Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 2, Januari 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i2.1745

Pengamanan Data File Teks (Word) Menggunakan Algoritma RC4

Dedy Ronald Saragi, Janter Manuel Gultom, Jose Andreas Tampubolon, Indra Gunawan

STIKOM Tunas Bangsa Pematangsiantar, Indonesia
Email: dedyronalsaragi@gmail.com, jantermanuel243@gmail.com, josetampubolon28@gmail.com, indra@amiktunasbangsa.ac.id

Abstrak–Keamanan data pada komputer dapat memberikan sebuah perlindungan terhadap data kita.Teknik kriptografi merupakan teknik yang sangat penting dalam mengamankan data. Kriptografi adalah ilmu mengenai teknik enkripsi dimana "naskah asli" (plaintext) diacak menggunakan suatu kunci enkripsi menjadi "naskah acak yang sulit dibaca" (ciphertext) oleh seseorang yang tidak memiliki kunci dekripsi, salah satu metode kriptografi modern yang dikembangkan adalah algoritma RC4. Algoritma RC4 (Ron's Code / Rivest's Cipher) adalah salah satu algoritma yang dapat digunakan untuk melakukan enkripsi data sehingga data asli hanya dapat dibaca oleh seseorang yang memiliki kunci enkripsi tersebut.

Kata Kunci: Algoritma RC4, Kriptografi, Enkripsi, Dekripsi

Abstract–Data security on computers can provide a protection for our data. Cryptographic techniques are very important techniques in securing data. Cryptography is the science of encryption techniques where the "original text" (plaintext) is encrypted using an encryption key into "random text that is difficult to read" (ciphertext) by someone who does not have a decryption key, one of the modern cryptographic methods developed is the RC4 algorithm. RC4 algorithm (Ron's Code / Rivest's Cipher) is one algorithm that can be used to encrypt data so that the original data can only be read by someone who has the encryption key.

Keywords: RC4 Algorithm, Cryptography, Encryption, Decryption

1. PENDAHULUAN

Proses perkembangan teknologi pada saat ini sangatlah pesat,memanipulasi terhadap teks atau berkas-berkas termasuk dokumen sangat lah mudah dilakukan. Pemalsuan terhadap dokumen umumnya dilakukan dengan cara memanipulasi isi dari dokumen baru dengan desain dan tampilan yang serupa dengan aslinya.Untuk menjaga kerahasian informasi tesebut adalah dengan menyamarkan menjadi bentuk tersandi yang bermakna.

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyinkan pesan. Namun pada pengertian modern kriptografi adalah ilmu yang berdasarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian kriptografi modern adalah tidak saja berurusan hanya dengan penyembunyian pesan namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi [1].

Algoritma yang berfungsi untuk melakukan tujuan kriptografis disebut sebagai algoritma sandi. Algoritma tersebut harus memiliki kekuatan untuk melakukan [2]:

a. Konfusi/pembingungan, mempersulit pembaca biasa untuk memecahkan pesan yang sudah dienkripsi menjadi sandi-sandi tanpa memakai algoritma pendekripsinya.
b. Difusi/peleburan,yaitu dengan cara menghilangkan karakteristik dari informasi yang dienkripsi.

Algoritma kriptografi yang akan digunakan pada penelitian ini adalah algoritma simetris yaitu Rivest Code 4 (RC4). Algoritma RC4 adalah algoritma yang bersifat stream cipher dimana proses penyandiannya berorientasi pada satu bit/ byte data [3].

Algoritma dari metode RC4 Stream Cipher ini terbagi menjadi dua bagian, yaitu : key setup dan stream generation. Pada Key Setup terdapat tiga tahapan proses di dalamnya, yaitu Inisialisasi S-Box, Menyimpan key dalam Key Byte Array, Permutasi pada S-Box. Pada Stream Generation akan menghasilkan nilai pseudorandom yang akan dikenakan operasi XOR untuk menghasilkan ciphertext ataupun sebaliknya yaitu untuk menghasilkan plaintext.

