Journal Global Tecnology Computer
Vol 1, No. 1, Desember 2021, Hal 1 - 12
ISSN: **2809-6118** (Online - Elektronik)
Website https://ejurnal.seminar-id.com/index.php/jogtc

Penerapan Metode Algoritma Spread Spectrum Untuk Menyembunyikan Pesan Terenkripsi Algoritma RC4A

Hendri Wahyu Kleaver Lase

Prodi Studi Teknik Informatika, Fakultas Ilmu Komputer Dan Teknologi Informasi, Universitas Budi Darma
Jl. Sisingamangaraja No. 338, Medan, Sumatera Utara, Indonesia
Email: cleverhendrik@gmail.com

Abstrak-Keamanan data pada saat ini sangat penting dilakukan,karena peningkatan penggunaan teknologi media komunikasi yang sangat meningkat.Teknik Kriptografi Teknik Stenografi merupakan teknik yang dapat di gunakan untuk mengamankan data. Algoritma RC4A merupakan salah satu algoritma kriptografi yang dapat digunakan dalam pengamanan data. Teknik stenografi dapat dimanfaatkan untuk mengoptimalkan keamanan terhadap data yang telah diamankan berdasarkan salah satu algoritma dari teknik kriptografi.Salah satu metode Penelitian ini menguraikan bagaimana menyembunyikan pesan terenkripsi berdasarkan algoritma RC4A kedalam objek citra digital yang dilakukan berdasarkan metode Spread Spectrum, sehingga keamanan data rahasia lebih terjaga.

Kata kunci: Kriptografi, RC4A, Steganografi, Spread Spectrum, Citra

Abstract-Nowadays, data security is very important to do, because of the increasing use of communication media technology. Cryptography Technique Stenography technique is a technique that can be used to secure data. The RC4A algorithm is a cryptographic algorithm that can be used in data security. Stenography techniques can be used to optimize the security of data that has been secured based on one of the algorithms of cryptographic techniques. One of the methods of this research describes how to hide encrypted messages based on the RC4A algorithm into digital image objects that are carried out based on the Spread Spectrum method, so that the security of confidential data is more secure. .

Keywords: Cryptography, RC4A, Steganography, Spread Spectrum, Image

1. PENDAHULUAN

Keamanan data pada saat ini sangat penting dilakukan, karena peningkatan penggunaan teknologi media komunikasi. Pemanfaatan berbagai media komunikasi yang sudah ada saat ini telah memudahkan pengguna untuk melakukan pendistribusian pesan atau informasi seperti yang besifat pribadi atau rahasia. Bila data penting tidak diamankan, maka informasi tesebut dapat dengan mudah dimanfaatkan oleh pihak-pihak yang tidak berkepentingan, misalnya saja melakukan penyadapan kemudian memanipulasi pesan tersebut. Bila informasi jatuh ketangan pihak lain dapat menimbulkan kerugian bagi pemilik informasi.Berdasarkan penelitian terlebih dahulu mengatakan bahwa masalah keamanan dan kerahasiaan data dan informasi merupakan suatu hal yang penting. Salah satu cara menjaga keamanan dan kerahasiaan data dan informasi adalah dengan teknik enkripsi dan dekripsi atau yang dikenal juga dengan kriptografi[1]. Umumya ada tiga teknik keamanan data yang umum digunakan yaitu teknik kriptografi, teknik stenografi, dan teknik *Watermaking*.

Kriptografi *(cryptography)* berasal dari Bahasa Yunani, yaitu *cryptos* dan graphia yang berarti 'penulisan rahasia'. Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan yang dikirim oleh pengirim dapat disampaikan kepada penerima dengan aman.Kriptografi dalam proses mengambil pesan/message dan menggunakan beberapa fungsi untuk menggenerasi materi kriptografis (sebuah *digest* atau *message* terenkripsi).. Teknik ini digunakan untuk mengubah data ke dalam kode- kode tertentu, dengan tujuan informasi yang disimpan atau ditransmisikan melalui jaringan yang tidak aman tidak dapat dibaca oleh siapa pun kecuali orang-orang yang berhak[2].

Watermarking adalah sebuah proses untuk menyisipkan suatu informasi, yang biasanya disebut sebagai watermark, pada suatu data (*digital*) penampung, seperti gambar, audio, dokumen text, video dan bentuk produk *digital* lainnya. *Audiowatermarking* merupakan salah satu jenis dari *digital watermarking*, adalah suatu proses penyisipan pesan yang berisikan informasi dari berkas *audio* seperti nama pencipta, tanggal pembuatan, tujuan, atau informasi lainnya tanpa mempengaruhi kualitas audio tersebut. Dari hasil evaluasi performansi metode *Phase Coding* memiliki performasi dapat digunakan sebagai teknik audio *watermarking*[3].

Steganografi *"steganography"* adalah ilmu, teknik atau seni menyembunyikan pesan rahasia *"hiding message"* atau tulisan rahasia *"covered writing"* sehingga keberadaan pesan tidak terdeteksi orang lain kecuali pengirim dan penerima pesan tersebut. Steganografi berasal dari bahasa Yunani yaitu *steganos* "tersembunyi/menyembunyikan" dan *graphy* "tulisan", sehingga secara lengkap bermakna tulisan yang disembunyikan. Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama yang berisi informasi sensitif yang hanya boleh diketahui isinya oleh pihak yang berhak saja, apalagi jika pengirimannya dilakukan melalui jaringan publik (*Internet*), apabila data tersebut tidak diamankan terlebih dahulu, maka akan sangat mudah disadap dan diketahui isinya oleh pihak- pihak yang tidak berhak. Salah satu cara yang digunakan untuk pengamanan data adalah menggunakan sistem Stenografi. Kelebihan steganografi adalah untuk menyembunyikan data ke dalam data-data lainnya sehingga tidak memungkinkan pihak ketiga untuk mendeteksi keberadaan pesan[4]. Salah satu metode stenografi untuk mengamankan data adalah metode *spread spectrum*.

