk mengamankan data kemiliteran sehingga data tidak dapat dilihat dan dibaca oleh penyusup. Selain itu data yang telah dienkripsi harus aman dari berbagai macam serangan, seperti halnya brute force, yang cara kerjanya adalah melakukan berbagai kombinasi serangan penebakan sampai kunci untuk dekripsi nya ditemukan [18].

## 2. LANDASAN TEORI

### 2.1. Kriptografi

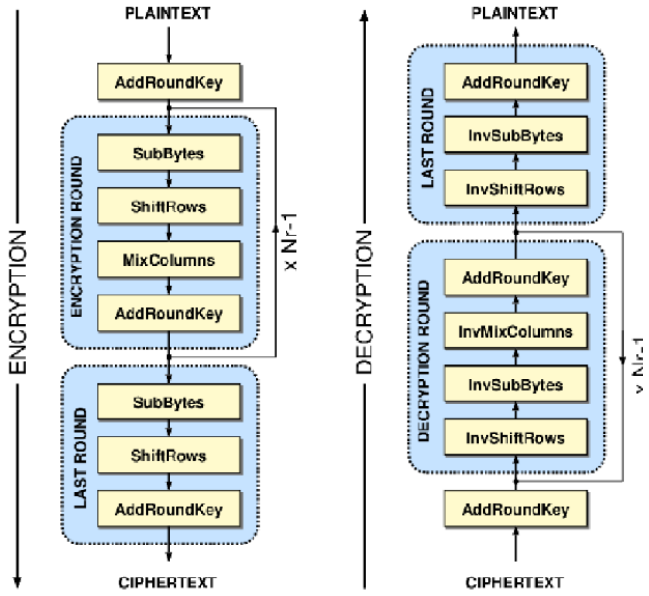
Dalam ilmu matematika, Kriptografi adaklah sebuah ilmu atau teknik untuk merahasiakan atau menyembunyikan pesan atau data agar tidak dapat terbaca atau terlihat, singkatnya kriptografi merupakan tehnik menyembunyikan data sebelum dikirim ke penerima data.[2]. Kriptografi memiliki dua teknik utama dalam prosesnya, yaitu enkripsi dan dekripsi, dimana enkripsi artinya adalah mengacak dan mengubah struktur dan di dalam pesan atau data sehingga tidak dapat terbaca atau ditak dapat dilihat isi pesan yang asli, dan dekripsi artinya mengubah pesan yang tidak terbaca menjadi pesan yang dapat dibaca. Kedua proses tersebut memerlukan sebuah kunci agar dapat melakukan masing masing prosesnya. [2]

### 2.2. Kriptografi Kunci Simetris

Kriptografi kunci simetris adalah salah satu jenis algoritma kriptografi yang tehnik untuk melakukan enkripsi dan dekripsinya sama, oleh karena itu kunci untuk melakukan proses enkripsi dan dekripsi nya juga sama. Lain halnya dengan kriptografi kunci asimetris dimana kunci untuk melakukan enkripsi dan dekripsinya berbeda atau tidak sama. Algoritma kriptografi dengan kunci simetris saat ini masih sering digunakan karena proses enkripsi dan dekripsinya cukup cepat dan kuncinya dapat mudah diingat oleh pengguna. [18]

Berbanding terbalik dengan algoritma enkripsi yang jenis kuncinya merupakan kunci asimetris dimana kunci untuk melakukan enkripsi dan dekripsinya berbeda, sehingga pengguna harus menyimpan kedua kunci untuk melakukan enkripsi dan dekripsi. Jika salah satu kunci tersebut hilang maka pengguna tidak dapat mengakses atau mengelola file yang terenkripsi tersebut. Namun untuk kriptografi kunci simetris juga dari sisi keamanannya juga harus diperhatikan agar keamanannya tidak kalah dibandingkan dengan HAKI yang dilakukan oleh penulis naskah, berikut algoritma kriptografi dengan kunci asimetris, seperti contoh semisal memilik kunci yang cukup Panjang.