2. METODE PENELITIAN

2.1 Kriptografi

Kriptografi penting dalam dunia teknologi informasi saat ini terutama dalambidang komputer yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi. Kriptografi juga menjadi salah satu syarat penting dalam keamanan teknologi informasi dalam pengiriman pesan penting dan rahasia.

Tujuan Kriptografi

1. Kerahasiaan (*confidentiality*)
   Kerahasiaan adalah layanan yang digunakan untuk menjaga informasi dari setiap pihak yang tidak berwenang untuk mengaksesnya. Dengan demikian informasi hanya akan dapat diakses oleh pihak-pihak yang berhak saja.
2. Integritas data (*data integrity*)

Submitted: **16/12/2019**; Accepted: **16/01/2020**; Published: **25/01/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 2, Januari 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i2.1745

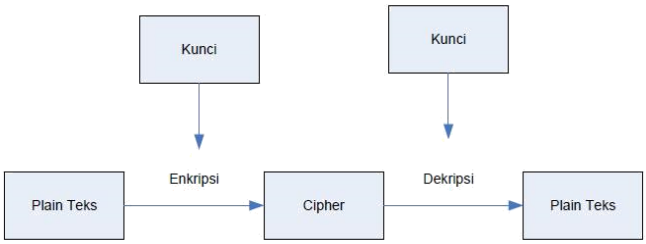
Integritas data merupakan layanan yang bertujuan untuk mencegah terjadinya pengubahan informasi oleh pihak-pihak yang tidak berwenang. Untuk meyakinkan integritas data ini harus dipastikan agar sistem informasi mampu mendeteksi terjadinya manipulasi data. Manipulasi data yang dimaksud meliputi penyisipan, penghapusan, maupun penggantian data.

3. Keaslian (*authentication*)
   Keaslian merupakan layanan yang terkait dengan pembuktian identifikasi terhadap pihak-pihak yang ingin mengakses sistem informasi (*entity authentication*) maupun pembuktian data dari sistem informasi itu sendiri (*data origin authentication*).
4. Ketiadaan penyangkalan (*non-repudiation*)
   Ketiadaan penyangkalan merupakan layanan yang berfungsi untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengiriman pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan.

Mekanisme Kriptografi

1. *Plaintext* (*message*) merupakan pesan asli yang ingin dikirimkan dan dijaga keamanannya.
2. *Chipertext* merupakan pesan yang telah dikodekan atau disandikan sehingga siap untuk dikirimkan.
3. *Cipher* merupakan algoritma matematis yang digunakan untuk proses penyandian *plaintext* menjadi *ciphertext*.
4. Enkripsi (*encryption*) merupakan proses yang dilakukan untuk menyandikan *plaintext* sehingga menjadi *chipertext.*
5. Dekripsi (*decryption*) merupakan proses yang dilakukan untuk memperoleh kembali *plaintext* dari *ciphertext*.
6. Kriptosistem merupakan sistem yang dirancang untuk mengamankan suatu sistem informasi dengan memanfaatkan kriptografi.

Prosesnya pada dasarnya sangat sederhana. Sebuah *plaintext* (m) akan dilewatkan pada proses enkripsi (E) sehingga menghasilkan suatu *ciphertext* (c). Kemudian untuk memperoleh kembali *plaintext*, maka *ciphertext* (c) melalui proses dekripsi(D) yang akan menghasilkan kembali *plaintext* (m). Kriptografi modern selain memanfaatkan algoritma juga menggunakan kunci (*key*) untuk memecahkan masalah tersebut dengan (e)= kunci enkripsi dan (d) = kunci dekripsi. Mekanisme kriptografi seperti ini dinamakan kriptografi berbasis kunci. Dengan demikian kriptosistemnya akan terdiri atas algoritma dan kunci, beserta segala *plaintext* dan *ciphertext*-nya. Proses enkripsi dan dekripsi dilakukan dengan menggunakan kunci ini yang digambarkan seperti pada Gambar 1.



