Jurnal CyberTech

Vol.x. No.x, September 201x, pp. xx~xx
P-ISSN : 9800-3456
E-ISSN : 2675-9802

IMPLEMENTASI STEGANOGRAFI DAN KRIPTOGRAFI DATA PASIEN PADA KLINIK PRATAMA SITI RAHMAH MENGGUNAKAN METODE *LEAST SIGNIFICANT BIT* DAN ALGORITMA RC4

Sri Winda Nesti Putri Santi Mendrofa[1], Azannudin[2], Vina Winda Sari[3]

[1,3] Program Studi Sistem Informasi, STMIK Triguna Dharma
[2] Program Studi Sistem Komputer, STMIK Triguna Dharma

Article Info

Article history:
Received Jun 12th, 201x
Revised Aug 20th, 201x
Accepted Aug 26th, 201x

Keyword:
Data Pasien
Kriptografi
Steganografi
RC4 (Rivest Code 4)
LSB (Least Significant Bit)

ABSTRACT

Dalam suatu perusahaan/organisasi, keamanan data merupakan suatu hal yang sangat penting. Untuk menjaga informasi ataupun data agar tidak jatuh ke pihak yang tidak berwenang atau pihak yang tidak berkepentingan dituntut adanya pengamanan data/informasi. Ilmu yang berkaitan untuk menjaga keamanan pesan adalah kriptografi, dimana kriptografi sendiri terbagi atas dua jenis yaitu kriptografi simetris dan asimetris. Selain itu untuk menjaga kerahasiaan pesan dapat menggunakan steganografi. Dalam penelitian ini, mengkombinasikan algoritma RC4 untuk proses enkripsi dan dekripsi data/informasi dan metode Least Significant Bit (LSB) untuk penyisipan atau penyembunyian pesan kedalam media citra. Hasil dari aplikasi ini adalah dapat menyisipkan pesan tersembunyi berupa teks yang sudah dienkripsi ke dalam media citra berformat JPEG dan dapat mengekstraksi kembali pesan tersembunyi tersebut dari dalam media citra.

Corresponding Author:
Nama : Sri Winda Nesti Putri Santi Mendrofa
Program Studi : Sistem Informasi
STMIK Triguna Dharma
Email: sw.nesti@gmail.com

1. PENDAHULUAN

Saat ini teknologi informasi sudah sangat berkembang dan menjadi salah satu media yang paling populer di dunia. Akan tetapi, seiring berkembangnya teknologi informasi maka semakin berkembang pula tindak penyalahgunaan informasi tersebut. Ada kalanya informasi bersifat penting dan rahasia sehingga keberadaannya tidak boleh diketahui oleh pihak yang tidak berkepentingan. Keamanan dan integritas data adalah sesuatu yang harus diperhatikan. Upaya untuk menjaga agar informasi tidak terjerumus ke tangan-tangan yang tidak berwenang dituntut perlunya menerapkan mekanisme keamanan yang baik [1] . Salah satu informasi atau data yang perlu dijaga keamanannya adalah data pasien. Data pasien seringkali disalahgunakan oleh pihak yang tidak bertanggungjawab dan digunakan untuk hal-hal yang tentunya bisa menimbulkan masalah, baik dari pihak pasien maupun pihak tempat pasien dirawat.

Kriptografi adalah ilmu dan seni yang mempelajari cara-cara untuk menyembunyikan informasi dengan cara menyamarkan menjadi sandi yang sulit ditemukan maknanya [2]. Kriptografi berasal dari bahasa Yunani yang terdiri dari dua buah kata yaitu *Crypto* dan *Graphia*. Kata *crypto* berarti *secret* (rahasia) sedangkan *graphia* berarti *writing* (tulisan), yang berarti secara umum makna dari kata kriptografi adalah tulisan rahasia. Di dalam ilmu kriptografi, pesan disembunyikan dengan cara diacak melalui proses enkripsi. Namun, pesan yang teracak dapat menimbulkan kecurigaan. Untuk menghindari kecurigaan tersebut, maka metode kriptografi dapat di kombinasikan dengan ilmu Steganografi. Kriptografi secara umum adalah ilmu dan seni untuk menjaga kerahasiaan pesan, sedangkan steganografi berasal dari bahasa Yunani yaitu *steganos* yang artinya tersembunyi dan *Graphein* yang artinya menulis, sehingga steganografi adalah seni dan juga ilmu menulis atau menyembunyikan pesan dengan menggunakan cara tertentu sehingga selain pengirim dan penerima, tidak ada akan orang lain yang mengetahui dan menyadari bahwa adanya suatu pesan rahasia yang disembunyikan [3]. Dalam kriptografi dan steganografi data yang dianggap rahasia akan disamarkan dengan sedemikian rupa sehingga jika data itu bisa didapatkan maka tidak akan bisa dimengerti oleh pihak yang tidak berhak. Adapun salah satu metode yang dapat digunakan pada teknik kombinasi kriptografi dan steganografi *yaitu Rivest Code 4 (RC4)* dan *Least Significant Bit (LSB)* [4].