### 2.3. Algoritma AES




Gambar 2.1. Enkripsi dan Dekripsi AES

*AES* standar adalah *enkripsi* tingkat lanjut yang telah diperkenalkan pada tahun 2000 oleh NIST. Panjang data dalam *AES* adalah 128 bit, yaitu 16 byte. Namun, kuncinya bisa memperoleh panjang yang berbeda (misalnya, 128 bit, 192 bit, 256 bit). *AES* memiliki 10, 12 dan 14 putaran untuk kunci 128-bit, 192-bit dan 256-bit, masing-masing. Gambar 2.1 menunjukkan blok tersebut diagram algoritma *AES*.*AES* memiliki empat blok operasional utama:

- Subbytes: S-box digunakan untuk menggantikan setiap blok databyte dengan blok lain.
- Shiftrows: Setiap baris dari matriks status diberikan pergeseran siklikke sisi kanan sesuai dengan lokasinya.
- MixColumns: Merupakan operasi perkalian matriks dimana setiap kolom dari matriks status dikalikan dengan matriks tetap.
- AddRoundKey: Operasi XOR dilakukan antara matriks state yang baru dan kunci bulat

Kunci simetris *AES* terbagi dalam kelompok, ada tiga jenis panjang kunci dalam cara *enkripsi* ini: 128 bit, 196bit dan 256 bit, ukuran paket semuanya 128 bit, seluruh algoritmanya memiliki fleksibilitas yang baik, sehingga banyak digunakan dalam perangkat lunak dan perangkat keras. Dalam tiga panjang kunci algoritma *AES*, yang paling sering digunakan adala kunci sepanjang 128 bit. Bila di bawah panjang kunci, maka waktu komputasi iteratifnya akan menjadi 10. Pada saat ke tahap final, setiap tahap terdiri dari lima bagian: SubBytes,S-box, ShiftRows, MixColumns, AddRoundKey. [7]

### 2.4. Algoritma RC5

Algoritma pertama kali ditemukan oleh Ron Rivest di sebuah laboratorium tempat dimana RSA dan MIT dirancang. [17] Algoritma kriptografi ini memiliki beberapa karakteristik sebagai berikut :

- RC5 berjenis cipher block dengan beberapa metode operasinya diantaranya adalah CBC (Cipher Block Chaining)
- RC5 dapat diimpelementasikan pada perangkat lunak maupun perangkat keras
- RC5 memiliki panjang kunci 32 bit sampai dengan 64 bit.
- RC5 diharuskan untuk tidak memakan terlalu banyak memory agar dapat diimplementasikan pada chip chip kecil atau perangkat yang ukuran memorynya terbatas.
- RC5 menggunakan metode data depended rotations dimana data akan diputar sebanyak N, dan besar N juga ditentukan dari data data yang lain.

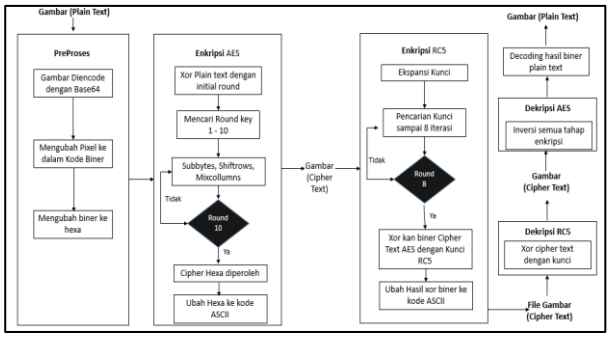
Untuk prosesnya, RC5 memiliki 2 tahap yaitu proses ekspansi kunci, dan enkripsi dengan cara melakukan xor setiap bit plain text dengan bit dari hasil ekspansi kunci yang dihasilkan, sehingga akan menghasilkan cipher text. Lalu untuk proses dekripsinya masih mirip seperti proses enkripsi dimana nilai biner dari cipher text akan di xor kan dengan biner dari kunci sehingga akan menampilkan plain text yang dihasilkan dari proses dekripsi.