Gambar 1. Kriptografi Berbasis Kunci

Dokumen merupakan data yang sangat penting baik itu berupa dokumen pribadi, perusahaan atau organisasi dan lain sebagainya. Oleh karena itu, sebuah dokumen seharusnya dijaga kerahasiaannya agar tidak disalahgunakan oleh orang yang tidak berhak. Disini seringkali masalah keamanan menjadi urutan kedua atau bahkan urutan yang terakhir dalam daftar hal yang dianggap penting. Disini seringkali masalah keamanan menjadi urutan kedua atau bahkan urutan yang terakhir dalam daftar hal-hal Apabila menggangu performa sistem, masalah keamanan ini sering dikurangi atau bahkan dihilangkan.

Salah satu cara untuk mengamankan sebuah dokumen yaitu dengan mengubah dokumen asli menjadi dokumen yang tidak bisa dibaca oleh orang lain atau sering disebut dengan menggangu performa sistem, masalah keamanan ini sering dikurangi atau bahkan dihilangkan. Salah satu cara untuk mengamankan sebuah dokumen yaitu dengan mengubah dokumen asli menjadi dokumen yang tidak bisa dibaca oleh orang lain atau sering disebut dengan enkripsi.

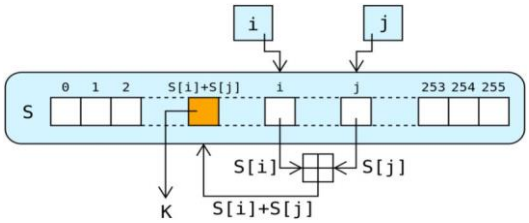
2.2 RC4 (Rivest Chiper 4)

RC4 (Rivest Chiper 4) adalah sebuah synchrone stream chiper, yaitu cipher yang memiliki kunci simetris dan mengenkripsi plainteks secara digit per digit atau byte per byte dengan cara mengkombinasi dengan operasi biner dengan sebuah angka semi acak [2].

Tahap pencarian RC4. Byte keluaran dipilih dengan mencari nilai-nilai S [i] dan S [j], menambahkan mereka bersama-sama modulo 256, dan kemudian menggunakan jumlah sebagai indeks ke dalam S; S (S [i] + S [j]) digunakan sebagai byte dari stream kunci, K.

Submitted: **16/12/2019**; Accepted: **16/01/2020**; Published: **25/01/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 2, Januari 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i2.1745



Gambar 2. Kunci Byte Pengeluaran Modula

Untuk sebanyak mungkin iterasi yang diperlukan, PRGA memodifikasi state dan menghasilkan byte byte keystream. Dalam setiap iterasi, PRGA:

1. Kenaikani i
2. mencari elemenke- i dari S, S [i], dan menambahkannya ke j
3. menukar nilai S [i] dan S [j] kemudian menggunakan jumlah S [i] + S [j] (modulo 256) sebagai indeks untuk mengambil elemen ketiga S (nilai keystream K di bawah)
4. kemudian bitwise eksklusif ORed (XORed) dengan byte pesan berikutnya untuk menghasilkan byte berikutnya baik ciphertext atau plaintext.

Setiap elemen S ditukar dengan elemen lain setidaknya sekali setiap 256 iterasi.

I:= 0
J:=0

Saat generating output ;

I := (i+1) mod 256
J := (j+S [i]) mod 256
Nilai swap S [i] dan S [j]
K := S[(S [I] + S [j]) mod 256]
Keluaran K
Endwhile

Untuk menunjukkan cara kerja dari algoritma RC4 dapat dilihat dalam blok diagram pada Gambar 4. RC4menggunakan dua buah kotak substitusi (S-Box) array256 byte yang berisi permutasi dari bilangan 0 sampai 255 dan S-Box kedua yang berisi permutasi fungsi dari kunci dengan panjang yang variabel. Cara kerja algoritma RC4 yaitu inisialisasi Sbox pertama, S[0],S[1],...,S[255], dengan bilangan 0 sampai 255.Pertama isi secara berurutan S[0] = 0, S[1] = 1,..., S[255] = 255. Kemudian inisialisasi array lain (S-Box lain), misal array K dengan panjang 256. Isi array K dengan kunci yang diulangi sampai seluruh array K[0], K[1],...,K[255] terisi seluruhnya.

