Implementasi Sistem Keamanan Data Berbasis Kriptografi Rivest Code 6, Vigenere Chipper dan Kompresi Data LZW

Eko Suharyanto, S.T.,M.Kom
STMIK Eresha, Jurusan Teknik Informatika
Jl Raya Puspitek No. 11, Serpong, Kecamatan Setu, Kota Tangerang Selatan, Banten 15314

Email : ekosuharyanto354@gmail.com

ABSTRAK

Masalah keamanan data dan informasi menjadi salah satu hal yang penting di era kemajuan teknologi informasi seperti sekarang ini. Dalam pertukaran data dan informasi dibutuhkan tingkat keamanan yang tinggi, hal ini disebkan sering terjadinya pencurian data oleh pihak-pihak yang tidak bertanggung jawab. Data yang dicuri tersebut dapat dengan mudah dimanfaatkan untuk tujuan manipulasi ataupun tujuan criminal lainnya. Oleh karena itu, dibutuhkan suatu metode yang tepat untuk melindungi data tersebut, yaitu menggunakan metode kriptografi. Algoritma kriptografi terdiri atas algoritma enkripsi (E) dan algoritma dekripsi (D). Untuk lebih menjaga kemanan data atau informasi penting maka dilakukan penigkatan kemanannya dengan mengkombinasikan dua metode kriptografi dan kompresi data. Metode enkripsi Rivest Code 6 (RC6) merupakan algoritma kunci simetris yang dapat membentuk block chipper. Vigenere Cipher salah satu kriptografi klasik yang paling tangguh. Kedua metode kriptografi tersebut kemudian dikombinasikan dengan Lampel-Ziv-Welch (LZW).

Kata Kunci *Kriptografi, Rivest Code 5, Vigenere Cipher, Kompresi Data LZW, File*

1. PENDAHULUAN

Metode kriptografi yaitu sebuah seni dan bidang keilmuan dalam penyandian informasi atau pesan dengan tujuan menjaga keamanannya. Kriptografi adalah ilmu mengenai teknik enkripsi dimana naskah asli (plaintext) diacak menggunakan suatu kunci enkripsi menjadi naskah acak yang sulit dibaca (ciphertext) oleh orang yang tidak memiliki kunci dekripsi. Untuk membaca data yang sudah terenkripsi seseorang harus mempunyai kunci dekripsi untuk mendekripsikan data tersebut agar bisa mengetahui isi data atau informasi tersebut.

1.1 Latar Belakang

Pengamanan pesan, data, atau informasi selain bertujuan untuk meningkatkan keamanan juga dapat berfungsi untuk:

a. Melindungi pesan, data, atau informasi agar tidak dapat di baca oleh orang-orang yang tidak berhak.
b. Mencegah agar orang-orang yang tidak berhak, menyisipkan atau menghapus pesan, data, atau informasi.

Perjalanan informasi sering kali menjadi tidak aman. Banyak terjadi gangguan-gangguan pihak yang tidak berhak dan tidak bertanggung jawab. Salah satu ilmu untuk menjaga keamanan dan kerahasiaan data atau informasi yaitu kriptografi. Algoritma kriptografi terdiri dari Algoritma Enkripsi (E) dan Algoritma Dekripsi (D).

1.2 Tujuan Penelitian

Tujuan dari penelitian ini adalah memberikan tinjauan literatur tentang teknik keamanan file dengan mempergunakan gabungan 2 metode kriptografi Riverst Code 6 dan Vigenere Chipper kemudian dikombinasikan dengan kompresi data LZW.

2 LANDASAN TEORI

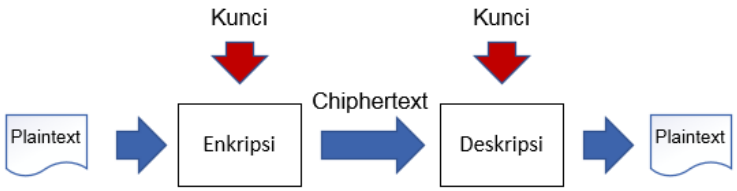
2.1 Konsep Dasar Kriptografi

Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan Enkripsi dan Dekripsi. Pesan yang akan

dienkripsi disebut sebagai plaintext (teks biasa/ data asli). Algoritma yang dipakai untuk mengenkripsi dan mendekripsi sebuah plaintext melibatkan penggunaan suatu bentuk kunci. Plaintext yang telah dienkripsi atau dikodekan dikenal sebagai ciphertext/ teks sandi. Secara umum proses enkripsi dan deskripsi digambarkan secara matematis sebagai berikut:

Ek(P) → C (Proses Enkripsi)
Dk(C) → P (Proses Dekripsi)
Dk(E(P)) = P (Proses Dekripsi)

Dalam proses tersebut, plaintext disandikan dengan P dengan suatu kunci K lalu dihasilkan pesan C. Pada proses dekripsi, C diuraikan dengan menggunakan kunci K sehingga menghasilkan M yang sama dengan sebelumnya.



Gambar 1. *Cryptosystem*

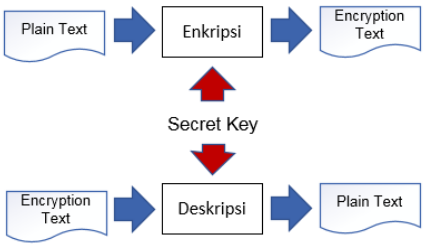
Setiap cryptosystem yang baik memiliki karakteristik sebagai berikut:

a. Keamanan sistem terletak pada kerahasiaan kunci, bukan pada kerahasian algoritma yang digunakan.
b. Cryptosystem yang baik memiliki ruang kunci (keyspace) yang besar.
c. Cryptosystem yang baik akan menghasilkan ciphertext yang terlihat acak dalam seluruh test statistik yang dilakukan.
d. Cryptosystem yang baik mampu menahan seluruh serangan yang telah dikenal sebelumnya.

2.2 Jenis Algoritma Cryptography

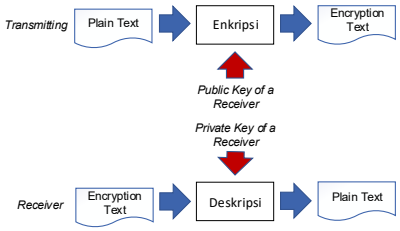
a. Algoritma Klasik
Menerapkan teknik enkripsi konvensional (simetris). Dua teknik dasar yang biasa digunakan, yaitu:
- Teknik Subsitusi: penggantian setiap karakter plaintext dengan karakter lain.
- Teknik Transposisi: Teknik ini menggunakan permutasi karakter.

b. Algoritma Modern
Algoritma modern fokus kepada tingkat algoritma dan kunci yang digunakan.
1) Algoritma Simetris:
menggunakan kunci yang sama untuk proses enkripsi dan deskripsinya.



Gambar 2. Algoritma Kriptograsi Simetris

2) Algoritma Asimetris
Kunci dari ekripsi berbeda dengan kunci dari deskripsi. Dalam sistem ini kunci enkripsi sering disebut public key sedangkan key dekripsi sering disebut private key.



Gambar 3. Algoritma Kriptograsi Asimetris

2.3 Algoritma Rivest Code 6 (RC 6)

Algoritma RC6 adalah versi yang dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC6-w/r/b, dimana parameter w merupakan ukuran kata dalam satuan bit, r adalah bilangan bulat bukan negatif yang menunjukan banyaknya iterasi selam proses enkripsi, dan b

menunjukan ukuran kunci enkripsi dalam byte. Ketika algoritma ini masuk sebagai kandidat AES, maka ditetapkan nilai parameter w = 32, r = 20 dan b bervariasi antara 16, 24, dan 32 byte.[ABD02]
RC6-w/r/b memecah block 128 bit menjadi 4 buah block 32 bit, dan mengikuti enam aturan operasi dasar sebagai berikut:
A + B : Operasi penjumlahan bilangan integer.
A – B : Operasi pengurangan bilangan integer.
A B : Operasi exclusive-OR (XOR)
A x B : Operasi perkalian bilangan integer.
A<<<B : A dirotasikan ke kiri sebanyak variabel kedua (B)
A>>>B : A dirotasikan ke kanani sebanyak variabel kedua (B)