### 2.5. Base64

Base64 merupakan algoritma untuk mengkonversikan bilangan biner ke kode ASCII, atau disebut dengan encode dan decode. Karakter yang dihasilkan dari hasil konversi base64 mecakup A – Z, a – z, 0 – 9, dan simbol simbol lainnya [19]. Base64 ini digunakan untuk mengkonversi pixel pixel gambar menjadi biner yang kemudian akan dilakukan proses algoritma AES dan RC5.

## 3. PERANCANGAN APLIKASI

Pada aplikasi, gambar yang akan dienkripsi dengan aes akan melewati beberapa tahap. Untuk penjelasannya akan dijelaskan di bawah ini




Gambar 3.1. Perancangan aplikasi

**Tahap pertama**, gambar akan diubah dulu ke pixel menggunakan base64. Hasil ubahannya adalah pixel pixel pada gambar akan dikonversi kedalam kode biner, setelah menjadi kode biner maka dikonversikan lagi masing masing 8 bit kode binernya ke kode hexa.

**Tahap kedua**, hasil kode hexa akan dienkripsikan menggunakan algoritma AES. Kode hexa yang dihasilkan dari base64 akan dixor kan dengan initial round atau key round awal, lalu menghitung roundkey nya sampai dengan 10 dengan rumus AES. Setelah itu, hasil xor tadi akan dilakukan tahap subbytes, shiftrows, dan mixcollumns, lalu addroundkey, dimana hasil mixcolumn akan dixor kan dengan round 1 (setelah initial round). Lalu ketiga proses diatas akan diulangi sampai ke round 10. Jika sudah sampai round 10 maka cipher dari hexa akan diperoleh, dan hexa nya akan dilihat dan dibuah ke kode ASCII.

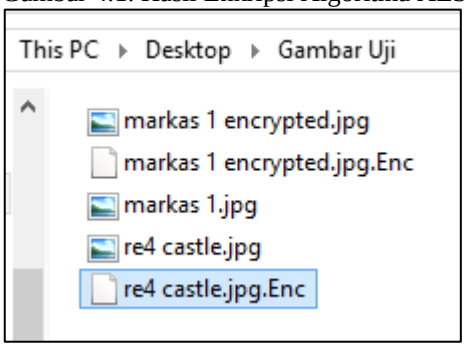
**Tahap ketiga**, hasil hexa pada cipher AES akan dikonversikan lagi ke biner, lalu algoritma AES akan mengekspansi kunci. Setelah ekspansi kunci dilakukan maka akan dilakukan pencarian kunci sampai 8 iterasi, setelah sampai iterasi ke 8 maka biner dari plain text (biner dari cipher text AES) akan xor kan dengan biner dari kunci akhir hasil pencarian kunci RC5. Hasil dari xor biner akan dirubah ke kode ASCII, maka akan terbentuk cipher dari hasil enkripsi RC5.

**Tahap Keempat**, yaitu proses dekripsi. Dimana cipher text dari RC5 akan dixor dengan kunci sehingga menghasilkan plain text RC5. Lalu biner hasil plain text RC5 akan diubah ke hexa dan dilakukan proses dekripsi dengan AES dimana seluruh tahap enkripsi AES dilakukan dengan cara yang sama, namun dibalik atau diinvers. Hasil plain text dekripsi AES akan berupa kode hexa kembali, dan kode hexa akan diubah kembali ke biner dan binernya akan didecode kembali menggunakan base64, sehingga hasil akhirnya akan berupa plain text gambar kembali.

## 4. HASIL DAN PEMBAHASAN

Pada penelitian ini terdapat hasil yang didapat dari proses enkripsi, dekripsi, pengujian kunci, serta pengujian terhadap file yang terenkripsi. Untuk hasil dari enkripsi dapat dilihat dari gambar 4.1 dan gambar 4.2