Berikut ini proses bagian dari metode RC4 [2]

1. *Key Setup* atau *Key Schedulling Algorithm* (KSA)
   Pada bagian ini terdapat tiga tahapan proses di dalamnya yaitu:
   a. Inisialisasi S-Box
      Pada tahapan ini S-Box akan diisi dengan nilai sesuai indeksnya untuk mendapatkan S-Box awal. Algoritmanya adalah sebagai berikut :
      - untuk i = 0 hingga i = 255 lakukan,
      - isikan s dengan nilai i,
      - tambahkan i dengan 1, kembali ke langkah 2.

Dari algoritma tersebut akan didapat urutan nilai S-Box yang direpresentasikan dalam Gambar 3.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Gambar 3. Modula Hasil Inisialisasi Sbox

Submitted: **16/12/2019**; Accepted: **16/01/2020**; Published: **25/01/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 2, Januari 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i2.1745

b. Menyimpan kunci dalam *Key Byte Array*
Pada tahapan ini, kunci (*key*) yang akan digunakan untuk mengenkripsi atau dekripsi akan dimasukkan ke dalam *array* berukuran 256 secara berulang sampai seluruh *array* terisi. Algoritmanya adalah sebagai berikut:
- isi j dengan 1,
- untuk i = 0 hingga i = 255 lakukan,
- jika j > panjang kunci maka,
- j diisi dengan nilai 1,
- akhir jika,
- isi k ke i dengan nilai ascii karakter kunci ke j,
- nilai j dinaikkan 1,
- tambahkan i dengan 1, kembali ke 2.

Dari algoritma tersebut akan didapatkan urutan *array key* misalkan sebagai berikut untuk kunci dengan panjang 8 karakter dengan urutan karakter dalam ASCII "109 97 104 98 98 97 104".

109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104
109	97	104	97	98	98	97	104	109	97	104	97	98	98	97	104

Gambar 4. Hasil Pengacakan Sbox

2. *Stream Generation* / *Pseudo Random Generation Alg*orithm (PRGA)
Pada tahapan ini akan dihasilkan nilai *pesuodorandom* yang akan dikenakan operasi XOR untuk menghasilkan *ciphertext* ataupun sebaliknya yaitu untuk menghasilkan *plaintext*. Algoritmanya adalah sebagai berikut:
- isi indeks i dan j dengan nilai 0,
- untuk i=0 hingga i=panjang *plaintext*, isi nilai i dengan hasil operasi (i+1) mod 256,
- isi nilai j dengan hasil operasi (j+s(i)) mod 256,
- tukar nilai s(i) dan s(j), isi nilai t dengan hasil operasi (s(i)+(s(j) mod 256))mod 256,
- isi nilai y dengan nilai s(t), nilai y dikenakan operasi XOR terhadap *plaintext*,
- tambahkan i dengan 1, kembali ke 2.

Dengan demikian akan dihasilkan *cipher*text dengan hasil XOR antar *sream key* dari S-Box dan *plaintext* secara berurutan.

3. HASIL DAN PEMBAHASAN

Pada bagian ini berisi analisa, hasil serta pembahasan dari topik penelitian, yang bisa di buat terlebih dahulu metodologi penelitian. Bagian ini juga merepresentasikan penjelasan yang berupa penjelasan, gambar, tabel dan lainnya.

