Journal of Computing and Informatics Research
ISSN 2808-375X (Media Online)
Vol 2, No 1, November 2022
Hal 1−9
DOI 10.47065/comforch.v2i1.348

Implementasi Algoritma RC4A Dalam Pengamanan *Citra* Digital

Yanti Nur Iman Giawa

Fakultas Ilmu Komputer dan Teknologi Informasi, Program Studi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email: yantyndruru233@gmail.com
Email Penulis Korespondensi: yantyndruru233@gmail.com

Abstrak−*Citra* digital yang bersifat pribadi dan rahasia sangat rentang terhadap penyadapan oleh pihak-pihak yang tidak bertanggung jawab, terutama bila *citra* tersebut didistribusikan melalui *internet*. Tindakan pencurian dan penyalahgunaan terhadap *citra* yang sifatnya rahasia tentu saja dapat merugikan pihak pemilik *citra*. Salah satu teknik untuk untuk mengurangi masalah diatas adalah dengan pemanfaatan teknik *kriptografi*. Teknik *kriptografi*dapat mengamankan *citra* dengan cara *enkripsicitra* digital dengan mengimplementasikan algoritma RC4A kemudian menggunakan kata kunci awal sebagai *inputan* dalam proses yang dilakukan untuk menghasilkan kata kunci baru yang lebih acak, berdasarkan *Key Schedule Algorithm (KSA) dan pseudo random generation Algorithm (PRGA)*. Untuk mengetahui perbandingan antara citra asli (*PlainImage*) dengan Citra asli (*CipherImage*) dilakukan pemrosesan dengan *MeanSquareError* (MSE) dan *PeakSiqnalToNoiseRation* (PSNR).Hasil penelitian ini adalah *citra* digital yang di *enkripsi* dengan algoritma RC4A di tinjau dari perbedaan waktu dan hasil *citra* setelah di*enkripsi* dimana waktu yang digunakan untuk melakukan *enkripsi* lebih singkat di bandingkan dengan *deskripsi* serta hasil *citra* sebelum di *enkripsi* tidak sama setelah di *enkripsi*.

Kata Kunci: Algoritma RC4A; Kriptoografi; PlainImage; Enkripsi

Abstract−Digital images that are private and confidential are very vulnerable to wiretapping by irresponsible parties, especially if the images are distributed via the internet. The act of theft and abuse of secret images of course can harm the image owner. One of the techniques to reduce the above problems is by using cryptographic techniques. Cryptographic techniques can secure images by means of digital image encryption by implementing the RC4A algorithm then using the initial keyword as input in the process carried out to generate new, more random keywords, based on the Key Schedule Algorithm. (KSA) and pseudo random generation algorithm (PRGA). To determine the comparison between the original image (PlainImage) and the original image (CipherImage), processing was carried out with MeanSquareError (MSE) and PeakSiqnalToNoiseRation (PSNR). The results of this study were digital images encrypted with the RC4A algorithm in terms of time differences and image results after encryption where the time used to encrypt is shorter than the description and the image results before encryption are not the same after encryption.

Keywords: RC4A Algorithm; Cryptography; PlainImage; Encryption

1. PENDAHULUAN

Keamanan data yang bersifat rahasia dan penting merupakan suatu yang sangat penting dijaga pada era perkembangan teknologi informasi saat ini. Keamanan data penting tidak hanya tergantung pada firewall dan intrusion detection system saja, tetapi keamanan data dari data itu sendiri merupakan hal yang sangat perlu diperhatikan. Citra digital merupakan salah satu jenis data yang saat ini banyak di gunakan dalam berkomunikasi baik secara langsung maupun melalui media sosial internet. Perkembangan pemanfaatan medai sosial saat ini sudah semakin memudahkn untuk saling berkomunikasi, bertukar informasi atau bertukar pesan baik bentuk teks, gambar (citra, audio maupun video. Dari sisi lain pekermbangan tersebut salah satu penyebab semakin mudahnya orang-orang tertentu untuk melakukan kejahatan dengan penyalahgunaan terhadap data atau informasi yang didistribusikan seperti tindakan pencurian data informasi, pemantauan informasi, manipulasi atau menggunakan informasi untuk kepentingan tertentu.

Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama, tetapi penulis menemukan penelitian lain dengan menggunakan metode yang sama antara lain "Analisa Algoritma Kriptografi RC4 Pada EnkripsiCitra Digital"[1], "Pengamanan Citra Digital Berdasarkan Modivikasi Algoritma RC4"[2], "Pengamanan File Teks Menggunakan Algoritma RC4"[3].