RC4A merupakan sebuah upaya untuk meningkatkan keamanan dari RC4 tanpa mengurangi efisiensi. Menurut penelitian terdahulu 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[5].
Pesan atau file teks yang akan dikirimkan terlebih dahulu dienkripsi dengan menggunakan algoritma RC4A. Selanjutnya untuk mendapatkan kembali pesan asli dengan cara dekripsi RC4A maka terlebih dahulu dilakukan proses verifikasi file untuk menjamin bahwa file yang diterima belum mengalami perubahan atau masih asli. Di sisi lain, timbul permasalahan akan kemanan data yang dapat terganggu oleh pihak-pihak yang tidak bertanggung jawab seperti adanya penyadapan, perusakan, pencurian data, ataupun tindakan penyalahgunaan lainnya, maka untuk mengamankan data tersebut di perlukan lah algoritma RC4A.
Metode *spread spectrum* adalah sebuah teknik transmisi dimana kode *pseudo noise* independen dari data informasi yang digunakan sebagai gelombang modulasi untuk menyebarkan energi sinyal melalui sebuah *bandwith* jauh lebih besar dari pada *bandwith* sinyal informasi.Sistem *spread spectrum*memiliki ciri yang khas, yaitu sinyal yang ditransmisikan memiliki lebar pita yang jauh lebih besar dibandingkan dengan lebar pita sinyal informasi yang ditransmisikan. Penyebaran *spectrum* yang terjadi dilakukan oleh fungsi penyebar tersendiri, yang tidak tergantung pada informasi yang disampaikan[6]. Berdasarkan penelitian terdahulu mengatakan bahwa Metode *spread spectrum* mentransmisikan sebuah sinyal pita informasi yang sempit ke dalam sebuah kanal pita lebar dengan penyebaran frekuensi,penyebaran frekuensi sendiri berfungsi menambah tingkat redundansi [7].
Penelitian ini menguraikan bagaimana menkombinasikan algoritma RC4A dengan *spread spectrum method*. Pesan teks yang bersifat rahasia disandikan bedasarkan algoritma RC4A, kemudian di sembunyikan ke dalamcitra digital berdasarkan *spread spectrum method*, sehingga keamanan data lebih optimal.

2. METODOLOGI PENELITIAN

2.1 Kerangka Kerja Penelitian

Pada metodologi penelitian dijabarkan tahapan-tahapan yang dilakukan dalam penelitian. Metodologi penelitian terdiri dari beberapa tahapan yang terkait secara sistematis. Tahapan ini diperlukan untuk mempermudah dalam melakukan penelitian. Sebelum membuat kerangka penelitian, penulis terlebih dahulu menganalisa topik yang akan diteliti. Adapun metodologi Pelaksanaan penelitian ini adalah :

a. Studi pustaka (Library study)
   Bab ini merupakan tahapan yang dilakukan untuk mencari dan mempelajari teori-teori yang berhubungan dengan topik penelitian ini yang dapat bersumber dari buku, jurnal dan referensi lainnya yang relevan.
b. Tahap analisa dan perancangan
   Tahap ini,merupakan kegiatan penganalisaan permasalahan keamanan data serta analisis terhadap pengkombinasian algoritma RC4A dan metode spread spectrum sebagai suatu cara dalam pengoptimalan proses pengaman pesan yang bersifat rahasia dengan membangun perancangan system yang akan digunakan agar pesan rahasia dapat aman.
c. Tahap imlementasi dan pengujian
   Tahap ini, merupakan tahapan penerapan pengkombinasian algoritma RC4A dan metode spread spectrum untuk mengoptimalkan keamanan pesan rahasia serta melakukan pengujian terhadap setiap tahapan-tahapan implementasi yang dilakukan .
d. Dokumentasi
   Tahap ini, merupakan tahapan untuk mendokumentasikan seluruh kegiatan pada saat melakukan penelitian, mulai dari awal penelitian hingga akhir penelitian, kemudian mendokumentasikannya dalam bentuk laporan penelitian.

2.2 Kriptografi

Kriptografi (cryptography) berasal dari Bahasa Yunani, yaitu cryptos dan graphia yang berarti ‘penulisan rahasia’. Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan yang dikirim oleh pengirim dapat disampaikan kepada penerima dengan aman.Kriptografi dalam proses mengambil pesan/message dan menggunakan beberapa fungsi untuk menggenerasi materi kriptografis (sebuah digest atau message terenkripsi).Di dalam menjaga pesan, maka pesan tersebut dapat diubah menjadi satu kode yang tidak dapat dimengerti oleh pihak lain yang disebut dengan proses enkripsi dan dekripsi.Hasilnya berupa waktu proses enkripsi dan dekripsi serta perubahan ukuran gambar setelah dienkripsi. Lamanya waktu proses enkripsi dan dekripsi sangat bergantung pada banyaknya jumlah karakter, lamanya waktu proses enkripsi dan dekripsi berbanding lurus dengan banyaknya jumlah karakter yang digunakan. Perubahan ukuran file setelah disisipkan pesan, mengalami perubahan walaupun tidak significant. Terlihat perubahan file hanya 1 KB tapi sesungguhnya jika dilihat dalam ukuran bytes gambar, gambar yang sudah dienkripsi mengalami perubahan sekitar 200 bytes. Prinsip kerja teknik kriptografi adalah menyandikan *plaintext* menjadi *ciphertext* yang disebut dengan enkripsi *(encryption)* atau *enciphering* sedangkan proses mengembalikan *cipherteks* menjadi *plainteksnya* disebut dengan dekripsi *(decryption)* atau *deciphering*[11].