a. Proses Enkripsi

RC6 memecah 128 bit menjadi 4 buah block 32 bit, maka algoritma ini bekerja dengan 4 buah register 32-bit A, B, C, D. Byte yang pertama dari plaintext atau ciphertext ditempatkan pada byte A, sedangkan byte yang terakhirnya ditempatkan pada byte D. Maka didapatkan (A, B, C, D) = (B, C, D, A) yang berarti bahwa nilai yang terletak pada sisi kanan berasal dari register disisi kiri.[ABD02]. Berikut ini adalah algoritma enkripsi RC6:

```
B = B + S[0]
D = D + S[1]

for i = 1 to 20 do
{
        t   = (B x (2B + 1)) <<< 5
        A = ((A ⊕ t) <<< u) + S[2i]
        C = ((C ⊕ u) <<< t) + S[2i + 1 ]
        (A, B, C, D) = (B, C, D, A)
}
A = A + S[42]
C = C + S[43]
```

Algoritma RC6 menggunakan 44 buah sub kunci yang berasal dari kunci dan dinamakan dengan S[0] hingga S[43]. Masing-masing sub kunci panjangnya 32 bit. Proses enkripsi pada algoritma RC6 dimulai dan diakhiri dengan proses whitening bertujuan untuk menyamarkan iterasi yang pertama dan yang terakhir dari proses enkripsi dan dekripsi. Pada proses whitening awal, nilai B akan dijumlahkan dengan S[0], dan nilai D dijumlahkan dengan S[i]. Pada masing-masing iterasi pada RC6 menggunakan 2 buah sub kunci. Sub kunci pada iterasi yang pertama menggunakan S[2] dan S[3], sedangkan iterasi-iterasi berikutnya menggunakan sub-sub kunci lanjutannya. Setelah iterasi ke-20 selesai, dilakukan proses whitening akhir dimana nilai A dijumlahkan dengan S[42], dan nilai C dijumlahkan dengan S[43].[ABD02]. Setiap iterasi pada algoritma RC6 mengikuti aturan sebagai berikut, nilai B dimasukan ke dalam fungsi f, yang didefinisikan sebagai f(x) =x(2x+1), kemudian diputar kekiri sejauh lg-w atau 5 bit. Hasil yang didapat pada proses ini dimisalkan sebagai u. Nilai u kemudian di XOR dengan C dan hasilnya menjadi nilai C. Nilai t juga digunakan sebagai acuan bagi C untuk memutar nilainya kekiri. Begitu pula dengan nilai u, juga digunakan sebagai acuan bagi nilai A untuk melakukan proses pemutaran kekiri. Kemudian sub kunci S[2i] pada iterasi dijumlahkan dengan A, dan sub kunci S[2i+1] dijumlahkan dengan C. keempat bagian dari block kemudian akan dipertukarkan dengan mengikuti aturan, bahwa nilai A ditempakan pada D, nilai B ditempatkan pada A, nilai C ditempatkan pada B, dan nilai (asli) D ditempatkan pada C. Demikian iterasi tersebut akan terus berlangsung hingga 20 kali.[ABD02]

b. Proses Dekripsi

Proses dekripsi ciphertext pada algoritma RC6 merupakan pembalikan dari proses enkripsi. Pada proses whitening, bila proses

enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan. Sub kunci yang digunakan pada proses whitening setelah iterasi terakhir diterapkan sebelum iterasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses whitening sebelum iterasi pertama digunakan pada whitening setelah iterasi terkahir. Akibatnya, untuk melakukan dekripsi, hal yang harus dilakukan semata-mata hanyalah menerapkan algoritma yang sama dengan enkripsi, dengan tiap iterasi menggunakan sub kunci yang sama dengan yang digunakan pada saat enkripsi, hanya saja urutan sub kunci yang digunakan terbalik.[ABD02]. Berikut ini adalah algoritma deskripsi RC6:

```
C = C - S[43]
A = A - S[42]

for i = 20 down to 1 do
{
    (A, B, C, D) = (D, A, B, C)
    u  = (D x (2D + 1)) <<< 5
    t   = ( B x (2B + 1)) <<< 5
    C = (( C - S[ 2i + 1 ] ) >>> t ) ⊕
u
    A = ( ( A - S[ 2i ] ) >>> u) ⊕ t
}
D = D - S[1]
B = B - S[0]
```

c. Kunci

Pengguna memasukkan sebuah kunci yang besarnya b byte, dimana 0£b£255. byte kunci ini kemudian ditempatkan dalam array c w-bit words L[0]…L[c-1]. Byte pertama kunci akan ditempatkan sebagai pada L[0], byte kedua pada L[1], dan seterusnya. (Catatan, bila b=0 maka c=1 dan L[0]=0). Masing-masing nilai kata w-bit akan dibangkitkan pada penambahan kunci round 2r+4 dan akan ditempatkan pada array S[0,…,2r+3].[ABD02].