Kriptografi merupakan salah cara yang dapat digunakan untuk menjaga keamanan data atau informasi yang bersifat pribadi dan rahasia serta data yang sangat rentang terhadap penyadapan oleh pihak-pihak lain. Teknik kriptografi melakukan pengamanan data dengan menyandikan data asli menjadi simbol-simbol yang tidak dapat lagi dimengerti maknanya. Berdasarkan penelitian terdahulu, mengatakan bahwa teknik kriptografi dapat digunakan untuk menjaga dan mengamankan informasi pada saat didistribusikan dari suatu tempat ketempat lain. Teknik kriptografi bekerja dengan berbagai algoritma untuk melakukan proses pengamanan data, salah satunya adalah algoritma RC4A.

Algoritma RC4A merupakan algoritma stream cipher yang dirancang di RSA Security oleh Ron Riverst tahun 1987. Sifat kunci dalam algoritma RC4A adalah simetris serta melakukan proses senkrip siplain per digit atau byte per bytedengan operasi biner. Algoritma RC4A bekerja dengan tiga tahap utama yaitu Key Scheduling Algorithm (KSA), Pseudo Random Generation Algorithm (PRGA) dan proses Enkripsidan dekripsi. Berdasarkan penelitian terdahulu, mengatakan bahwa algoritma ini merupakan modifikasi algoritma kriptografi byteoriented stream cipher dari cipher RC4 yang tingkat keamanannya lebih kuat dibandingkan dengan RC4.

Penelitian ini menguraikan bagaimana mengamankan Citra digital berdasarkan algoritma RC4A, sehingga keamanan data Citra digital lebih terjamin. Agar proses pengamanan data Citra digital lebih mudah dilakukan, maka dibangun sebuah aplikasi pengamanan Citra digital yang bekerja berdasarkan algoritma RC4A

Journal of Computing and Informatics Research
ISSN 2808-375X (Media Online)
Vol 2, No 1, November 2022
Hal 1−9
DOI 10.47065/comforch.v2i1.348

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi (Cryptography) berasal dari bahasa Yunani, terdiri dari dua suku kata yaitu kripto dan graphia. Kripto artinya menyembunyikan, sedangkan graphia artinya tulisan. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, seperti kerahasian data, keabsahan data, integritas data, serta autentikasi data. Tetapi tidak semua aspek keamanan informasi dapat diselesaikan dengan kriptograf. Kriptografi dapat pula diartikan sebagai ilmu atau seni untuk menjaga keamanan pesan. Ketika suatu pesan dikirim dari suatu tempat ke tempat lain, isi pesan tersebut mungkin dapat disadap oleh pihak lain yang tidak berhak untuk mengetahui isi pesan tersebut. Untuk menjaga pesan, maka pesan tersebut dapat diubah menjadi sebuah kode yang tidak dapat dimengerti pihak lain. Enkripsiadalah sebuah proses penyandian yang melakukan perubahan sebuah kode (pesan) dari yang bisa dimengerti (plaintext) menjadi sebuah kode yang tidak bisa dimengerti (chipertext). Sedangkan proses kebalikannya untuk mengubah cipertext menjadi plaintext disebut dekripsi. Proses Enkripsidan Deskripsi memerlukan suatu mekanisme dan kunci tertentu[5].

2.2 Algoritma RC4A

RC4A merupakan sebuah upaya untuk meningkatkan keamanan dari RC4 tanpa mengurangi efisiensi, RC4A adalah stream cipher yang berorientasi byte. Tahap pembentukan dari RC4A lebih efisien dibanding RC4, tetapi tahap inisialisasinya memerlukan setidaknya dua kali proses inisialisasi dari RC4. Berikut adalah algoritma KSA dari RC4A.
for i = 0 ... l-1
WK[i] = RC4_PRGA(S1)
RC4_KSA(WK,S2)
Berikut adalah algoritma PRGA dari RC4A
Initialization:
i = 0
j1= j2 = 0
Generation loop
i = i + 1
j1= j1 + S1[i]
Swap (S1[i], S1[j1])
Output z = S2[S1[i] + S1[j1]]
j2 = j2 + S2[i]
Swap (S2[i], S2[j2])
Output z = S1[S2[i] + S2[j2]]

RC4A menggunakan dua state array, S_1 dan S_2, dan tiga buah indeks i, j_1, dan j_2. RC4A menggunakan KSA yang sama dengan RC4 kecuali satu hal dimana KSA digunakan dua kali, masing-masing sekali untuk S_1 dan S_2. Semua operasi aritmetika dihitung dengan modulo 256.
Proses KSA dari RC4A terdiri atas 2 bagian yaitu :

1. KSA dengan masukan K dan S_1.
2. KSA dengan masukan WK dan S_2.

WK dihasilkan dari PRGA milik RC4. l di sini merupakan panjang kunci dalam byte. Setelah S_1 dan S_2 diperoleh dari proses KSA, selanjutnya digunakan kembali dalam PRGA. Tiap putaran dalam PRGA menghasilkan 2 byteoutput[6].