Rivest Code 4 (RC4) merupakan salah satu algoritma kunci simetris yang berbentuk *stream cipher*, yaitu memproses *unit* atau *input* data pada suatu saat. Kelebihan dari algoritma RC4 adalah proses penyandian pesan teks yang melakukan enkripsi per bit sehingga kerusakan pada satu bit tidak mempengaruhi keseluruhan isi pesan. Sedangkan, metode *Least Significant Bit (LSB)* merupakan salah satu metode dalam steganografi yang mengambil bit-bit terakhir warna pada citra dan menggantinya dengan bit-bit data. Kelebihan dari metode *Least Significant Bit (LSB)* adalah mudah diimplementasikan serta perubahan bit pada *Cover-Object* tidak terlalu signifikan sehingga secara kasat mata tidak menimbulkan kecurigaan [5].

2. METODE PENELITIAN

Metode penelitian merupakan langkah-langkah yang di lakukan untuk mengumpulkan data atau in*form*asi yang dibutuhkan oleh seorang pengembang perangkat lunak (*Software*) sebagai tahapan serta gambaran penelitian yang akan dibuat. Berikut adalah metode dalam penelitian ini yaitu :

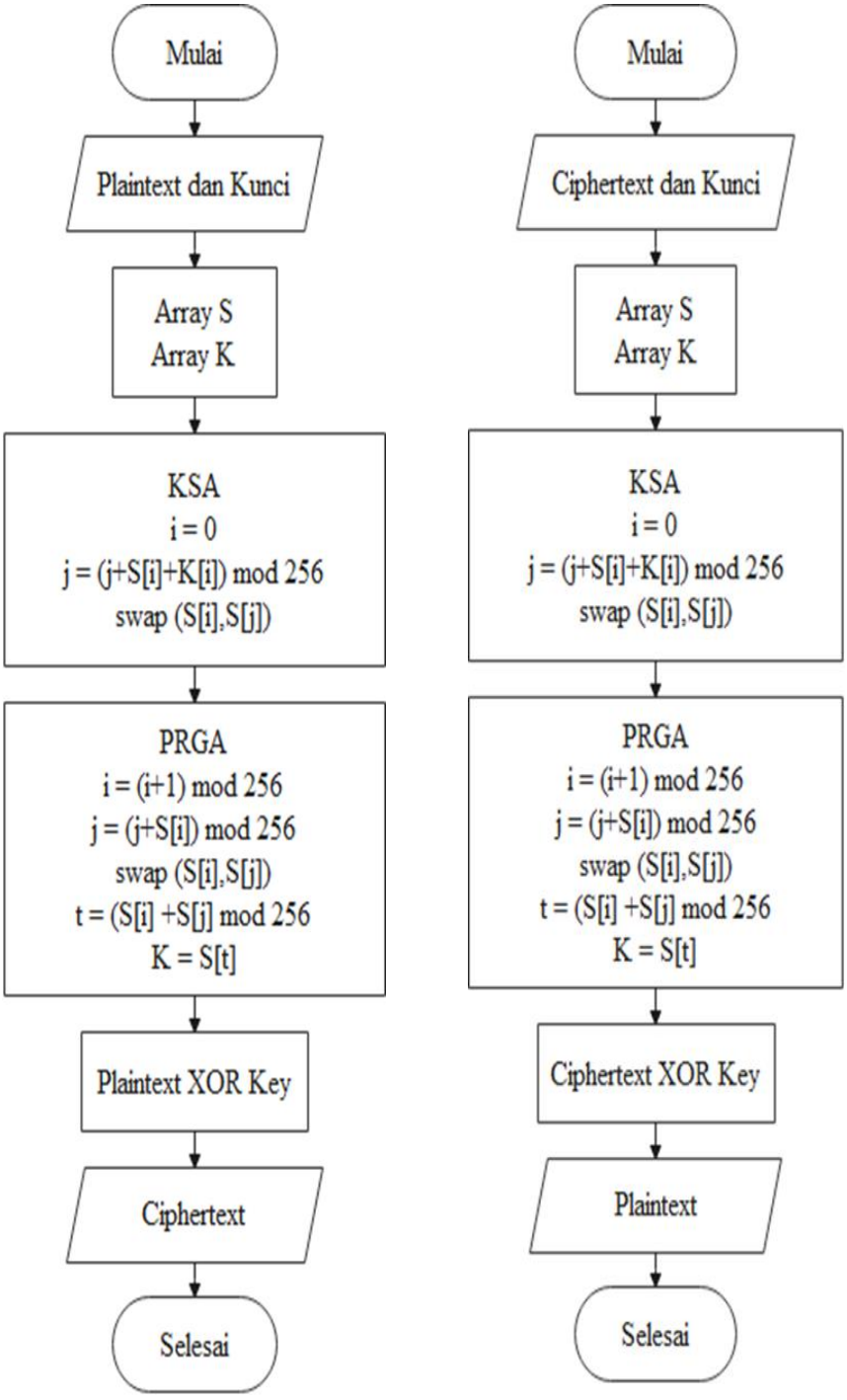
2.1 Pengumpulan Data (*Data Collecting*)