Konstanta P32 = B7E15163 dan Q32 = 9E3779B9 (dalam satuan heksadesimal) adalah "konstanta ajaib"yang digunakan dalam penjadwalan kunci pada RC6. Nilai P32 diperoleh dari perluasan bilangan biner e-2, dimana e adalah sebuah fungsi logaritma. Sedangkan nilai Q32 diperoleh dari perluasan bilangan biner $\varnothing$-1, dimana $\varnothing$ dapat dikatakan sebagai "golden ratio" (rasio emas). Algoritma untuk pembangkitan kunci RC6 adalah sebagai berikut:

```
S[ 0 ] = 0xB7E15163
for i = 1 to 43 do S[i] = S[i-1] +
0x9E3779B9
A = B = i = j = 0
for k = 1 to 132 do
{
    A = S[i] = (S[i] + A + B) <<< 3
    B = L[j] = (L[j] + A + B) <<< (A +
B)
    i = (i + 1) mod 44
    j = (j + 1) mod c
}
```

2.4 Vigenere Chipper

Vigenere cipher adalah metode menyandikan teks alphabet dengan menggunakan deretan sandi Caesar berdasarkan huruf – huruf pada kata kunci. Teknik dari subtitusi vigenere cipher bisa dilakukan dengan du acara:

- Angka: angka metode menyajikan teks alphabet dengan menggunakan deretan sandi Caesar berdasarkan huruf-huruf pada kata kunci
- Huruf: Vigenere Cipher dengan huruf berisi alfabet yang dituliskan dalam 26 baris, masing - masing baris digeser ke kiri dari baris sebelumnya membentuk ke-26 kemungkinan sandi Caesar setiap huruf disediakan dengan menggunakan baris yang berbeda-beda sesuai kunci yang diulang.

Rumus dari enkripsi dan dekripsi data vigenere cipher adalah:

- Enkripsi : Ci = (Pi + Ki) mod 26
- Dekripsi : Pi = (Ci – Ki) mod 26; untuk Ci ≥ Ki
  Pi = (Ci+ 26 – Ki) mod 26; untuk Ci ≤ Ki

A	B	C	D	E	F	G	H	I	J

0	1	2	3	4	5	6	7	8	9

K	L	M	N	O	P	Q	R	S	T
10	11	12	13	14	15	16	17	18	19

U	V	W	X	Y	Z
20	21	22	23	24	25

Tabel 1. Subtitusi Algoritma Vigenere

Metode lain untuk melakukan proses enkripsi dengan metode vigenere cipher yaitu menggunakan tabula recta (disebut juga bujursangkar vigenere).

2.5 Algoritma *Lample-Ziv-Welch (LZW)*

Algoritma LZW dikembangkan oleh Abraham Lempel, Jacob Ziv, dan Terry Welch dan dipublikasikan pada tahun 1984 oleh Terry Welch. LWZ dirancang sebagai peningkatan dari algoritma LZ78. Algoritma ini mereduksi jumlah token yang dibutuhkan menjadi satu simbol saja. Simbol ini merujuk kepada index dalam dictionary. Proses kerjanya mirip dengan algoritme LZ78, tetapi jika pada algoritme LZ78 dictionary dimulai dari keadaan kosong, LZW mengisi dictionary ini dengan seluruh simbol alfabet yang dibutuhkan. Pada kasus yang umum, 256 index pertama dari dictionary diisi dengan karakter ASCII dari 0-255. Karena dictionary telah diisi dengan semua kemungkinan karakter terlebih dahulu, maka karakter masukan pertama akan selalu dapat ditemukan dalam dictionary. Inilah yang menyebabkan token pada LZW hanya memerlukan satu simbol saja, yang merupakan pointer pada dictionary (Fitria Diani, Yudi Widhiyasana, 2018).