3. HASIL DAN PEMBAHASAN

Informasi yang didistribusikan dalam bentuk Citra digital sangat penting untuk diamankan agar kerahasiaannya tetap terjaga dari serangan berbagai serangan yang dapat dilakukan oleh pihak yang tidak bertanggung jawab. Salah satu jenis serangan yang dapat dilakukan terhadap data yang telah diamankan adalah menggunakan teknik bruteforce. Serangan bruteforce adalah serangan yang digunakan untuk mencoba menDeskripsi data yang dienkripsi. Apabila pengamanan Citra digital diabaikan, maka akan sangat merugikan pihak pengirim maupun penerima pesan, karena penyerang dapat saja melakukan pencurian, modifikasi bahkan penyalahgunaan Citra digital.

Salah satu teknik yang dapat digunakan untuk mengamankan data penting atau rahasia yang didistribusikan adalah memanfaatkan teknik kriptografi. Teknik kriptografi mengamankan data rahasia dengan merubah data asli menjadi simbol-simbol atau nilai-nilai lain yang tidak dapat dipahami lagi oleh orang lain, sehingga makna aslinya sulit dikenali. Teknik pengamanan data rahasia berdasarkan teknik kriptografi dilakukan berdasarkan Algoritma yang dimilikinya.

Penerapan teknik kriptografi dalam mengamankan Citra digital dilakukan dengan merubah nilai-nilai warna pixelCitra digital berdasarkan Algoritma yang dimiliki oleh teknik kriptografi. Salah satu Algoritma kriptografi yang dapat digunakan dalam mengamankan data rahasia adalah Algoritma RC4A. Nilai-nilai warna Citra digital asli akan dirubah sehingga menghasilkan nilai-nilai warna baru yang berakibat terhadap perubahan warna Citra digital asli, sehingga Citra digital yang dihasilkan tidak lagi memperlihatkan pola warna dari Citra asli. Perubahan warna ini tentu

Journal of Computing and Informatics Research
ISSN 2808-375X (Media Online)
Vol 2, No 1, November 2022
Hal 1−9
DOI 10.47065/comforch.v2i1.348

akan menyulitkan pihak lain untuk dapat langsung mengetahui pola Citra asli, sehingga dapat meminimalkan tindakan-tindakan penyalahgunaan.

3.1 Penerapan Algoritma RC4A

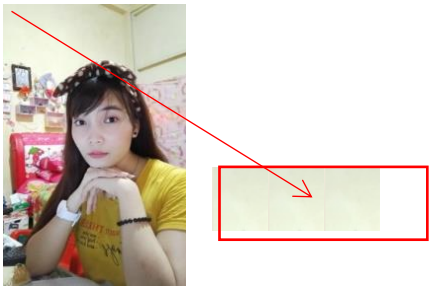
Algoritma RC4A merupakan pengembangan dari Algoritma RC4 yang memiliki kelebihan untuk menghasilkan kata kunci yang acak sehingga akan menambah kerumitan dalam pemecahan Algoritma ini. Algoritma ini menggunakan kata kunci awal sebagai input dalam proses yang dilakukan untuk menghasilkan kata kunci baru yang lebih acak. Secara umum, tahap yang dilakukan proses pengamanan data rahasia berdasarkan Algoritma RC4A adalah :

1. Proses Pembangkitan Kunci
   Kunci awal yang diinput oleh pengguna menjadi input yang digunakan dalam proses pembangkitan kunci yang baru. Karakter kunci awal ini akan dibagikan kepada penerima Citra digital agar penerima dapat melakukan proses Dekripsi. Jumlah karakter kata kunci yang baru akan dibangkitkan sejumlah data (dalam kasus ini jumlah pixelCitra) yang akan dienkripsi atau Dekripsi. Proses pembangkitan kunci berdasarkan Algoritma RC4A, harus melalui dua proses utama, yaitu :
   a. Key Schedule Algorithm (KSA)
      Proses pembentukan dan pengacakan sejumlah nilai array untuk mendapatkan input pada proses PRGA. Proses ini membutuhkan nilai kunci dan nilai-nilai array dari 0 sampai 255.
   b. Pseudo Random Generation Algorithm (PRGA)
      Proses ini merupakan proses yang dilakukan untuk mendapatkan nilai-nilai kunci yang baru. Proses ini membutuhkan nilai-nilai arrai KSA sebagai input untuk menghasilkan kunci baru yang digunakan dalam proses enkripsi dan Dekripsi.
2. Proses Enkripsi
   Proses enkripsi Citra digital berdasarkan Algoritma RC4A, dilakukan dengan melakukan operasi XOR terhadap masing-masing nilai warna Citra digital dengan kunci yang menjadi pasangannya, sehingga didapatkan nilai-nilai warna yang baru yang selanjutnya disebut dengan cipherimage.
3. Proses Dekripsi
   Proses ini dilakukan oleh penerima cipherimage yang diawali dengan proses pembangkitan kunci dengan cara yang sama seperti di atas. Proses Dekripsi juga dilakukan dengan cara yang sama seperti pada proses enkripsi. Nilai-nilai pixel cipherimage menjadi input dalam proses ini, kemudian melakukan operasi XOR antara nilai-nilai pixel cipherimage dengan nilai-nilai kunci.