Untuk mendapatkan data dan in*form*asi yang dibutuhkan terkait pengamanan data pasien pada klinik Pratama Siti Rahmah, ada 2 tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Observasi
   Observasi dalam penelitian ini dilakukan dengan tinjauan langsung ke klinik Pratama Siti Rahmah. Di klinik tersebut dilakukan analisis masalah yang dihadapi selama ini terkait keamanan data pasien.
2. Wawancara
   Setelah melakukan observasi, peneliti melakukan wawancara kepada dokter dan para tenaga medis yang terlibat dalam proses pengolahan data pasien di klinik Pratama Siti Rahmah.

2.2 *Flowchart* Enkripsi dan Dekripsi Algoritma *Rivest Code (RC4)*

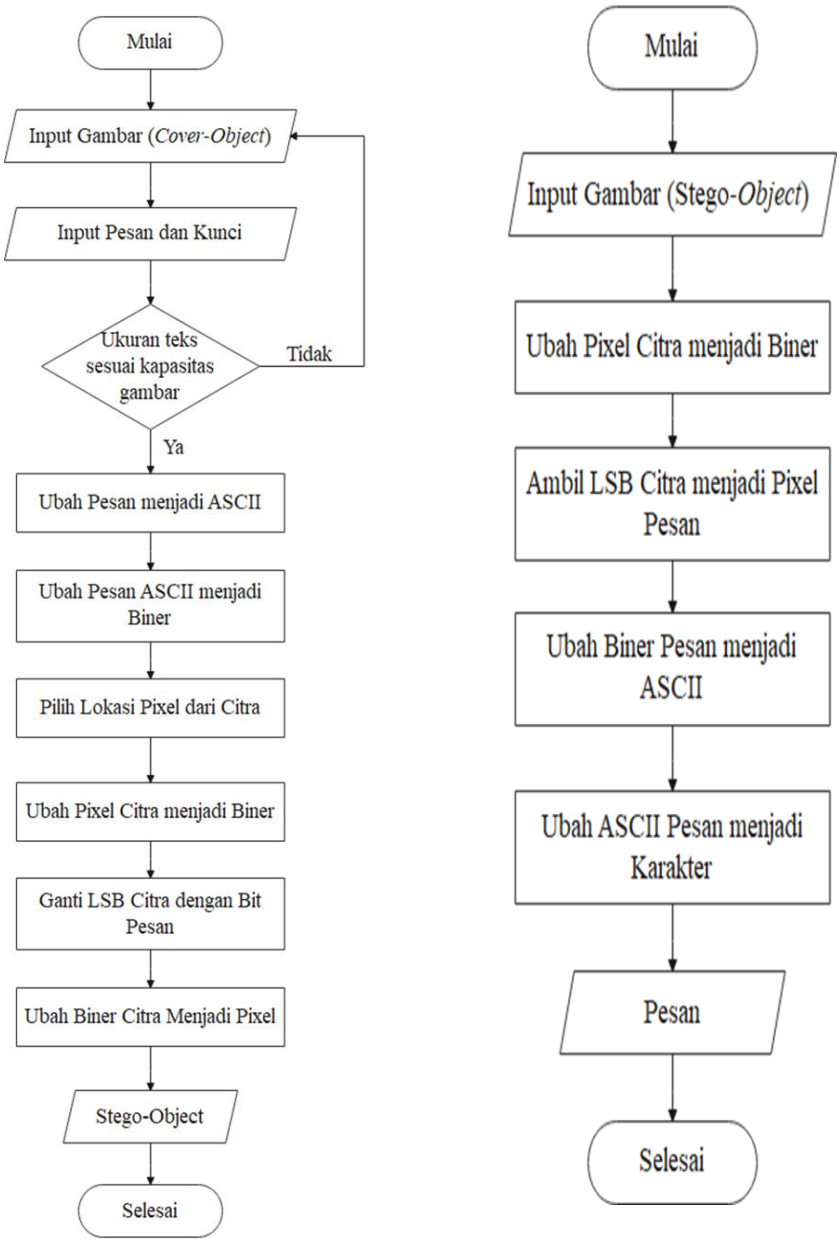
Flowchart algoritma RC4 merupakan keterangan yang lebih rinci tentang bagaimana prosedur sesungguhnya yang dilakukan oleh suatu metode. *Flowchart* ini menggambarkan urutan dari suatu prosedur pemecahan masalah. Berikut ini adalah *flowchart* enkripsi dan dekripsi dari algoritma *Rivest Code 4 (RC4)* adalah sebagai berikut :