2.3 Algoritma RC4A

Algoritma RC4A adalah salah satu algoritma yang dapat digunakan untuk melakukan enkripsi data sehingga data asli hanya dapat dibaca oleh seseorang yang memiliki kunci enkripsi tersebut. *Plaintext* akan dienkripsi pertama sekali dengan

algoritma MDTM *Cipher* kemudian hasil enkripsi tersebut dienkripsi lagi dengan algoritma RC4A dan menghasilkan *ciphertext* [21]. Adapun tahap-tahap utama dalam algoritma RC4A adalah proses pembangkitan kunci, proses *enkripsi* dan proses *dekripsi*

a. Proses Pembangkitan Kunci
   Proses pembentukan kunci pada algoritma RC4A terdiri dari dua proses, yaitu proses *Key SchedullingAlgorithm* dan *Pseudo Random Generation AlgorithmKey Schedulling Algorithm* (KSA). Struktur dari algoritma *RC4A* sama seperti struktur algoritma *RC4+* Kedua algoritma tersebut memiliki *Key Scheduling Algorithm* (KSA). Adapun *pseudocode*KSA dari algoritma *RC4A* sebagai berikut :

```
For i from 0 to 255
        S[i] := i
endfor
j := 0
for i from 0 to 255
        j := (j + S[i] + key[i mod keylength])
        mod 256
        swap values of S[i] and S[j]
endfor
```

   nilai i dan j pada *Key Scheduling Algorithm* (KSA) adalah variabel awal yang bernilai 0 dan S adalah 256 permutasi yang bisa mungkin terjadi.

   1. *Pseudo Random Generation Algorithm*
      Pseudo Random Generation Algorithm (PRGA) algoritma *RC4A*, menggunakan dua *state array*, yaitu S1 dan S2, dan tiga buah *indeks* i, j1, dan j2. RC4A menggunakan KSA yang sama dengan *RC4* kecuali satu hal dimana KSA digunakan dua kali, masing-masing sekali untuk S1 dan S2. Semua operasi aritmetika dihitung dengan modulo 256. Proses PRGA dilakukan sebanyak jumlah karakter *plaintext.Pseudo Random Generation Algorithm* (PRGA) algoritma *RC4A* dapat dituliskansebagai berikut :

```
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]]
```

b. Proses Enkripsi
   Langkah-langkah untuk melakukan enkripsi [22], Yaitu melakukan operasi XOR antara *Keystream* yang dihasilkan dari proses PRGA dengan karakter-karakter *plaintext*, kemudian hasil operasi XOR diubah menjadi bilangan hexadesimal untuk menghasilkan *ciphertext*.
c. Proses Dekripsi
   Proses *dekripsi* merupakan proses kebalikan dari proses *enkripsi* yang bertujuan untuk membalikkan data kembali menjadi informasi semula (*plainteks*) yang dapat digunakan oleh pengguna. Proses dekripsi dilakukan dengan [22]:
   1. Bangkitkan Kunci dengan cara yang sama seperti proses pembangkitan *keystream* pada proses enkripsi
   2. Melakukan operasi XOR antara *Keystream* yang dihasilkan dari proses PRGA dengan karakter-karakter *ciphertext*,kemudian hasil operasi XOR diubah menjadi bilangan karakter untuk menghasilkan *ciphertext*.

2.4 Metode *Spread Spectrum*

Spread spectrum adalah sebuah metode komunikasi dimana semua sinyal komunikasi disebar di seluruh spektrum frekuensi yang tersedia. Istilah *spread spectrum* digunakan karena pada sistem ini sinyal yang ditransmisikan memiliki *bandwidth* yang jauh lebih lebar dari *bandwidth* sinyal informasi (mencapai ribuan kali).*Spread Spectrum* merupakan salah satu metode yang dapat digunakan dalam teknik steganografi pada media berkas *audio digital* dalam *domain transform*[26]. Sebuah sistem *spread-spectrum* harus memenuhi kriteria[27], sebagai berikut :

a. Sinyal yang dikirimkan menduduki bandwidth yang jauh lebih lebar daripada *bandwidth* minimum yang diperlukan untuk mengirimkan sinyal informasi.
b. Pada pengirim terjadi proses *spreading* yang menebarkan sinyal informasi dengan bantuan sinyal kode yang bersifat independen terhadap informasi.
c. Pada penerima terjadi proses *despreading* yang melibatkan korelasi antara sinyal yang diterima dan replika sinyal kode yang dibangkitkan sendiri oleh suatu generator lokal.

Proses*embedding* dalam teknik*spread spectrum* mempunyai beberapa tahapan proses[31], yaitu :

a. Siapkan citra *cover* yang menjadi media penyembunyi pesan rahasia.

b. Lakukan pembangkitan kunci stegano menggunakan proses pembangkit bilangan acak semu berdasarkan algoritma *Linear Congruen Method (*LCG).
c. Melakukan proses *spreading* (penyebaran biner-biner pesan yang akan disembuntikan pada citra *cover*). Proses *spreading* dilakukan sesuai dengan menentukan sebuah bilangan pengali skalar. Biner-biner pesan yang akan disisipkan akan disebar sesuai bilangan pengali skalar yang telah ditentukan, maka hasil keluaran dari proses *spreading* ini adalah deret bilangan biner yang telah tersebar dengan panjang setiap deretnya sebesar 32 bit
d. Melakukan proses modulasi pesan
   Proses ini merupakan proses yang dilakukan untuk mengacak hasil spreading dengan bilangan *pseudonoise* yang dibangkitkan berdasarkan LCG. Jumlah dari bilangan *pseudonoise* disesuaikan dengan jumlah biner pesan pesan. Bila jumlah pesan lebih kecil dari jumlah bilangan *pseudonoise*, bilangan *pseudonoise* tersebut akan dipotong sesuai dengan ukuran pesan. Sebaliknya, bila jumlah biner pesan lebih besar daripanjang bilangan *pseudonoise*, maka bilangan tersebut akan diulang sampai panjangnya sama dengan jumlah biner pesan. Proses modulasi tersebut dilakukan dengan menggunakan fungsi XOR (Exclusive OR). Biner-biner hasil modulasi adalah biner-biner yang disisipkan ke dalam *cover image*.
e. Proses penyembunyian biner-biner hasil modulasi dilakukan dengan mengganti bit LSB dari masing-masing citra *cover*.
f. Citra *cover* yang telah disipkan pesan akan dipetakan kembali menjadi citra baru yang disebut dengan *stegano image*.