Gambar 4.1. Hasil Enkripsi Algoritma AES



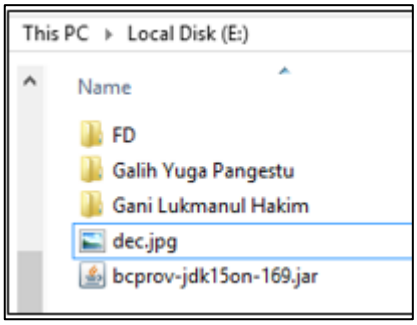

Gambar 4.2. Hasil Enkripsi Algoritma RC5

### 4.1. Hasil Enkripsi

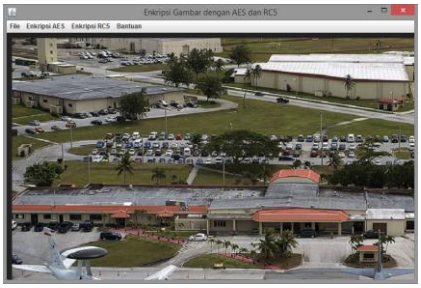
Berdasarkan Hasil enkripsi AES pada gambar 4.1, tampilan gambar akan disamarkan menggunakan algoritma AES sehingga gambar yang asli tidak ditampilkan, lalu hasil dari enkripsi RC5 akan mengenkripsi format file gambar menjadi .Enc agar gambar tidak dapat dibuka.

### 4.2. Hasil Dekripsi

Hasil Dekripsi merupakan tahap perubahan kembali file gambar menjadi plain text agar file gambar dapat dibuka kembali dan gambarnya dapat ditampilkan kembali. Prosesnya adalah menginvers seluruh proses enkripsi RC5 dan menginvers seluruh proses Enkripsi AES. Untuk hasil dari dekripsi dapat dilihat pada Gambar 4.3. dan 4.4.




Gambar 4.3. Hasil Dekripsi RC5



Gambar 4.3. Hasil Dekripsi AES

### 4.3. Pengujian Kunci

Untuk pengujian kunci pada *AES* dan *RC5*, penulis menggunakan tools untuk pengecekan kunci bernama *How Secure Is My Password*. *How secure is my password* adalah tools yang berfungsi untuk mengecek seberapa aman password atau kunci yang digunakan untuk mengamankan suatu data dengan cara membaca inputan kunci yang akan dicek dan dilihat di dalam database kunci yang berisi kunci kunci lemah sampai kuat. *How secure is my password* memiliki 3 metode faktor kunci sebagai berikut:

- **Jumlah Karakter**: kunci harus setidaknya memiliki minimal 8 sampau 10 karakter, namun 16 sampai 20 adalah ideal.
- **Kombinasi**: kunci harus termasuk huruf, angka, dan berbagai jenis simbol. Setiap karakter akan dicek dan dihitung ada berapa huruf, angka, dan simbol yang dimilikinya dan nanti akan.
- **Keunikan**: kunci seharusnya tidak repetitif atau memiliki nilai yang sama pada setiap digit kuncinya.

Dengan ketiga faktor di atas, nantinya tools akan mengkalkulasikan waktu yang dibutuhkan untuk menebak atau menyerang kunci. Tabel pengujian kunci adalah sebagai berikut :

Tabel 4.1. Pengujian Kunci AES

| No | Nama File | Kunci | Lama waktu untuk crack |
|---|---|---|---|
| 1 | Gambar 1 | 1234567890123456 | 2 Hari |
| 2 | Gambar 2 | abcdefghijklmnop | 34 Ribu Tahun |
| 3 | Gambar 3 | qwerty*123/dgvm’ | 2 Ratus Miliar Tahun |
| 4 | Gambar 4 | paSSw0rd1s/sdwad | 2 Triliun Tahun |
| 5 | Gambar 5 | //////////////// | 2 Ratus Tahun |

Tabel 4.2. Pengujian Kunci RC5

| No | Nama File | Kunci | Lama waktu untuk crack |
|---|---|---|---|
| 1 | Gambar 1 | 12345678 | Langsung tertebak |
| 2 | Gambar 2 | abcdefgh | Langsung tertebak |
| 3 | Gambar 3 | qwertyui | Langsung tertebak |
| 4 | Gambar 4 | A*0/>_.& | 4 Jam |
| 5 | Gambar 5 | =a_Z% ‘k | 20 Jam |

Dari hasil tabel pengecekan lama penebakkan kunci AES dan RC5, kunci AES membutuhkan waktu lebih lama untuk ditebak atau diketahui karena kunci AES sepanjang 128 bit atau 16 digit, dibandingkan RC5 yang hanya 64 bit atau 16 digit. Kedua kunci algoritma enkripsi diatas perlu dikombinasikan dengan angka, huruf, dan simbol agar waktu penebakan kuncinya semakin lama.