Citradigital yang digunakan pada contoh kasus penerapan Algoritma RC4A dalam penelitian ini adalah Citra warna dengan resolusi 315 x 190 pixel, kemudian untuk mempermudah proses yangdilakukan, maka diimplementasikan terhadap 3 pixelCitra sampel. Jumlah 3 pixel dari Citra sampel akan menghasilkan 9 elemen warna, karena setiap pixel pada Citra warna memiliki 3 elemen warna (red, green dan blue).

Gambar 1. Citra Plainimage dengan resolusi 315x190

Berdasarkan gambar 1, akan diambil 3 pixel sebagai sampel dalam perhitungan manual dimulai dari pixel 0 sampai 3. Tiga pixel tersebut akan di ambil nilai desimal warna pada setiap elemen warna pixelnya.



Gambar 2. Plainimage diambil 3 pixel

Nilai elemen warna dari tiga pixelplainimage sampel di atas di ambil dengan menggunakan software Matlab :

```
RGB=imread (‘yanti.jpg’);
figure,imshow(yanti);
```

Journal of Computing and Informatics Research
ISSN 2808-375X (Media Online)
Vol 2, No 1, November 2022
Hal 1−9
DOI 10.47065/comforch.v2i1.348

```
R=(;,;,;1);
G=(;,;,;2);
B=(;,;,;3);
merahgray2=0.3*red+0.5*green+0.5*blue;
figure,imhist(merahgray2);
```

sehingga diperoleh nilai warna dari Citra sampel adalah :

Tabel 1. Nilai RGB CitraSample

Pixel 0			Pixel 1			Pixel 2		
R	G	B	R	G	B	R	G	B
154	96	60	133	58	33	71	26	28

Berdasarkan tabel 1. diatas maka nilai desimal 154,96,60,113,58,33,71,26,28. Kata kunci awal yang digunakan adalah YANTI. Kata kunci awal ini dibuat dalam bentuk array agar mempermudah penggunaanya dalam proses pembangkitan kunci baru.

Tabel 2. Array Kunci Awal

Index	0	1	2	3	4
Karakter Kunci	Y	A	N	T	I
Desimal	89	65	78	84	73

1. Proses Pembangkitan Kunci

Proses pembangkitan kunci berdasarkan Algoritma RC4A, terdiri dari dua proses utama, yaitu :

a. Proses KeySchedulingAlgorithm (KSA)

Pembentukan Tabel Array S, dilakukan berdasaran pseudocode-nya dan menghasilkan array dengan nilai 0 sampai dengan 255, sehingga tabel array S sebagai berikut:

Tabel 3. Tabel Array S

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
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Seluruh nilai-nilai pada tabel array S di atas akan diacak dengan melibatkan nilai-nilai array tabel kunci awal. Adapun peseudocode yang digunakan untuk mengacak seluruh nilai-nilai tabel S-Box di atas adalah :

```
j=0
for i = 0 to 256
        j = (j + S[i] + Key[i mod keyleght]) mod 256
        Swapvalues of S[i] and S[j]
endfor
```

berdasarkan tabel 3.2 (tabel array kunci), diperoleh bahwa panjang kunci (key length) adalah 5.

untuk iterasi i = 0

j = 0
j = (j + S[i] + key[i mod keylength]) mod 256
j = (0 + S[0] + key[0 mod 5]) mod 256
j = (0 + 0 + key[89 mod 5]) mod 256
j = (0 + key[4]) mod 256
j = (0 + 73) mod 256
j = 73 mod 256
j = 73

swapvalues of S[i] and S[j]

tukarkan (swap) nilai S[0] dengan S[73]

nilai S[0] = 0 dan nilai S[73] adalah 73, sehinga setelah ditukarkan menjadi S[0] = 73 dan S[73] = 0.

untuk iterasi i = 1

j = 73 (nilai j proses sebelumnya atau nilai j pada iterasi i=0)
j = (7 + S[i] + key[i mod keylength]) mod 256
j = (73 + S[1] + key[1 mod 5]) mod 256
j = (73 + 1 + key[64 mod 5]) mod 256
j = (74 + key[4]) mod 256
j = (74 + 73) mod 256
j = 147 mod 256
j = 147