3. HASIL DAN PEMBAHASAN

Peningkatan pengamanan terhadap data rahasia sangat perlu dilakukan agar keaslian data tetap terjaga. Keamanan yang dilakukan berdasarkan teknik kriptografi dapat dioptimalkan dengan menyembunyikan pesan yang telah diamankan berdasarkan sebuah algoritma kriptografi ke dalam sebuah media penyembunyi berdasarkan salah satu metode dari teknik steganografi. Algoritma RC4A akan digunakan untuk melakikan proses penyandian terhadap pesan yang akan disembunyikan, sedangkan meode yang digunakan untuk menyembunyikan pesan terenkripsi adalah metode *Spread Spectrum Method*.

3.1 Implementasi Metode

Berikut ini akan menguraikan proses penerapan algoritma RC4A dan metode *Spread Spectrum* dalam mengamankan data. Diasumsikan data yang diamankan adalah potongan dari teks kata sandi akun email, yang selanjutnya disebut *plaintext*. Citra *cover* yang digunakan sebagai media penyembunyi pesan adalah citra warna berextensi bmp dengan resolusi 7 x 7 (49 *pixel*citrawarna).

Plaintext = hen_97
Kunci Awal = lasese
CoverImage adalah Citra Warna

Gambar 1. Citra *Cover* ukuran 532 x 689 *pixel extensi bmp*

a. Proses Enkripsi Pesan
   Langkah awal yang dilakukan adalah, merubah karakter *plaintext* dan kunci menjadi biner.

Tabel 1. Biner Kunci

Karakter	Decimal
l	108
a	97
s	115
e	101
s	115
e	101

Tabel 2. Biner *Plaintext*

Karakter	Decimal
h	104

Journal Global Tecnology Computer
Vol 1, No. 1, Desember 2021, Hal 1 - 12
ISSN: **2809-6118** (Online - Elektronik)
Website https://ejurnal.seminar-id.com/index.php/jogtc

e	101
n	110
_	95
9	57
7	55

b. Proses Pembangkitan Kunci

Proses pembangkitan kunci dilakukan berdasarkan algoritma RC4A. Kunci awal akan digunakan sebagai input untuk menghasilkan kunci acak melalui proses KSA dan PRGA.

1. *Key Schedulle* Algorithm (KSA)

Proses KSA membutuhkan tabel array S dengan nilai 0 hingga 255 atau 256 array.

Tabel 3. Tabel Array S

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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
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
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
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
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
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
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
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
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
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
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
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
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
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
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
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Nilai-nilai pada tabel array S akan diacak untuk mendapatkan input yang dibutuhkan pada proses pembangkitan kunci melalui proses PRGA. Proses pengacakan nilai-nilai pada tabel array S dilakukan sebanyak 255 iterasi yang dimulai dari iterasi ke-0 hingga iterasi ke-255 serta membutuhkan nilai-nilai array kunci awal, sehingga kata kunci awal akan dibuat dalam bentuk array.

Tabel 4. Tabel Kunci Awal

Karakter Kunci	l	a	s	e	s	e
Desimal	108	97	115	101	115	101
Index	0	1	2	3	4	5

Proses pengacakan nilai-nilai pada tabel array S:
Iterasi ke-i = 0; j = 0
Panjang kunci = 6 karakter
j= (j + S[i] + key[I mod keylength] mod 256
i = 0 j=0
j= (0+0+ key [0 mod 6] mod 256
j = (0 + 0+ key[0] mod 256

j = (0+0 +108) mod 256
j = 108 mod 256 =108
Maka pada proses ini didapatkan nilai i = 0 dan nilai j adalah 108
Swap S[i], S[j], maka :
S[0] = 0 dan S[108]=108, dua nilai pada posisi ini saling ditukarkan sehingga :
Nilai array tabel S[0] = 108 dan nilai array tabel S[108] = 0

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
199	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
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
96	97	98	99	100	101	102	103	104	105	106	107	199	109	110	111
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Iterasi ke-i = 1
j = (j + S[i] +key [i mod keylength) mod 256
j =1 j=108
j = (108+ S [i]+key[0 mod 6] mod 256
j = (108+1+key[1] mod 256
j = (108+1+97) mod 256
j = 206 mod 256 =206
maka pada proses ini didapatkan nilai i = 1 dan nilai j adalah 97
Swap S[i], S[j], maka :
S[0] = 0 dan S[108]=108, dua nilai pada posisi ini saling ditukarkan sehingga :
Nilai array tabel S[0] = 108 dan nilai array tabel S[108] = 0
Iterasi ke-i = 2
j = (j + S[i] +key [i mod keylength) mod 256
j =2 j=206
j = (206+ S [i]+key[2 mod 6] mod 256
j = (206+2+key[2] mod 256
j = (206+2+115) mod 256
j = 323 mod 256 = 67

Proses pengacakan untuk iterasi selanjutnya dilakukan dengan cara yang sama seperti di atas, sehingga dihasilkan hasil keseluruhan dari proses KSA adalah :

Tabel 5. Hasil Proses KSA Array S

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
108	206	67	171	34	140	40	47	55	64	74	86	100	116	134	154
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
176	200	226	254	18	40	64	90	118	145	177	211	247	29	59	91
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
125	161	199	239	25	69	115	163	203	245	33	79	127	177	229	27
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
83	141	191	243	41	97	155	215	21	85	151	219	23	85	149	215
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
27	97	169	243	63	141	211	27	101	177	255	79	161	245	75	163
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
243	69	153	239	71	161	253	91	187	29	119	211	49	145	243	87

96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
189	37	143	251	95	197	45	151	3	113	225	83	199	61	171	27
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
141	1	119	239	105	229	99	227	91	213	81	207	79	209	85	219
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
99	237	111	243	121	1	139	23	165	53	199	91	231	117	5	151
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
43	193	89	243	143	45	195	91	245	145	47	207	101	91	175	87
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
247	153	61	227	139	53	225	143	63	241	155	71	245	165	87	11
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
193	121	51	239	163	89	17	203	135	69	5	199	139	81	91	207
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
145	85	27	227	173	121	71	23	203	169	119	69	21	231	187	145
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
105	67	21	233	191	151	113	77	43	11	237	209	173	139	107	77
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
49	23	255	233	213	195	169	145	123	103	85	69	55	43	33	25
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
9	251	239	229	221	215	211	209	209	211	205	201	199	199	201	205

2. *Pseudo Random Generation* Algorithm (PRGA)
   Proses PRGA dilakukan untuk mendapatkan kunci sebanyak jumlah karakter *plaintext*. Proses ini membutuhkan dua tabel KSA yang sama yang diambil dari hasil tabel KSA di atas. Nilai-nilai dari tabel KSA akan diacak dengan jumlah iterasi sama dengan jumlah karakter *plaintext* untuk mendapatkan kunci yang digunakan dalam proses enkripsi maupun dekripsi.