### 4.3. Pengujian Penetrasi File

Pengujian Penetrasi File merupakan tahap uji kedua dimana akan dilakukan pengecekan apakah file yang telah ter*enkripsi* oleh *AES* dan *RC5* dapat di*dekripsi* atau tidak, menggunakan aplikasi pihak ketiga. Pada Penetrasi File ini penulis mengunakan *OpenSSL*, salah satu *Tools* untuk melakukan *enkripsi* dan *dekripsi* yang cukup mampu membuka file yang ter*enkripsi* algoritma *AES*, sedangkan untuk algoritma *RC5*, penulis menggunakan *Tools* bernama *AlphaDecrypter*, dimana *Tools* tersebut merupakan *Tools* untuk men*dekripsi* file yang ter*enkripsi* ransomware dengan format file *.Enc*.

*OpenSSL* merupakan tools untuk melakukan enkripsi dan dekripsi secara online yang disediakan oleh vendor untuk keperluan keamanan sistem atau data yang terdapat cukup banyak algoritma kriptografi untuk mengamankan data. Pada pengujian penetrasi File gambar yang terenkripsi AES ini akan menggunakan OpenSSL karena OpenSSL memiliki fitur dekripsi algoritma kriptografi AES 128 bit [20]. Pengujian keamanan dengan menggunakan aplikasi pihak ketiga ini disebut dengan *Cyber Attack.* Untuk langkah langkah pengujiannya akan dijelaskan pada bagian selanjutya.

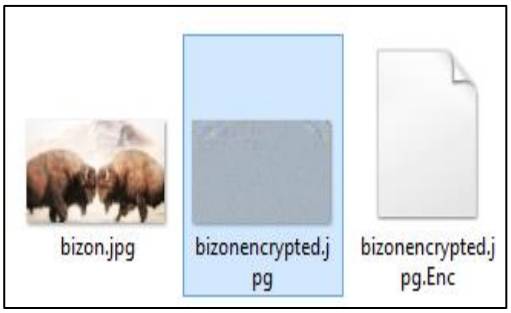
### 4.4. Pengujian Algoritma AES

Pada pengujian kali ini, file yang ter*enkripsi AES* akan diuji dengan cara melakukan *dekripsi* menggunakan *OpenSSL*. Berikut adalah tahap pengujiannya:

1. Buka dahulu openssl pada cmd:


Gambar 4.5. Langkah awal membuka OpenSSL

2. Kita akan melakukan *dekripsi* pada gambar yant ter*enkripsi* *AES* dengan nama bizonencryted.jpg. Masukkan perintah *dekripsi* dibawah ini :




Gambar 4.6. File *AES* yang akan di*dekripsi*

```
enc -AES-128-cbc -d -a -salt -in bizonencrypted.jpg -out bizonplain.jpg
```

Seperti di bawah ini :

```
OpenSSL> enc -aes-128-cbc -d -a -salt -in bizonencrypted.jpg -out bizonplain.jpg
```

Gambar 4.7. Perintah untuk melakukan *dekripsi* file *AES*

3. Setelah itu kita akan memasukkan kunci yang kita gunakan untuk meng*enkripsi* gambar dengan *AES* (dari program) :

```
enter aes-128-cbc decryption password
```