3 METODOLOGI

Studi literature dari review paper, yang dicari dari IEEE Journal, Google Search dan Buku.

4. ANALISA DAN PEMBAHASAN

4.1 Proses Rancang Bangun

Alur proses dari rangcang bangun dari implmentasi: RC6, Vinegere dan Kompresi LZW adalah sebagai berikut:

Gambar 4. Proses Implementasi

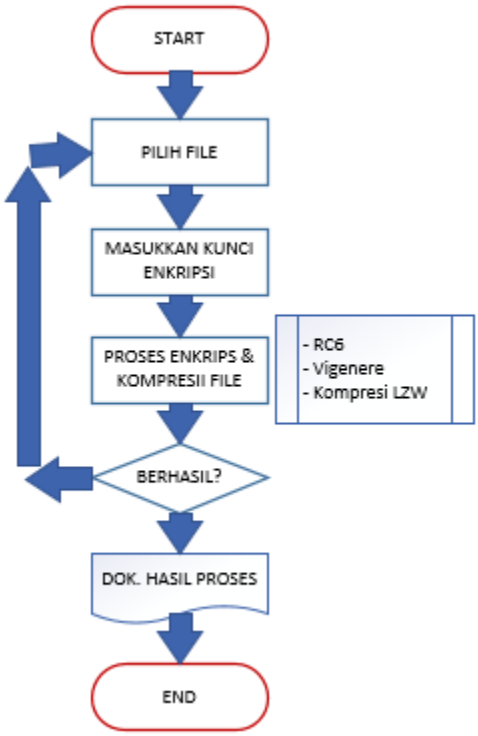
Berdasarkan gambar flowchart diatas adalah sebagai berikut:

a. Pengumpulan Data
   Dilakukan untuk mencari acuan dalam penelitian.
b. Pengolahan Data
   Data yang akan diolah adalah metode kriptografi Rivest Code 6 (RC6), kriptografi Vigenere Chiper dan kompresi Lample-Ziv-Welch (LZW).
c. Pengujian
d. Kesimpulan
   Kemudian mencari kesimpulan dari penelitian yang sudah dilakukan, yang bertujuan untuk mengetahui tingkat keberhasilan aplikasi.

4.2 Rancangan Aplikasi

a. Perancangan Enkripsi

Enkripsi adalah proses merubah atau mengacak isi file yang akan diamankan agar tidak bisa dibaca. Berikut flowchart pengujian enkripsi:



Gambar 5. Proses Enkripsi File

Proses diatas dapat dijelaskan sebagai berikut:

1) Pilih file yang akan dilakukan proses enkripsi
2) Masukan kunci enkripsi biasa disebut key, atau password berfungsi untuk mengkunci file yang akan diamankan. Kunci enkripsi ini akan digunakan saat proses dekripsi.
3) Apabila sudah menentukan kunci enkripsi kemudian lakukan proses enkripsi file. Didalam proses enkripsi file terdapat tiga metode kriptografi, bertujuan untuk meningkatkan keamanan data penting yang akan dilindungi. Alur metode yang digunakan dalam proses enkripsi ini:

a) Rivest Code 6 (RC6) dengan proses dasar enkripsi:

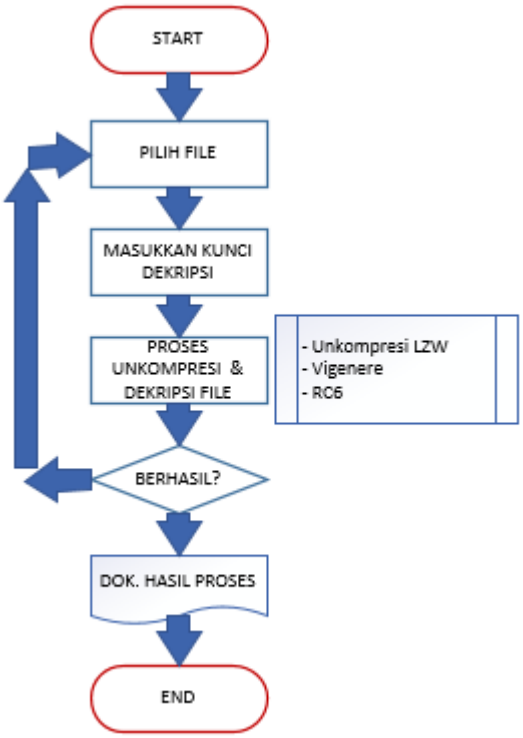
B = B + S[0]
D = D + S[1]