Tabel 6. Tabel KSA 1 dan KSA 2

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

iterasi i = 0

i = i +1 = 0+1
i= j1 =j2 0
i= +1
i= 0+1
i= 0
j1= (0+S1[i]) mod 256
j1= (0+S[108]) mod 256
j1= 108
swap S1[i], [S1[j1]
swap S1[0], S1 [108]

Tukarkan nilai indeks 0 dengan nilai indeks 108 pada tabel S1 berarti:
indeks 0 = 199
indeks 108 = 0
t1= (S1[i] + [j1]) mod 256
t1 = (S1[0] + S1[117] mod
t1 = (108 + 199) mod 256)
t1= 51
output S2[t1] = 243
berarti nilai t1 diambil dari tabel S2 indek ke 51
t1 = 243 j2= (j2 + S2[i]) mod 256
j2= 0 + S2[1]
j2= 0 + 108 = 108
swap S2[i], S2[j2]
swap S2[1], S2 [108]

Tukar nilai indeks 1 dengan nilai indeks 108 pada tabel S2
Pada tabel S2 : index 1 = 108
Pada tabel S2 : indeks 108 = 199
Berarti: indeks 1 menjadi nilai pada indeks 108= 199
indeks 199 menjadi nilai pada indeks 1 = 108

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
199	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
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
				.....	.....	.....	.....	.....	.....						
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
96	97	98	99	100	101	102	103	104	105	106	107	199	109	110	111
				.....	.....	.....	.....	.....	.....						
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

Lakukan hal yang sama sampai dengan **iterasi i = 5**
Berdasarkan hasil iterasi di atas, maka diperoleh kunci akhir yang digunakan dalam proses enkripsi dan dekripsi adalah :

Tabel 7. Kunci Enkripsi dan Dekripsi

Key Sream	Char	J	□	Ó	□	□	ç
	Dec	74	27	243	23	23	231

b. Proses Enkripsi
Proses enkripsi dilakukan dengan meng-XOR-kan masing-masing karakter *plaintext* dengan karakter kunci sesuai dengan indexnya. Oleh karena itu masing-masing karakter *plaintext* dan kunci yang sudah didapatkan dari proses PRGA harus dikonversi menjadi biner.

Tabel 8. Nilai Biner Kunci

Index	Char	Desimal	Biner
K[0]	J	74	01001010
K[1]	□	27	00011011
K[2]	Ó	243	11110011
K[3]	□	23	00010111
K[4]	□	23	00010111
K[5]	Ç	231	11100111

Tabel 9. Biner Karakter *Plaintext*

Karakter	Karakter	Decimal	Biner
P0	h	104	01101000
P1	e	101	01100101
P2	n	110	01101110
P3	_	95	01011111
P4	9	57	00111001
P5	7	55	00110111

C0 = P0 XOR K0
P0 = 01101000
K0 = 01001010XOR
C0 = 00100010
Lakukan hal sama sampai langkah ke 5

Tabel 10. Konversi Biner *Ciphertext*

Cipher	Biner	Decimal	Karakter *Cipher*
C0	00100010	34	"
C1	01111110	126	~
C2	10011101	157	□
C3	01001000	72	H
C4	00101110	46	.
C5	11010000	208	Đ

Berdasarkan proses *enkripsi* di atas, maka diperoleh *ciphertext* adalah "~□H. Đ

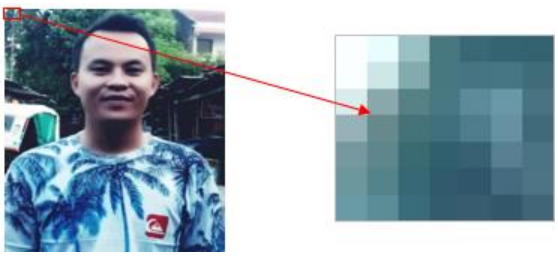
c. Proses *Embedding Ciphertext*

Sebelum proses *embedding* dilakukan, maka terlebih dahulu dilakukan proses s*pread* terhadap bit-bit *ciphertext* yang akan disembunyikan ke dalam citra sebagai media penyembunyi. Kunci stegano (*stegano key*) pada kasus ini disembunyikan dengan mengganti nilai *pixel* yang pertama pada media penyembunyi, sehinngga yang perlu disebar (*spread*) adalah binerbit-bit *ciphertext*. Masing-masing karakter kunci stegano di XOR agar nilai kunci yang didapatkan hanya terdiri dari satu nilai.

Kunci Stegano = HENDRIK

Kata kunci stegano inilah yang didistribusikan kepada penerima pesan, namun yang disimpan di dalam citra adalah hasil proses operasi XOR masing-masing karakter kunci stegano.