Gambar 4.8. Perintah untuk menginputkan kunci untuk *dekripsi*

4. Dan hasilnya seperti ini :

```
error reading input file
error in enc
```

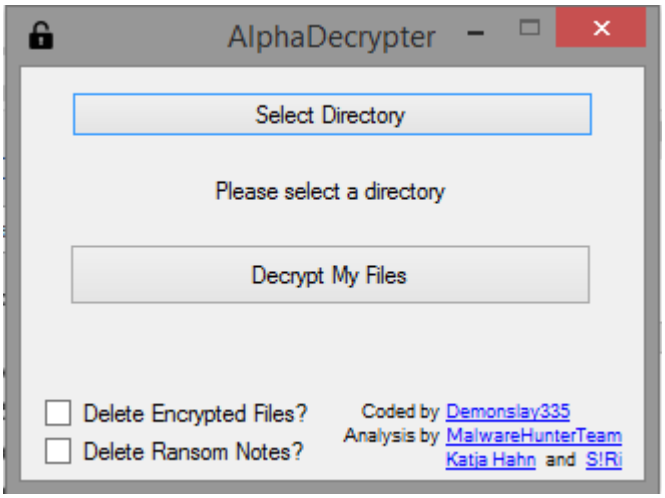
Gambar 4.9. Pesan Error OpenSSL

Hasil di atas menunjukkan gambar yang telah di*enkripsi* oleh *AES*, tidak dapat terbaca oleh OpenSSL ketika memasukkan kunci untuk *dekripsi*. Ini membuktikkan bahwa gambar yang telah ter*enkripsi* oleh *AES* cukup aman dan tidak mudah ter*dekripsi* bahkan dengan OpenSSL sekalipun.

### 4.5. Pengujian Algoritma RC5

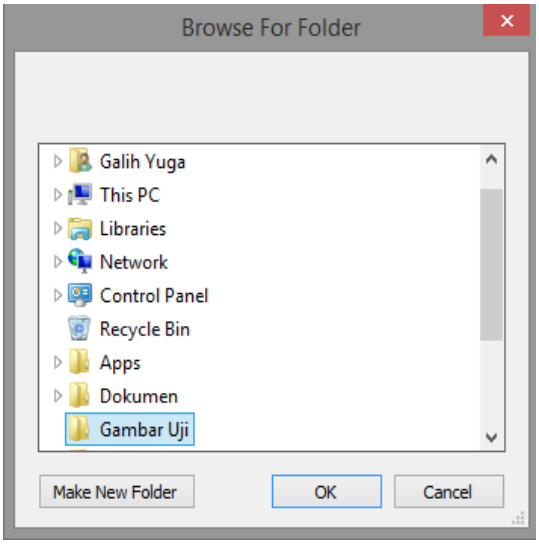
Untuk penetrasi file yang Ter*enkripsi* *RC5* dengan format .enc, *Tools* yang akan digunakan akan berbeda dengan pengujian penetrasi file *AES*. *Tools* yang akan digunakan untuk penetrasi File yang *RC5* adalah *AlphaDecrypter* yang mampu men*dekripsi* berbagai macam ransomware dengan format .enc. Cara kerja AlphaDecrypter masih sama seperti OpenSSL, yaitu AlphaDecrypter merupakan tools pihak ketiga yang disediakan vendor untuk melakukan dekripsi terhadap file yang telah terenkripsi oleh ransomware berformat .enc. pengujian ini disebut dengan *Cyber Attack*. Untuk proses dan langkah pengujian algoritma RC5 adalah sebagai berikut:

1. Buka aplikasi AlphaDecrypter

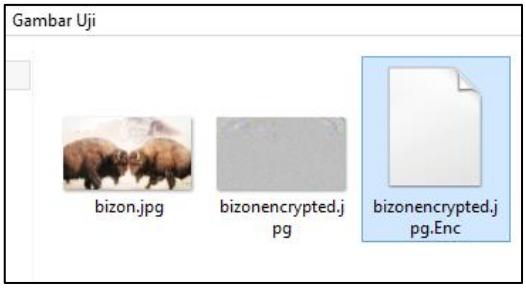



Gambar 4.10. Tampilan AlphaDecrypter

2. Kemudian Klik Select Directory untuk memilih dimana kita menyimpan file yang Ter*enkripsi* *RC5*.

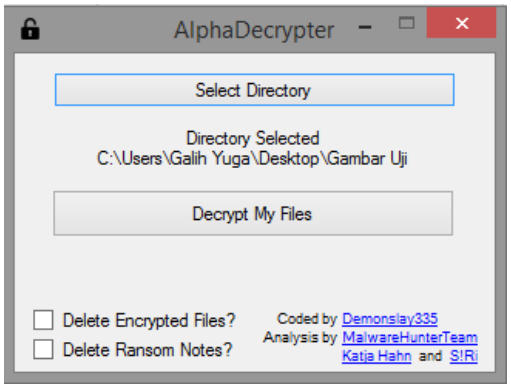



Gambar 4.12. Pilih Directory File

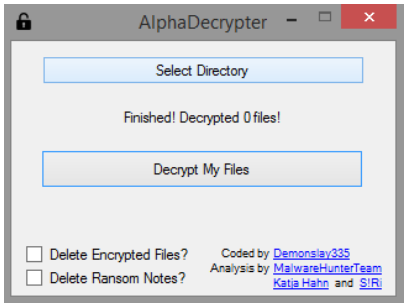



Gambar 4.13. Target File yang akan di*dekripsi*

3. Setelah directory tempat gambar terpilih, kemudian klik tombol *Decrypt My Files*. Nantinya akan menampilkan ada berapa file yang akan ter*dekripsi*




Gambar 4.13. Directory yang akan di*dekripsi*




Gambar 4.14. Hasil *Dekripsi* AlphaDecrypter

Dari Hasil analisa diatas terbukti bahwa *AlphaDecrypter* tidak mampu membaca dan melakukan *dekripsi* terhadap file yang ter*enkripsi RC5* dari sistem yang telah dibuat penulis, sehingga hal ini akan semakin mendukung keamanan file *RC5*.

## 5. KESIMPULAN

*Kriptografi* dalam dunia komputer merupakan bidang ilmu untuk menyandikan atau merahasiakan suatu data. *Kriptografi* ganda menerapkan 2 lapisan keamanan pada file gambar yaitu menerapkan algoritma *AES* (Advanced Encryption Standard) dan *RC5* (Rivest Code 5) yang berfungsi sebagai keamanan ekstra. Dari hasil pengujian pada file gambar yang ter*enkripsi AES* menunjukkan bahwa gambar yang telah disamarkan tidak dapat ditebak atau di*dekripsi* tampilannya jika tidak mempunyai kuncinya dan program untuk melakukan *dekripsi*nya, dan algoritma *RC5* pun tidak dapat dibuka filenya jika tidak mempunyai kuncinya. Kedua algoritma diatas memiliki kunci yang simetris, sehingga proses untuk *enkripsi* dan *dekripsi*nya cukup cepat dan mudah.

Dari hasil pengujian Penetration Testing di atas, menunjukkan bahwa pengujian kunci bertujuan untuk menghitung seberapa lama kunci dapat diserang atau diketahui oleh penyerang, sementara pengujian penetrasi file bertujuan untuk melihat seberapa amankah file yang telah ter*enkripsi* jika dilakukan *dekripsi* oleh aplikasi pihak ketiga. Kedua pengujian ini cukup penting untuk melihat dan menguji keamanan suatu data sehingga data gambar menjadi aman dan tidak memungkinkan untuk diakses, dan dicuri oleh siapapun yang tidak memiliki hak.

Hasil *penetration test* dengan menggunakan aplikasi *openssl* dan *alphadecrypter* menunjukkan bahwa gambar yang telah ter*enkripsi* oleh *AES* dan *RC5* dari program yang telah dibuat, tidak dapat di*dekripsi* lagi oleh aplikasi pihak ketiga.

## DAFTAR PUSTAKA

[1] A. Abdulgader, M. Ismail, N. Zainal, and T. Idbeaa, “Enhancement of AES algorithm based on chaotic maps and shift operation for image encryption,” *J. Theor. Appl. Inf. Technol.*, vol. 71, no. 1, pp. 1–12, 2015.

[2] B. Schneier, “Applied Cryptography,” *Electr. Eng.*, vol. 1, no. [32, pp. 429–455, 1996, doi: 10.1.1.99.2838.

[3] A. Muhammad Abdullah and A. Muhamad Abdullah, “Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data View project Call for papers View project Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt,” 2017, [Online].

[4] P. Karthigaikumar, “Simulation of Image Encryption using AES Algorithm,” pp. 166–172, 2011.