3.1 Analisa Sistem

Adapun analisa kebutuhan sistem adalah sebagai berikut :
1. Aplikasi dapat memberikan fungsi otentifikasi *user* melalui proses *login.*
2. Aplikasi dapat memberikan layanan proses enkripsi (pengacakan isi data).
3. Aplikasi dapat memberikan layanan proses dekripsi (mengembalikan isi data seperti semula).
4. Aplikasi dapat memberikan layanan kompresi (pemampatan data).
5. Aplikasi dapat memberikan layanan *file* jika telah melakukan proses ataupun dekripsi.

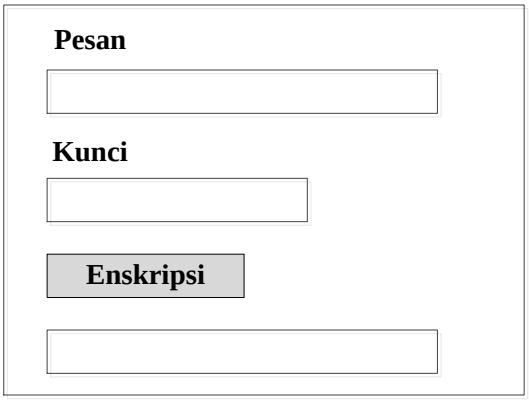
3.2 Perancangan Enkripsi

1. Tampilan Enkripsi

Submitted: **16/12/2019**; Accepted: **16/01/2020**; Published: **25/01/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 2, Januari 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i2.1745

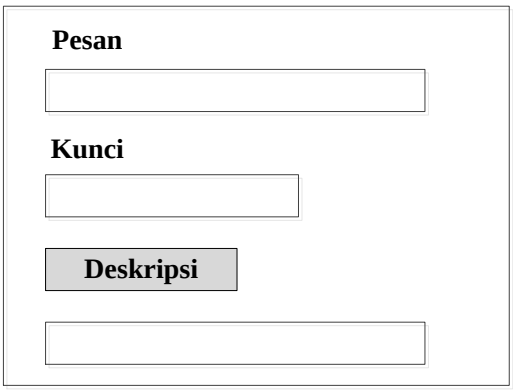
Perancangan ini akan ditampilkan sebuah tabel dari database tabel_mahasiswa. Di bagian atas halaman, terdapat tombol yang digunakan untuk mengenkripsi data dari tabel tersebut. Rancangan tampilan dapat dilihat pada Gambar 5.



Gambar 5. Rancangan Tampilan Enkripsi

2. Tampilan Deskripsi

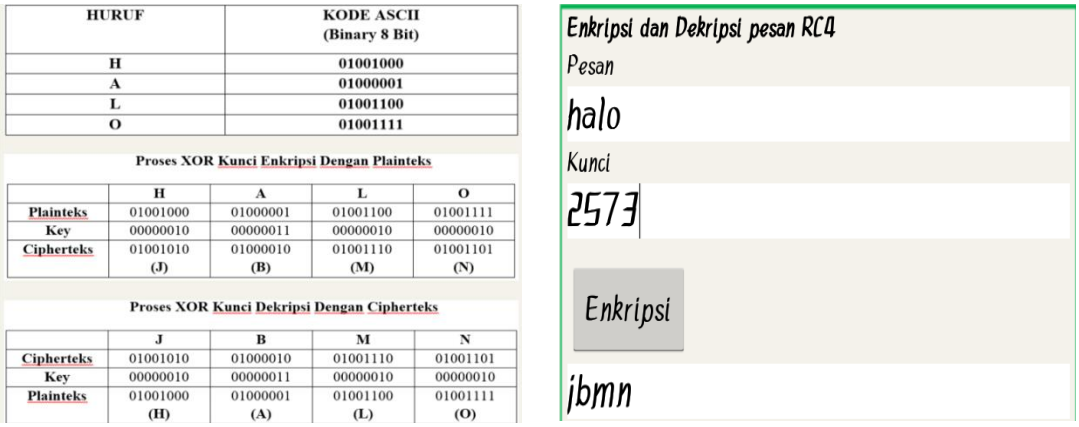
Perancangan ini akan ditampilkan sebuah tabel dari database tabel mahasiswa_enkripsi. Di bagian atas halaman, terdapat tombol yang digunakan untuk mendekripsi data dari tabel tersebut. Rancangan tampilan dapat dilihat pada Gambar 6.