Setelah biner *ciphertext* dimodulasi, maka didapatkan jumlah bit akhir dari *ciphertext* adalah 144 bit. Biner inilah yang disembunyikan ke dalam citra sebagai media penyembunyi *ciphertext* (citra *cover*). Citra *cover* yang digunakan adalah citra warna dengan format *bmp* dengan resolusi 532 x 689*pixel*, namun agar mempercepat proses pengerjaan contoh kasus, maka diambil sebagian *pixel* dari citra *cover* asli sesuai dengan jumlah yang dibutuhkan untuk menyembunyikan *ciphertext* dan kunci *stegano*. Berdasarkan jumlah biner *ciphertext* yang telah dimodulasi dibutuhkan minimal 49 *pixel* citra warna, ditambah 1 *pixel* untuk menampung kunci steganografi menjadi 49 *pixel*. Resolusi citra *cover* yang digunakan pada contoh kasus ini adalah citra dengan resolusi 7 x 7 *pixel*.

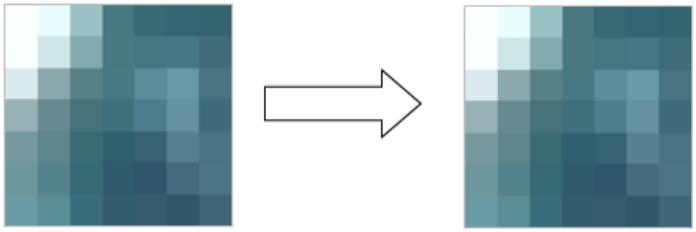


Gambar 2. Citra *Cover* ukuran 7 x 7 *pixel*

Nilai-nilai warna citra cover di atas yang berukuran 7 x 7 *pixel* diambil menggunakan aplikasi matlab, dengan *code* :

Clc; clear; close all;
I = *imread* ('hendrik.*jpg*');
Red = I(:,:,1);
Green = I(:,:,2);
Blue = I(:,:,3);

Setelah dilakukan proses penggantian bit LSB citra *cover* dengan bit *ciphertex*t, maka nilai-nilai *pixel* akan dipetakan kembali menjadi citra baru yang selanjutnya disebut dengan citra *stegano*.



(a) Citra Cover (b) Citra Stegano)

Gambar 3. Hasil Proses *Embedding* (Citra Stegano)

Berdasarkan citra *stegano* yang dihasilkan pada proses *embedding* di atas, maka terlihat citra hasil *stegano* masih terlihat sama seperti citra *cover*, karena perubahan yang terjadi tidak signifikan. Nilai *pixel* akan naik satu bit bila nilai LSB *pixel* citra *cover* adalah 0 diganti dengan bit 1 dari bit pesan, sebaliknya akan turun 1 bit bila nilai LSB dari *pixel* citra *cover* adalah 1 dan bit pesan adalah 0, serta akan bernilai tetap bila LSB dari *pixel* citra *cover* sama dengan bit pesan yang akan disisipkan.

d. Proses *Decoding Ciphertext*

Proses *decoding ciphertext* atau proses pengambilan pesan tersebunyi dari dalam citra stegano dilakukan dengan mengambil kembali nilai-nilai LSB dari citra stegano, kemudian biner-biner tersebut didekripsi berdasarkan algoritma RC4A agar diketahui pesan asli. Sebelum proses *decoding* dilakukan, terlebih dahulu diuji kebenaran atau kesamaan kunci stegano yang *diinput* oleh pengguna dengan kunci stegano yang telah tersimpan di dalam nilai *pixel* 0 citra *stegano*. Kunci *stegano* awal yang *diinput* oleh penerima pesan adalah kata kunci yang diterima dari pengirim yaitu HENDRIK. Masing-masing karakter dari kata kunci ini kemudian di XOR seperti proses yang dilakukan pada tahap *encoding*, sehingga dihasilkan biner 01010111 dengan desimal 87. Langkah selanjutnya adalah merubah warna citra stegano menjadi nilai biner, agar nilai-nilai LSB setiap *pixelnya* dapat digunakan. Khusus untuk *pixel* 0, maka akan digunakan sebagai nilai parameter untuk mencocokan nilai kunci stegano. Bila nilai desimal atau biner *pixel* pertama (baik nilai *red, green* dan *blue)* sama dengan nilai desimal atau biner hasil proses XOR kunci stegano, maka proses *ekstraksi* dilanjutkan. Namun bila tidak sama, maka proses *ekstraksi* tidak dapat dilanjutnya.

e. Proses Dekripsi *Ciphertext*

Input proses dekripsi adalah biner *ciphertext* yang dihasilkan dari proses *de-spreading* (tabel 3.11). Proses ini bertujuan untuk mengetahui makna asli dari *ciphertext* yang diperoleh dari proses *ekstraksi*. Proses dekripsi dilakukan berdasarkan algoritma RC4A sesuai dengan algoritma yang digunakan ada saat melakukan proses enkripsi. Formulai dekripsi berdasarkan algoritma RC4A dilakukan dengan meng-XOR-kan masing-masing biner karakter *ciphertext* dengan biner karakter kunci. Proses pembangkitan kunci pada saat melakukan dekripsi dilakukan dengan cara yang sama seperti pada proses enkripsi yang meliputi ptoses KSA dan proses PRGA, sehingga nilai kunci yang dihasilkan dan digunakan sama seperti nilai-nilai kunci yang digunakan pada proses enkripsi.

Proses dekripsi disajikan pada proses di bawah ini:

Tabel 11. Proses *Dekripsi Ciphertext*

	Cipher					
Index	*Biner*	Operasi	Kunci	*Plaintext*	Desimal	Karakter
Biner C0	00100010	XOR	01001010	01101000	104	h
Biner C1	01111110	XOR	00011011	01100101	101	e
Biner C2	10011101	XOR	11110011	01101110	110	n
Biner C3	01001000	XOR	00010111	01011111	95	-
Biner C4	00101110	XOR	00010111	00111001	57	9
Biner C5	11010000	XOR	11100111	00110111	55	7

Berdasarkan tabel di atas, maka diperoleh karakter *plaintext* adalah hen_97

4. KESIMPULAN

Berdasarkan pembahasan dan hasil implementasi penelitian, maka di peroleh kesimpulan yaitu Penyembunyian pesan rahasia yang telah terenkripsi berdasarkan metode steganografi dapat meningkatkan keamanan pesan rahasia, karena

selain pesan asli yang telah dienkripsi, pesan terenkripsi tersebut disembunyikan ke dalam media lain seperti citra digital, sehingga keamanan pesan lebih terjamin. Berdasarkan hasil dari algorita *Speread Spectrum* dan RC4A dalam mengamankan data,nilai MSE dan PSNR yang dihasilkan dapat disimpulkan bahwa kualitas gambar yang dihasilkan setelah dilakukan penyisipan *ciphertext* cukup baik, karena nilai PSNR yang dihasilkan lebih dari 30 dB. Aplikasi pengamanan pesan rahasia berdasarkan pengkombinasian algoritma *Speread Spectrum* dengan metode RC4A dapat membantu untuk mempermudah pengguna dalam mengamankan pesan penting yang bersifat rahasia.