[5] F. International and Q. Zhang, “Digital Image Encryption Based On Advanced Encryption Standard ( AES ) Algorithm,” 2015, doi: 10.1109/IMCCC.2015.261.

[6] P. Singh and P. K. Singh, “IMAGE ENCRYPTION AND DECRYPTION,” vol. 4, no. 7, pp. 150–154, 2013.

[7] A. Arab, M. Javad, and R. Behnam, “An image encryption method based on chaos system and AES algorithm,” *J. Supercomput.*, vol. 75, no. 10, pp. 6663–6682, 2019, doi: 10.1007/s11227-019-02878-7.

[8] C. Li, D. Lin, and F. Hao, “Cryptanalyzing an image encryption algorithm based on autoblocking and electrocardiography,” no. Chengqing Li, 2018.

[9] Z. Hua, Y. Zhou, C. Pun, and C. L. P. Chen, “2D Sine Logistic modulation map for image

encryption," *Inf. Sci. (Ny).*, vol. 297, pp. 80–94, 2015, doi: 10.1016/j.ins.2014.11.018.

[10] K. R. Saraf, V. P. Jagtap, and A. K. Mishra, "Web Site : www.ijettcs.org Email : editor@ijettcs.org Text and Image Encryption Decryption Using Advanced Encryption Standard," vol. 3, no. 3, 2014.

[11] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, "A Comprehensive Evaluation of Cryptographic Algorithms : DES ," *Procedia - Procedia Comput. Sci.*, vol. 78, no. December 2015, pp. 617–624, 2016, doi: 10.1016/j.procs.2016.02.108.

[12] W. Cao, Y. Zhou, and C. L. P. Chen, "Author ' s Accepted Manuscript Medical Image Encryption Using Edge Maps Reference : To appear in : Signal Processing Accepted date : 3 October 2016," *Signal Processing*, 2016, doi: 10.1016/j.sigpro.2016.10.003.

[13] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Opt. Lasers Eng.*, vol. 88, pp. 197–213, 2017, doi: 10.1016/j.optlaseng.2016.08.009.

[14] W. A. Prabowo, A. F. Harahap, and R. Ismadiah, "Penyandian File Word Berdasarkan Algoritma Rivest Code 5 (RC5)," *J-SAKTI (Jurnal Sains Komput. dan Inform.*, vol. 2, no. 1, p. 47, 2018, doi: 10.30645/j-sakti.v2i1.55.

[15] I. N. Purnama, "Implementasi Algoritma Enkripsi Rc5 Untuk Mengamankan Gambar Pada Perangkat Android," *J. Inform. dan Rekayasa Elektron.*, vol. 2, no. 2, p. 1, 2019, doi: 10.36595/jire.v2i2.108.

[16] S. H. Suryawan, "PENGAMANAN DATA FILE DENGAN MENGGUNAKAN ALGORITMA ENKRIPSI RIVEST CODE 5," vol. 8, no. 2, pp. 44–49, 2013.

[17] R. Shahzadi, S. M. Anwar, F. Qamar, M. Ali, and J. J. P. C. Rodrigues, "Chaos based enhanced rc5 algorithm for security and integrity of clinical images in remote health monitoring," *IEEE Access*, vol. 7, pp. 52858–52870, 2019, doi: 10.1109/ACCESS.2019.2909554.

[18] M. I. Zulfikar, G. Abdillah, and A. Komarudin, "Kriptografi untuk Keamanan Pengiriman Email Menggunakan Blowfish dan Rivest Shamir Adleman (RSA)," *Semin. Nas. Apl. Teknol. Inf.*, pp. 19–26, 2019.

[19] R. Rahim *et al.*, "Combination Base64 Algorithm and EOF Technique for Steganography," *J. Phys. Conf. Ser.*, vol. 1007, no. 1, pp. 1–6, 2018, doi: 10.1088/1742-6596/1007/1/012003.

[20] S. Briongos, P. Malagón, J. M. de Goyeneche, and J. M. Moya, "Cache misses and the recovery of the full AES 256 Key," *Appl. Sci.*, vol. 9, no. 5, pp. 1–24, 2019, doi: 10.3390/app9050944.