for i = 1 to 20 do
{
t = (B x (2B + 1)) <<< 5
A = ((A $\oplus$ t) <<< u) + S[2i]
C = ((C $\oplus$ u) <<< t) + S[2i + 1]
(A, B, C, D) = (B, C, D, A)
}
A = A + S[42]
C = C + S[43]

b) Vigenere Cipher dengan rumusnya
Ci = (Pi + Ki) mod 26
c) Kompresi Lample-Ziv-Welch (LZW)
proses dari LZW ini adalah meng-encode file. Dimana file yang sudah diencode tidak bisa dibuka, dengan keterangan file rusak pada saat membuka file tersebut.
d) Hasil akhir

b. Perancangan Dekripsi

Dekripsi adalah proses mengembalikan isi file yang sudah diacak ke bentuk aslinya. Berikut flowchart pengujian dekripsi:



Gambar 6. Proses Dekripsi File

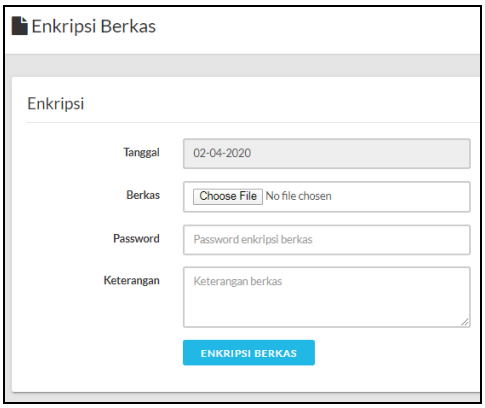
Proses diatas dapat dijelaskan sebagai berikut:

1) Pilih file yang akan dilakukan proses dekripsi. File yang dipilih harus merupakan file yang sudah melalui tahapan enkripsi.
2) Masukan kunci enkripsi biasa disebut key, atau password berfungsi untuk membuka file yang akan mengembalikan file enkripsi ke bentuk semula. Kunci deksipsi ini harus sama dengan kunci pada saat enkripsi file
3) Urutan Algoritma dekripsi adalah kebalikan dari proses dekripsi.

4.3 Implementasi Hasil

Implementasi aplikasi terdiri dari browse file yang akan dienkripsi, save file hasil enkripsi, browse file enkripsi yang akan dideskripsi, save file hasil dekripsi, tombol enkripsi file, tombol dekripsi file, clear file, chek boks untuk pilihan hapus file, serta informasi file yang terdiri nama file yang akan dienkripsi dan dekripsi, ukuran file dan waktu proses enkripsi dan dekripsi file. Subrutin program enkripsi dan dekripsi file ini:

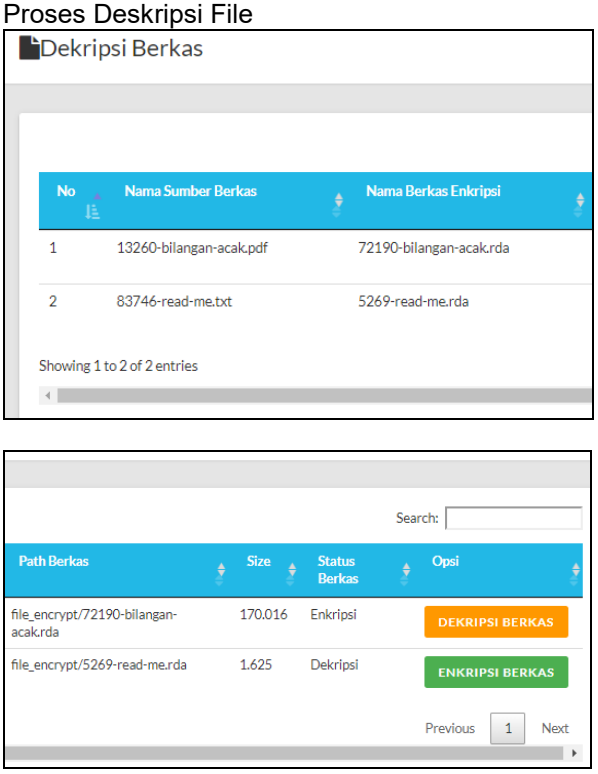
a. Proses Enkripsi File



Gambar 6. UI Kompresi File

Proses ini merupakan proses untuk enkripsi file. Operasi program dimulai dar inisialisasi masukkan program berupa file atau data elektronik kemudian pemberian key (kunci) dan setelah proses enkripsi yang akan menyimpan file enkripsi tersebut kedalam file berekstensi *.rda