REFERENCES

[1] R. Munir, "Kriptografi," in *2*, 2019.
[2] S. Kromodimoeljo, *Teori dan Aplikasi Kriptograf*. 2009.
[3] A. R. S and A. Ito, "Implementasi Digital Audio Watermarking pada Berkas Suara dengan Menggunakan Metode Least Significant Bit," *J. Tek. Pomits*, 2014.
[4] M. Marsofiyati, "ANALISIS RELEVANSI STENOGRAFI SEBAGAI MATA KULIAH PROGRAM STUDI D3 SEKRETARI DAN PENDIDIKAN ADMINISTRASI PERKANTORAN," *J. Pendidik. Ekon. dan Bisnis*, vol. 5, no. 1, p. 23, 2017, doi: 10.21009/jpeb.005.1.2.
[5] N. Hayati, "Implementasi Algoritma RC4A dan MD5 untuk Menjamin Confidentiality dan Integrity pada File Teks," *J. Penelit. Tek. Inform.*, vol. 1, no. April, pp. 51–57, 2017.
[6] R. Fauzi, "Spread Socetrum," pp. 1–13.
[7] C. A. Wael, "ANALISA PERFORMANSI SPREAD SPECTRUM IMAGE STEGANOGRAPHY (SSIS) PADA KANAL MULTIPATH RAYLEIGH FADING," *J. Dimens.*, 2016, doi: 10.33373/dms.v3i2.88.
[8] Paryati, "Keamanan Sistem Informasi," *Semin. Nas. Inform. 2008 (semnasIF 2008) UPN "Veteran" Yogyakarta, 24 Mei 2008*, 2008.
[9] N. M. Dahlan M., Latubessy A., "Analisa Keamanan Web Server Terhadap Serangan Possibility Sql Injection," *Pros. SNATIF*, 2015.
[10] N. B. Nugroho, "Aplikasi Keamanan Email Menggunakan Algoritma Rc4," *J. SAINTIKOM*, 2016.
[11] D. Wirdasari, "Prinsip Kerja Kriptografi dalam Mengamankan Informasi," *Saintikom*, 2008.
[12] S. Winiarti, "JURNAL INFORMATIKA Vol 2, No. 2, Juli 2008," *Pemanfaat. Teorema Bayes Dalam Penentuan Penyakit THT*, 2008.
[13] A. Wanto, "Analisis Mengatasi Sniffing Dan Spoofing Menggunakan Metode Enkripsi Dan Dekripsi Algoritma Hill Chiper," *Semin. Nas. Ilmu Komput. 2016*, 2016.
[14] R. Arifin and L. T. Oktoviana, "Implementasi Kriptografi Dan Steganografi Menggunakan Algoritma RSA Dan Metode LSB," *J. Din. Inform.*, 2013.
[15] R. Munir, "Algoritma Enkripsi Citra Digital Dengan Kombinasi Dua Chaos," *Chaos*, 2012.
[16] M. H. L. Louk, "Sistem Kriptografi di Komputasi Awan Untuk Kebutuhan Data Medis," *Teknika*, 2018, doi: 10.34148/teknika.v7i1.92.
[17] H. Aditya, I. N. Farida, and R. A. Ramadhani, "Heru Aditya Penerapan Algoritma Elgamal dan SSL Pada Aplikasi Group Chat," *Gener. J.*, 2018, doi: 10.29407/gj.v2i1.12052.
[18] M. E. Putra, S. Suroso, and A. Wasti, "PERANCANGAN APLIKASI PENGAMANAN INFORMASI TEKS DENGAN MENGGUNAKAN ALGORITMA KRIPTOGRAFI ALPHA-QWERTY REVERSE," *J. Elektro dan Telekomun. Terap.*, 2017, doi: 10.25124/jett.v4i1.996.
[19] S. Maharani and F. Agus, "Implementasi Perangkat Lunak Penyandian Pesan Menggunakan Algoritma RSA," *J. Inform. Mulawarman*, 2009.
[20] N. M. Latuconsina and P. W. Yunanto, "Pembuatan Bank Soal Dan Analisis Butir Soal Mata Kuliah Kriptografi Untuk Mahasiswa Program Studi Pendidikan Teknik Informatika Dan Komputer Universitas Negeri Jakarta," *PINTER J. Pendidik. Tek. Inform. dan Komput.*, 2017, doi: 10.21009/pinter.1.2.7.
[21] N. B. Nugroho, Z. Azmi, and S. N. Arif, "Aplikasi Keamanan Email Menggunakan Algoritma Rc4," *J. SAINTIKOM*, 2016.
[22] I. Mantin and A. Shamir, "A practical attack on broadcast RC4," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2002, doi: 10.1007/3-540-45473-X_13.
[23] P. Alatas, "Steganografi," *Implementasi Tek. Steganografi Dengan Metod. Lsb Pada Citra Digit.*, 2009.
[24] Y. Prayudi and P. S. Kuncoro, "IMPLEMENTASI STEGANOGRAFI," *SNATI*, 2005.
[25] N. Fuad, S. Suyono, and E. Setyati, "Teknik Stenografi dengan Menggunakan Metode Visual Attacks dan Statistical Attacks," *J. Ilm. Teknol. Inf. Asia*, 2011.
[26] M. Hizlan, "Spread Spectrum," in *Handbook of Computer Networks*, 2011.
[27] L. Nurmalia and M. Pinem, "Analisis Perbandingan Teknologi Spread Spectrum FHSS dan DSSS pada Sistem CDMA," *Sigunda Ensikom*, 2013.
[28] B. Herdiana, M. Aria, and J. Utama, "PEMBANGKITAN DATA ACAK TERSEBAR DIRECT SEQUENCE SPREAD SPECTRUM PADA LAJU DATA BERKECEPATAN RENDAH UNTUK APLIKASI TEKNOLOGI CODE DIVISION MULTIPLE ACCESS," *SINERGI*, 2017, doi: 10.22441/sinergi.2017.3.005.
[29] A. P. Ratnasari and F. A. Dwiyanto, "Metode Steganografi Citra Digital," *Sains, Apl. Komputasi dan Teknol. Inf.*, 2020, doi: 10.30872/jsakti.v2i2.3300.
[30] H. HARAHAP, G. BUDIMAN, and L. NOVAMIZANTI, "Implementasi Teknik Watermarking menggunakan FFT dan Spread Spectrum Watermark pada Data Audio Digital," *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, 2018, doi: 10.26760/elkomika.v4i1.98.
[31] A. N. Yohanes Nugraha, "PENGAMANAN PESAN TEKS MENGGUNAKAN TEKNIK STEGANOGRAFI SPREAD SPECTRUM BERBASIS ANDROID," *Antivirus J. Ilm. Tek. Inform.*, 2016, doi: 10.35457/antivirus.v10i1.85.
[32] T. Baskara, "Studi Dan Implementasi Steganografi Pada File Audio Dengan Teknik Spread Spectrum," *Media*, 2009.
[33] Sinaga ASRM, "Implemetentasi Teknik Thresholding Pada Segmentasi Citra Digital," *Mantik Penusa*, 2017.