Journal of Computing and Informatics Research
ISSN 2808-375X (Media Online)
Vol 2, No 1, November 2022
Hal 1−9
DOI 10.47065/comforch.v2i1.348

swapvalues of S[i] and S[j]
tukarkan (swap) nilai S[1] dengan S[147]
nilai S[1] = 1 dan nilai S[147] adalah 147, sehinga setelah ditukarkan menjadi S[1] = 147 dan S[147] = 1.
Proses ini akan dilakukan sampai pada iterasi i = 256, sehingga seluruh posisi nilai pada array S teracak. Hasil proses pengacakan keseluruhan adalah :

Tabel 4. Hasil Proses KSA

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

2. PseudoRandomGenerationAlgorithm (PRGA)
   Proses PRGA menggunakan hasil permutasi tabel Array S pada Tabel 3.3 di atas. Proses ini dilakukan untuk menghasilkan key stream baru yang akan digunakan pada proses enkripsi ataupun Dekripsi. Proses iterasi pada PRGA dilakukan sebanyak jumlah nilai-nilai elemen warna Citra yang akan dienkripsi atau diDekripsi (dalam contoh kasus ini jumlah elemen warna Citra adalah 9) yang berarti iterasi PRGA dilakukan sebanyak 9 kali untuk menghasilkan kunci masing-masing nilai elemen Citra. Proses PRGA dilakukan dengan peseudocode berikut ini :

   i =j1 = j2 = 0
   output keystream generation loop
   i = i + 1
   j1 = (j1 + S1[i]) mod 256
   swap S1[i] and S1[j1]
   t1 = (S1[i] + S1[j1]) mod 256
   output S2[t1]
   j2 = (j2 + S2[i]) mod 256
   swap S2[i] and S2[j2]
   t2 = (S2[i] + S2[j2]) mod 256
   output S1[t2]
   iterasi i = 0 (untuk mencari kunci ke-0), maka :
   i = i + 1
   i = 73 + 1
   i = 74
   j1 = (0 + S1[1]) mod 256
   j1 = 73
   tukarkan nilai array S1[1] dengan S1[j1]
   t1 = (S1[1] + S1[nilai j1]) mod 256
   output S2[nilai hasil t1]
   j2 = (nilai j2 awal + S2[1]) mod 256
   tukarkan nilai Array S2[1] dengan S2[nilai j2]
   t2 = (S2[1] + S2[nilai hasil j2]) mod 256
   output S1[nilai t2], nilainya adalah 73
   nilai ini akan digunakan sebagai karakter ke-0 dari kunci baru
   iterasi i = 1 (untuk mencari kunci ke-1), maka :
   i = i + 1
   i = 73 + 1
   i = 74
   j1 = (74 + 73[1]) mod 256
   j1 = 147
   tukarkan nilai array S1[1] dengan S1[j1]

Journal of Computing and Informatics Research
ISSN 2808-375X (Media Online)
Vol 2, No 1, November 2022
Hal 1−9
DOI 10.47065/comforch.v2i1.348

t1 = (73[1] + 74[nilai j1]) mod 256
output S2[nilai hasil t1]
j2 = (nilai j2 awal + S2[1]) mod 256
tukarkan nilai Array S2[1] dengan S2[nilai j2]
t2 = (S2[1] + S2[nilai hasil j2]) mod 256
output S1[nilai t2], nilainya adalah 147
nilai ini akan digunakan sebagai karakter ke-1dari kunci baru
iterasi i = 2 (untuk mencari kunci ke-2), maka :
i = i + 1
i = 147 + 1
i = 148
j1 = (147 + 148[1]) mod 256
j1 = 40
tukarkan nilai array S2[1] dengan S1[j2]
t1 = (40[1] + 74[nilai j2]) mod 256
output S2[nilai hasil t2]
j3 = (nilai j3 awal + S2[1]) mod 256
tukarkan nilai Array S2[1] dengan S3[nilai j2]
t3 = (S3[1] + S3[nilai hasil j3]) mod 256
output S1[nilai t3], nilainya adalah 40
nilai ini akan digunakan sebagai karakter ke-2 dari kunci baru
Proses untuk iterasi ini di lakukan hingga 8 kali (9 kali iterasi) sehingga key stream untuk keseluruhan adalah

Tabel 5. Key Stream

Index	Stream Key	
	Desimal	Biner
K0	73	01001001
K1	147	10010011
K2	40	00101000
K3	83	01010011
K4	170	10101010
K5	89	01011001
K6	184	10111000
K7	119	01110111
K8	246	11110110

Berdasarkan jumlah keystream yang dihasilkan di atas, maka disimpulkan bahwa jumlah keystream yang dibangkitkan berbanding lurus atau sama dengan jumlah banyaknya pixelCitra digital yang akan disandikan. Hal ini terjadi karena RC4A mengenkripsi setiap pixelCitra dengan kunci yang berbeda. Konsep ini hampir sama dengan konsep Algoritmavegeneerecipher.