b. Proses Deskripsi File



Gambar 7. UI Dekripsi File

Proses ini adalah proses untuk dekripsi yaitu membuka kembali file yang telah dienkripsi. Operasi program dimulai dari inisialisasi file yang telah terenkripsi yaitu dengan format *.rda kemudian memasukkan key (kunci) untukmembuka file enkripsi tersebut dan menyimpannya ke dalam file semula

c. Informasi hasil
Hasil yang diperoleh adalah sebagai berikut:
1) Proses Enkripsi dan Dekripsi pertama
   - Nama file asli : 13260-bilangan-acak.pdf
   - Nama file enkripsi : 72190-bilangan-acak.rda
   - Size : 170.016 bytes
2) Proses Enkripsi dan Dekripsi kedua
   - Nama file asli : 83746-read-me.txt
   - Nama file enkripsi : 5269-read-me.rda
   - Size : 1.625 bytes

5. Kesimpulan

1) RC6 adalah algoritma enkripsi dengan model private key/kunci pribadi yang mempunyai key dekripsi sama dengan key enkripsi.
2) Algoritma RC6 merupakan block cipher dengan ukuran block hingga128 bit dan parameter yaitu RC6-w/r/b dengan nilai w=32 sebagai ukuran kata dalam bit, r=20 sebagai banyaknya iterasi/round dan b ukuran kunci yang bervariasi antara 16, 24 dan 32 byte.
3) Algoritma RC6 terdiri dari 3 bagian yaitu key setup, whitening dan ciphering.
4) Vigenère Chiper (Sandi Vigenère) adalah metode menyandikan teks alfabet dengan menggunakan deretan sandi Caesar berdasarkan huruf-huruf pada kata kunci. Sandi Vigenère merupakan bentuk sederhana dari sandi substitusi polialfabetik.
5) Kelebihan sandi ini dibanding sandi Caesar dan sandi monoalfabetik lainnya adalah sandi ini tidak begitu rentan terhadap metode pemecahan sandi yang disebut analisis frekuensi
6) Prinsip umum kerja algoritma LZW adalah mengecek setiap karakter yang muncul kemudian menggabungkan dengan karakter selanjutnya menjadi sebuah string jika string baru tersebut tidak berada dalam dictionary atau belum diindekkan maka string baru tersebut akan diindekkan ke dalam dictionary.
7) Kombinasi 2 jenis algoritma RC6 dengan Vigenere Chipper ditambah dengan kompresi LZW menghasilkan tingkat pengamanan terhadap data yang lebih optimal jika dibandingkan dengan hanya mempergunakan 1 jenis tingkap pengamanan saja.

Daftar Pustaka

1. Erwin Gunadhi (2016). Pengamanan Data Rekam Medis Pasien Menggunakan Kriptografi Vigenere Cipher. Jurnal STT-Garut, ISSN : 2302 - 7339 Vol. 13 No. 1.
2. Fadhilah Atika. 2017. Implementasi Super Enkripsi dengan Algoritma Variably Modified Permutation Composition (VMPC) dan Two Square Cipher dalam Pengamanan File PDF Berbasis Android. Universitas Sumatera Utara, Repositori.
3. Fitri Diani, Yudi Widhiyasana. 2018. Enkripsi SMS Dengan Menggunakan One Time Pad (OTP) dan Kompresi Lempel-Ziv-Welch (LZW). JNTETI, Vol.7, No.3, ISSN: 2301 – 4156.
4. Gede Angga Pradipta. 2016. Penerapan Kombinasi Metode Enkripsi Vigenere Chiper dan Tranposisi Pada Aplikasi Client Server Chatting. JURNAL SISTEM DAN INFORMATIKA Vol.10, No.2. STMIK STIKOM Bali.
5. Hendrawati, Hamdani, Awang Harsa K. 2014. Keamanan Data Dengan Menggunakan Algoritma Rivest Code 4 (RC4) dan Steganografi Pada Citra Digital. Jurnal INFORMATIKA Mulawarman, Vol.9, No.1. ISSN : 1858-4853.
6. Imelda, Ega Prawira. 2018. Pengamanan Disposisi Dokumen Secara Online Menggunakan Kriptografi Twofish dan Kompresi Huffman pada CV. TMU. Jurnal Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri, Tema A – Penelitian, ISSN 2085-4218, ITN Malang.