[34] M. M. Amin, "IMAGE STEGANOGRAPHY DENGAN METODE LEAST SIGNIFICANT BIT (LSB)," *CSRID (Computer Sci. Res. Its Dev. Journal)*, 2015, doi: 10.22303/csrid.6.1.2014.53-64.
[35] K. Firdausy, I. Hawariyanta, and M. Murinto, "IMPLEMENTASI WATERMARKING UNTUK PENYEMBUNYIAN DATA PADA CITRA DALAM DOMAIN FREKUENSI MENGGUNAKAN DISCRETE COSINE TRANSFORM," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, 2006, doi: 10.12928/telkomnika.v4i1.1240.
[36] W. Gazali, H. Soeparno, and J. Ohliati, "DALAM PENGOLAHAN CITRA DIGITAL," *J. Mat Stat*, 2012.
[37] S. sahrul Asri, "TELAAH BUKU TEKS PEGANGAN GURU DAN SISWA PADA MATA PELAJARAN BAHASA INDONESIA KELAS VII BERBASIS KURIKULUM 2013," *RETORIKA J. Ilmu Bhs.*, 2017, doi: 10.22225/jr.3.1.94.70-82.
[38] A. Amborowati and R. Marco, "Data Manajemen Dan Teknologi Informasi," *J. Ilm. Data Manaj. dan Teknol. Inf.*, 2015.
[39] O. K and J. Devitra, "Analisis dan Perancangan Sistem Informasi Penggajian Karyawan (Studi Kasus : PT. Kosambi Laksana Mandiri)," *J. Manaj. Sist. Inf.*, 2017.
[40] R. S. Kharisma, R. Kurniawan, and A. C. Wijaya, "PERANCANGAN MEDIA PEMBELAJARAN BERHITUNG BERBASIS MULTIMEDIA FLASH Pendahuluan Tinjauan Pustaka Hasil dan Pembahasan," *J. Ilm. DASI*, 2015.
[41] Munawar, *Analisis Perancangan Sistem Berorientasikan Objek dengan UML (Unified Modeling Language)*. 2018.
[42] 2013 Rosa & Salahuddin, "UML, Use Case Diagram, Activity Diagram, Class Diagram," in *Rekayasa Perangkat Lunak Terstruktur*, 2013.
[43] S. L. Mufreni, "RANCANG BANGUN SISTEM ANTREAN MULTI FUNGSI DENGAN MENGGUNAKAN CREDIT CARD-SIZED COMPUTER UNTUK PENDAFTARAN MAHASISWA BARU," *Transmisi*, 2018, doi: 10.14710/transmisi.20.2.79-84.
[44] www.temukanpengertian.com, *Pengertian UML*. 2016.
[45] I. A. Ridlo, "Panduan pembuatan flowchart," *Fak. Kesehat. Masy.*, 2017.
[46] Malabay, "Pemanfaatan Flowchart Untuk Kebutuhan Deskripsi Proses Bisnis," *J. Ilmu Komput.*, 2016.
[47] D. Andika, "Pengertian Flowchart," *It.Jurnal.Com*, 2018.
[48] J. Qian, T. Hastie, J. Friedman, R. Tibshirani, and N. Simon, "Glmnet for Matlab, 2013," *URL http://www. stanford. edu/~ hastie/glmnet_matlab*. 2013.
[49] D. Apriyadi, "Pengertian MATLAB," *Desember*, 2013. .
[50] R. Risnawati and M. Handayani, "penerapan Jaringan Saraf Tiruan Untuk Proyeksi Logistik Berdasarkan Prediksi Pasien Menggunakan Algoritma Backpropagation," *JURTEKSI*, 2017, doi: 10.33330/jurteksi.v4i1.20.
[51] S. Attaway, "String Manipulation," in *Matlab*, 2012.
[52] *Visual Basic 2008 Recipes*. 2008.
[53] R. Priyanto, "Visual Basic Net 2008," *Andi*, 2009.
[54] R. Saputra, "DESAIN SISTEM INFORMASI ORDER PHOTO PADA CREATIVE STUDIO PHOTO DENGAN MENGGUNAKAN BAHASA PEMROGRAMAN VISUAL BASIC . NET 2010," *Momentum*, 2015.