3. Proses Enkripsi
Proses enkripsi di awali dengan mengkonversi nilai-nilai warna elemen pixelCitra ke bilangan biner. Hal yang sama juga dilakukan untuk karakter keystream.
   a. Nilai Decimal RGB plainimage Pada Citra di konversi ke biner, dengan cara.
   154/2 = 76 Sisah bagi adalah 1
   76/2 = 38 Sisah bagi adalah 0
   38/2 = 19 Sisah bagi adalah 0
   19/2 = 9 Sisah bagi adalah 1
   9/2 = 4 Sisah bagi adalah 1
   4/2 = 2 siah bagi adalah 0
   2/2 = 0 sisah bagi adalah 1
   0/2 = 0 sisah bagi adalah 0
   Langkah ini di lakukan berulang pada masing-masing nilai decimalplainimage hingga mendapatkan hasil seperti tabel 6. Berikut ini.

Tabel 6. Hasil Konversi Nilai pixelCitraDecimal ke Biner

Pixel	Warna	Dec	Biner	P[i]
0	R	153	10011010	P[0]
	G	96	01100000	P[1]
	B	60	00111100	P[2]
1	R	133	10000101	P[3]
	G	58	00111011	P[4]

Journal of Computing and Informatics Research
ISSN 2808-375X (Media Online)
Vol 2, No 1, November 2022
Hal 1–9
DOI 10.47065/comforch.v2i1.348

Pixel	Warna	Dec	Biner	P[i]
	B	33	00100001	P[5]
	R	71	00100001	P[6]
2	G	26	00011010	P[7]
	B	28	00011010	P[8]

Selanjutnya melakukan proses enkripsi dengan formula Ci = Pi ⊕ Ki, masing-masing biner elemen warna setiap pixel di XOR dengan masing-masing keystream yang telah dihasilkan pada proses PRGA

Enkripsi Blok Pixel 0 :

Biner Warna R = 10011010
Biner Kunci 0 = 01001001⊕
Plain Warna R = 11010011 = $211_{decimal}$
Biner Warna G = 01100000
Biner Kunci 1 = 10010011⊕
Plain Warna G = 11110011 = $243_{Decimal}$
Biner Warna B = 00111100
Biner Kunci 2 = 00101000⊕
Plain Warna B = 00010100 = $20_{Decimal}$

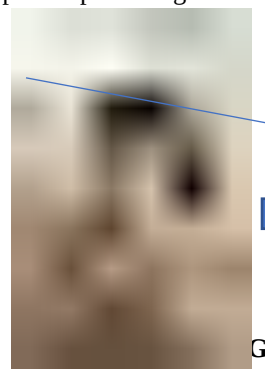
Tabel 7. Hasil Proses EnkripsiPlainimage

Pixel	Warna	Plainimage		Operasi XOR	Biner Kunci K[i]	CipherImage	
		Dec	Biner			Biner	Dec
0	R	153	10011010	⊕	01001001	11010011	121
	G	96	01100000	⊕	10010011	11110011	243
	B	60	00111100	⊕	00101000	00010100	20
1	R	133	10000101	⊕	01010011	11010110	214
	G	58	00111011	⊕	10101010	10010001	145
	B	33	00100001	⊕	01011001	01111000	120
2	R	71	01000111	⊕	10111000	11111111	255
	G	26	00011010	⊕	01110111	01101101	109
	B	28	00011010	⊕	11110110	11101010	234

Sehingga, warna Citra yang dihasilkan akibat perubahan nilai setiap elemen warna pixelplainimage adalah :

4. Proses Dekripsi

Penerima cipherimage akan menginput kata kunci awal, kemudian akan dilakukan proses KSA dan proses PRGA untuk membangkitkan kunci yang baru yang akan digunakan dalam proses Dekripsicipherimage menjadi plainimage. Proses KSA dan proses PRNG yang dilakukan padatahap Dekripsi sama seperti yang dilakukan pada proses enkripsi, karena dua proses ini dilakukan hanya untuk membangkitkan kunci yang baru, sehingga key stream yang dihasilkan sama seperti key stream pada proses enkripsi Proses Dekripsi dilakukan berdasarkan formual Pi = Ci ⊕ Ki. Masing-masing nilai biner cipherimage akan di XOR dengan nilai biner kunci yang didapatkan dari proses PRGA untuk mendapatkan plainimage.