7. Indra Gunawan. 2018. Kombinasi Algoritma Caesar Cipher dan Algoritma RSA Untuk Pengamanan File Dokumen dan Pesan Teks. Jurnal Nasional Informatika dan Teknologi Jaringan Vol. 2, No.2, Maret 2018. e-ISSN : 2540-7600, p-ISSN : 2540-7597.
8. Irham Mu'alimin Arrijal, Rusdi Efendi, Boko Susilo. 2016. Penerapan Algoritma Kriptografi Kunci Simetris Dengan Modifikasi Vigenere Cipher dalam Aplikasi Kriptografi Teks. Jurnal Pseudocode, Volume III Nomor 1, Februari 2016, ISSN 2355-5920.
9. Jonas Sahang Benor Tambunan, Muhammad Indra Sukaman, Sri Nofrida Siregar. 2018. Penyandian Pesan Berdasarkan Algoritma RC5 dan El-Gamal. Jurnal dan Penelitian Teknik Informatika, Vol. 2 No.2, e-ISSN: 2541-2019 p-ISSN: 2541-044X.
10. Mutiara Rizky Parlindungan. 2017. Implementasi Super Enkripsi Menggunakan Algoritma RC4A dan MDTM Cipher pada Pengamanan File PDF Berbasis Android. Universitas Sumatera Utara, Repositori Institusi USU, 2017.
11. Ni Luh Devi Lingga Pratiwi , I Gede Santi Astawa, Ida Bagus Made Mahendra, Luh Arida Ayu R. 2018. Penerapan Metode LZW dalam Kompresi File Chat Multimedia pada Sistem E-Marketplace Sarana Upakar Bali. Jurnal Ilmu Komputer, Vol.XI, No.2, e-ISSN: 1979-5661 p-ISSN: 2622-321X
12. Nurcahyo Budi Nugroho, Zulfian Azmi, Saiful Nur Arif. 2016. Aplikasi Keamanan Email Menggunakan Algoritma RC4. Jurnal Ilmiah SAINTIKOM Vol.15, No. 3, ISSN: 1978 – 6603.
13. Nurhadian, Ahmad Pudoli. 2016. Implementasi Keamanan File dengan Kompresi Huffman dan Kriptografi menggunakan Algoritma RC4 serta Steganografi menggunakan End of File Berbasis Desktop pada SMK Negeri 3 Kota Tangerang. Jurnal TICOM Vol. 5 No.1.
14. Rizal Yunan Rifai, Yuli Christyono dan Imam Santoso. 2016. Implementasi Algoritma Kriptografi Rivest Code 4, Rivest Shamir Adleman dan Metode Steganografi untuk Pengamanan Pesan Rahasia Pada Berkas Teks Digital. Jurnal, TRANSIENT, Vol.5, No.1, Maret 2016, ISSN: 2302-9927, 87.
15. Yonata Laia, Mardi Turnip. 2016. Optimasi Rasio Kompresi dan Kompleksitas Waktu Kompresi File Teks Menggunakan Algoritma Lempel-Ziv-Welch (LZW) dengan Fibonacchi Search. Publikasi Jurnal dan Penelitian Teknik Informatika, Vol. 1 Nomor 1, e-ISSN : 2541 – 2019 p-ISSN : 2541 – 044X.
16. Siswanto, Feriadi, Gunawan Pria Utama, Aditya Firdaus A. 2017. Pengamanan Data Dengan Menggunakan Algoritma Kriptografi AES, RC4 dan Kompresi LZ77 Berbasis Java Pada Badan Karantina Pertanian. Majalah Ilmiah INTI, Volume 12, Nomor 2, Mei 2017. ISSN 2339-210X.