Gambar 6. Rancangan Tampilan Deskripsi

3. Hasil Enkripsi

Tahap ini merupakan pengujian dari keseluruhan tahap-tahap yang telah dilalui dimulai dari analisis kebutuhan hingga tahap implementasi. Pengujian dilakukan terhadap hasil enkripsi (*ciphertext*) pada *database* dan hasil dekripsi. Pengujian terhadap enkripsi dilakukan dengan memproses apakah sebuah data dapat disandikan sedangkan pengujian terhadap dekripsi dilakukan dengan memproses apakah data yang terenkripsi dapat diubah menjadi seperti semula. Pengujian terhadap hasil enkripsi juga melihat apakah hasil enkripsi (*ciphertext*) mempunyai ukuran data yang sama dengan pesan asli atau tidak.



HURUF	KODE ASCII (Binary 8 Bit)
H	01001000
A	01000001
L	01001100
O	01001111

Proses XOR Kunci Enkripsi Dengan Plainteks

	H	A	L	O
Plainteks	01001000	01000001	01001100	01001111
Key	00000010	00000011	00000010	00000010
Cipherteks	01001010 (J)	01000010 (B)	01001110 (M)	01001101 (N)

Proses XOR Kunci Dekripsi Dengan Cipherteks

	J	B	M	N
Cipherteks	01001010	01000010	01001110	01001101
Key	00000010	00000011	00000010	00000010
Plainteks	01001000 (H)	01000001 (A)	01001100 (L)	01001111 (O)

Gambar 7. Hasil Enkripsi Metode RC4

Submitted: **16/12/2019**; Accepted: **16/01/2020**; Published: **25/01/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 2, Januari 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i2.1745

4. KESIMPULAN

Dari uraian yang telah disampaikan pada bab-bab sebelumnya mengenai pengamanan file(teks) menggunakan Algoritma RC4 ini, maka dapat diambil kesimpulan sebagai berikut:

1. Perangkat lunak ini dikembangkan sebagai pengaman data atau file (teks) berbasis bilangan biner.
2. Teknik kriptografi merupakan teknik yang sangat berperan/penting dalam mengamankan data.
3. Pesan tersandikan (ciphertext) yang dapat disisipkan ke dalam citra mempunyai nilai desimal tidak lebih dari 255 dalam kode ASCII dan belum ditemukan metode untuk memecahkan masalah tersebut.
4. Aplikasi Enkripsi dan Deskripsi dengan algoritma RC4 menggunakan sistem operasi windows 8 yang merubah sistemwindows32 yaitu bagian Shell32 [4].

REFERENCES

[1] M. Syahril and H. Jaya, "Aplikasi Steganografi Pengamanan Data Nasabah di Standard Chartered Bank Menggunakan Metode Least Significant Bit dan RC4," *Semin. Nas. Sains Teknol. Inf.*, pp. 505–509, 2019.
[2] A. P. Sutiono, "Algoritma RC4 sebagai Perkembangan Metode Kriptografi," *Bandung Inst. Teknol. Bandung*, pp. 1–6, 2011.
[3] R. T. Jurnal, "Penerapan Algoritma Rivert Code 4 (Rc 4) Pada Aplikasi Kriptografi Dokumen," *Petir*, vol. 11, no. 1, pp. 38–47, 2018, doi: 10.33322/petir.v11i1.6.
[4] J. Pseudocode, S. Informasi, U. M. Bengkulu, and H. Cipta, "Implementasi Algoritma Rc4," vol. V, 2018.

Submitted: **16/12/2019**; Accepted: **16/01/2020**; Published: **25/01/2020**