Pixel	Warna	Decimal ChiperImage
0	R	211
	G	234
	B	20
1	R	214
	G	145
	B	120
2	R	255
	G	109
	B	234

Gambar 3. Tampilan CipherImage dan nilai pixel-nya

Nilai-nilai pixelcipherimage di atas dikonversi menjadi biner, sehingga dihasilkan :

Tabel 8. Nilai Biner PixelCipherImage

Pixel	Warna	Chiper	
		Decimal	Biner
0	R	211	11010011
	G	211	11010011

Journal of Computing and Informatics Research
ISSN 2808-375X (Media Online)
Vol 2, No 1, November 2022
Hal 1−9
DOI 10.47065/comforch.v2i1.348

Pixel	Warna	Chiper	
		Decimal	Biner
	B	234	11110011
	R	20	00010100
1	G	214	11010110
	B	145	10010001
	R	120	01111000
2	G	255	11111111
	B	109	01101101

Langkah selanjutnya adalah melakukan proses Dekripsi dengan melakukan proses XOR antara biner elemen warna setiap pixelcipherimage dengan biner karakter key stream (tabel 3.5)

Dekripsi Blok CipherImage Pixel 0 :

Biner Warna R = 11010011
Biner Kunci 0 = 01001001⊕
Plain Warna R = 10011010 = $153_{decimal}$
Biner Warna G = 11110011
Biner Kunci 1 = 10010011⊕
Plain Warna G = 01100000 = 96Decimal
Biner Warna B = 00010100
Biner Kunci 2 = 00101000⊕
Plain Warna B = 00111100 =60Decima

Untuk mencari nilai cipher elemen warna pixel di lakukan dengan cara yang sama seperti di atas sehingga di peroleh nilai cipher image seluruhnya adalah :

Tabel 9. Hasil Proses DekripsiCipherImage

Pixel	Warna	CipherImage		Operasi XOR	Biner Kunci K[i]	Plainimage	
		Dec	Biner			Biner	Dec
	R	121	11010011	⊕	01001001	10011010	153
0	G	243	11110011	⊕	10010011	01100000	96
	B	20	00010100	⊕	00101000	00111100	60
	R	214	11010110	⊕	01010011	10000101	133
1	G	145	10010001	⊕	10101010	00111011	58
	B	120	01111000	⊕	01011001	00100001	33
	R	255	11111111	⊕	10111000	00100001	71
2	G	109	01101101	⊕	01110111	00011010	26
	B	234	11101010	⊕	11110110	00011010	28

Berdasarkan nilai-nilai image yang di dapatkan, setelah di konversi menjadi Citra, maka di dapat Citra yang sama seperti Citra awal (plainimage). Nilai-nilai dipetakan menjadi Citra yang baru yang selanjutnya di sebut dengan plainimage.

5. Proses Perhitungan Rata-rata Perubahan Pixel, MSE dan PSNR

Tabel 10. Nilai Citra Awal (PlainImage)

R	G	B
153	96	60
133	58	33
71	26	28

Tabel 11. Nilai Citra Akhir (CipherImage)

R	G	B
211	234	20
214	145	120
255	109	234

a. Menghitung rata-rata perubahan nilai setiap Pixel

$$= \frac{\begin{array}{l}(211-153)\ +(234-96)\ +(20-60)\ +(214-133)\ + \\ (145-58)\ +(120-33)\ +(255-171)\ +(109-26)\ + \\ (234-28)\end{array}}{3x3}$$

ISSN 2808-375X (Media Online)
Vol 2, No 1, November 2022
Hal 1–9
DOI 10.47065/comforch.v2i1.348

$$\text{Rata} - \text{rata Perubahan } Pixel = \frac{784}{9}$$

$$\text{Rata} - \text{rata Perubahan } Pixel = 87{,}11 \text{ atau } 87\%$$

b. Menghitung Nilai MSE

Proses Mean Square Error dilakukan untuk mencari berapa besar nilai error yang terjadi antara nilai PlainImage dengan CipherImage.Proses perhitungan MSE adalah dengan mengurangi nilai CipherImage dengan PlainImage seperti berikut :

$$MSE = \frac{\begin{array}{c}(211-153)^2+(234-96)^2+(20-60)^2+(214-133)^2+\\(145-58)^2+(120-33)^2+(255-171)^2+(109-26)^2+\\(234-28)^2\end{array}}{3x3}$$

$$MSE = 3364 + 19044 + 1600 + 6561 + 7569 + 7569 + 7056 + 6889 + 42436$$

$$MSE = \frac{102088}{9} = 11343{,}11$$

c. Menghitung nilai PSNR

$$\text{PSNR} = 10 * \text{Log}\ \frac{153}{\sqrt{11343{,}11}} = 1{,}57\text{dB}$$

Berdasarkan proses pengujian nilai rata-rata perubahan pixel, maka diperoleh bahwa perubahan nilai pixel citra awal sangat signifikan yaitu 87%, sedangkan nilai error yang terjadi (nilai MSE) adalah 11343,11 yang mengindikasikan bahwa perubahan nilai cetra awal sangat tinggia.Berdasarkan pengujian nilai PSNR menunjukkan bahwa kualitas cipherimage yang dihasilkan sudah sangat berbeda dengan plainimage dengan nilai 1,57dB (dibawah 30dB), sehingga cipherimage yang dihasilkan tidak lagi memperlihatkan pola dari citra asli (plainimage).

4. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut Berdasarkan hasil penelitian yang dilakukan, maka pengkombinasian algoritma kriptografi dengan metode steganografi dapat mengoptimalkan keamanan data penting atau rahasia. Berdasarkan hasil pengujian nilai MSE dan nilai PSNRdidapatkan bahwa cipherImage yang dihasilkan dari proses enkripsi berdasarkan algoritma RC4A sangat jauh berbeda dengan citra asli (plainimage), dimana nilai kemiripan plainimage dengan cipherimage berada di bawah 30dB, sehingga sangat sulit untuk mengetahui pola citra asli.

REFERENCES

[1] F. Kurniadi, Irwansyah et al., "Analisa algoritma kriptografi rc4 pada enkripsi citra digital," J. Ilm. R B, vol. 10, no. 12, hal. 1–7, 2014, [Daring]. Tersedia pada: http://jurnal.unived.ac.id/index.php/jmi/article/download/226/203%5Cnhttp://mfile.narotama.ac.id/files/Zakki Falani/Magang PDF/04104008 SITI MARIAM.pdf%5Cnhttp://library.umn.ac.id/jurnal/public/uploads/papers/pdf/ce8a2e6ffbf47103cfb493d743d173d3.pdf.

[2] T. Zebua dan E. Ndruru, "Pengamanan Citra Digital Berdasarkan Modifikasi Algoritma RC4," J. Teknol. Inf. dan Ilmu Komput., vol. 4, no. 4, hal. 275, 2017, doi: 10.25126/jtiik.201744474.

[3] D. R. Saragi, J. M. Gultom, J. A. Tampubolon, dan I. Gunawan, "Pengamanan Data File Teks (Word) Menggunakan Algoritma RC4," J. Sist. Komput. dan Inform., vol. 1, no. 2, hal. 114, 2020, doi: 10.30865/json.v1i2.1745.

[4] A. M. Hasibuan, "Rancang Bangun Aplikasi Keamanan Data Menggunakan Metode AES Pada Smartphone," MEANS (Media Inf. Anal. dan Sist., vol. 2, no. 1, hal. 29–35, 2017, doi: 10.17605/JMEANS.V2I1.20.

[5] I. Pendahuluan dan A. A. M. Hellman, "Perancangan Aplikasi Keamanan Data Teks," vol. 16, hal. 302–305, 2017.

[6] N. Hayati, "Implementasi Algoritma RC4A dan MD5 untuk Menjamin Confidentiality dan Integrity pada File Teks," J. Penelit. Tek. Inform., vol. 1, no. April, hal. 51–57, 2017.

[7] M. K. Sri Ratna Sulistiyanti, FX Arinto Setyawan, "Pengolahan Citra Dasar dan Contoh Penerapannya," עלון הנוטע, vol. 66, hal. 37–39, 2012.

[8] D. Putra, "Pengolahan Citra Digital," in Pengolahan Citra Digital, Westrining., Yogyakarta: CV.Andi Offset, 2010, hal. 38–39.

[9] K. Margi S dan S. Pendawa W, "Analisa Dan Penerapan Metode Single Exponential Smoothing Untuk Prediksi Penjualan Pada Periode Tertentu," Pros. SNATIF, no. 1998, hal. 259–266, 2015.

[10] S. Santoso dan R. Nurmalina, "Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna Pengembangan Kampus Cerdas (Studi Kasus Politeknik Negeri Tanah Laut)," J. Integr., vol. 9, no. 1, hal. 84–91, 2017.

[11] A. R. Darlis, P. Trisapto, dan L. Jambola, "Perancangan Dan Realisasi Sistem Pentransmisian Short Message Dan Sinyal Digital Pada Modulator Demodulator (Modem) Binary Phase Shift Keying (BPSK) Berbasis Matlab 7.4.," J. Rekayasa, vol. 17, no. 1, hal. 1–12, 2013.

[12] A. J. Nathan dan A. Scobell, "How China sees America," Foreign Affairs, vol. 91, no. 5. hal. 1689–1699, 2012, doi: 10.1017/CBO9781107415324.004.

[13] B. Data, "Rancang Bangun Aplikasi Toko Menggunakan Visual Basic 9.0 Â€Œstudi Kasus Roberta Superstoreâ€," J. Tek. Elektro dan Komput., vol. 1, no. 2, hal. 1–7, 2012, doi: 10.35793/jtek.1.2.2012.601.