Gambar 2.1 Flowchart Enkripsi dan Dekripsi Algoritma RC4

2.3 *Flowchart* Encode dan Decode Metode *Least Significant Bit (LSB)*

Flowchart Encode dan Decode merupakan keterangan yang lebih rinci tentang bagaimana prosedur sesungguhnya yang dilakukan oleh metode LSB. *Flowchart* ini menggambarkan urutan dari suatu prosedur penyisipan pesan pada media citra. Berikut ini adalah *flowchart* encode dan decode dari *Least Significant Bit (LSB)* adalah sebagai berikut :



Gambar 2.2 *Flowchart* Encode dan Decode Metode LSB

2.4 Penyelesaian Masalah Dengan Algoritma *Rivest Code (RC4)*

Berikut ini adalah penyelesaian masalah mengenai pengamanan data pasien dengan algoritma *Rivest Code 4 (RC4)* :

Plainteks : Data_Pasien
Kunci : Ramah

2.4.1 Enkripsi

1. Inisialisasi S-Box dengan panjang 256 byte, dengan S[0]=0, S[1]=1, S[2]=2, S[3]=3,....., S[255]=255 sehingga array S menjadi:

Tabel 3.2 Inisialisasi S-Box Dengan Panjang 256 byte

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

2. Inisialisasi kunci array Ki. Misalkan kunci terdiri dari 8 byte yaitu “Ramah” maka kalimat akan di ubah kedalam bentuk desimal. Setelah itu, ulang kunci sampai memenuhi seluruh array K sehingga array K menjadi:

Tabel 3.3 Inisialisasi Kunci Array Ki

i	0	1	2	3	4	...	255
Char	R	a	m	a	h	...	R
Dec	82	97	109	97	104	...	82

3. Berikutnya mencampur operasi dimana akan menggunakan variabel i dan j ke index array S[i] dan K[i]. Pertama diberi nilai inisial untuk i dan j dengan 0. Operasi pencampuran adalah pengulangan rumusan (j+S[i]+K[i]) mod 256 yang diikuti dengan penukaran S[i] dengan S[j].

   i = 0 to 256

   J = (j+S[i]+K[i]) mod 256

   Swap S[i] dan S[j]

   Dengan algoritma seperti diatas maka dengan nilai awal i = 0 sampai 255 akan menghasilkan array S seperti berikut ini:

   Iterasi ke-1:

   i = 0, maka

   j = (j+S[i]+K[i]) mod 256
   = (j+S[0]+K[0]) mod 256
   = (0+0+82) mod 256
   = 82 mod 256
   = 82

   Swap S[0] dan S[82]

   …

   …

   Iterasi ke-256:

   i = 255, maka

   j = (j+S[i]+K[i]) mod 256
   = (j+S[255]+K[255]) mod 256
   = (236+255+82) mod 256
   = 573 mod 256
   = 61

   Swap S[255] dan S[61]

   Hasil yang didapat setelah melakukan seluruh iterasi dari 0 sampai dengan iterasi ke 255 dan melakukan

pertukaran S-Box (Swap) adalah sebagai berikut :

Tabel 3.4 Tabel hasil pertukaran S-Box (Swap)

82	180	35	135	243	74	177	37	142	255	91	199	64	174	36	133
246	116	231	98	200	62	193	57	185	36	159	39	164	41	153	25
166	40	178	39	172	62	197	84	206	88	239	123	15	142	29	185
74	227	103	251	256	50	208	89	242	152	51	214	100	2	173	77
245	136	43	219	128	45	197	109	34	204	126	27	200	130	49	232
138	60	251	175	107	18	201	141	70	7	179	111	56	246	188	109
46	252	191	138	64	6	217	161	113	44	247	207	156	113	49	1
222	176	138	79	36	6	221	188	134	96	71	35	7	214	181	161
130	107	63	35	20	250	232	193	170	160	139	126	92	74	69	53
45	16	3	3	248	245	221	213	218	212	214	195	192	202	201	208
194	196	211	215	227	218	225	245	254	15	11	23	48	62	84	85
102	132	151	178	184	206	241	9	41	52	79	119	148	185	201	233
22	56	98	119	156	206	245	36	62	104	159	203	255	30	77	137
186	243	23	75	140	194	0	41	98	168	227	38	84	146	221	29
101	152	219	43	112	189	245	61	146	220	46	107	184	18	97	184
250	76	171	255	91	162	249	93	182	23	99	191	40	134	236	61

4. Proses pembuatan pseudorandom byte:

i = (i+1) mod 256
j = (j+S[i]) mod 256
swap S[i] dan S[j]
t = (S[i]+S[j]) mod 256
K = S[t]
Iterasi ke-1 : (Plainteks : D)
i = (i + 1) mod 256
  = (0 + 1) mod 256
  = 1
j = (j+S[i]) mod 256
  = (0 + S[1]) mod 256
  = (0 + 180) mod 256
  = 180
swap S[1] dan S[180]
t = (S[i]+S[j]) mod 256
  = (S[1]+S[180]) mod 256
  = (180 + 184) mod 256
  = 364 mod 256
  = 108
Key[0] = S[108] = 156
…
…
Iterasi ke-10 : (Plainteks : n)
i = (i + 1) mod 256
  = (10 + 1) mod 256
  = 11
j = (j+S[i]) mod 256
  = (89 + S[11]) mod 256
  = (89 + 199) mod 256

= 288 mod 256
= 32
swap S[11] dan S[32]
t = (S[i]+S[j]) mod 256
= (S[11]+S[32]) mod 256
= (199 + 166) mod 256
= 365 mod 256
= 109
Key[10] = S[109] = 113

Tabel 3.5 Enkripsi Plainteks

Plainteks	*Binary*		*Key* (K)		XOR		*Ciphertext*
	Dec	Bin	Dec	Bin	Dec	Bin	
D	68	01000100	156	10011100	216	11011000	Ø
a	97	01100001	200	11001000	169	10101001	©
t	116	01110100	219	11011011	175	10101111	¯
a	97	01100001	185	10111001	216	11011000	Ø
_	95	01011111	174	10101110	174	10101110	®
P	80	01010000	96	01100000	48	00110000	0
a	97	01100001	194	11000010	163	10100011	£
s	115	01110011	203	11001011	184	10111000	,
i	105	01101001	107	01101011	2	00000010	
e	101	01100101	191	10111111	218	11011010	Ú
n	110	01101110	113	01110001	30	00011110	

2.4.2 Dekripsi

Selanjutnya adalah melakukan dekripsi ciphertext menjadi plaintext yaitu dengan meng-XOR-kan dengan kunci yang sama seperti berikut ini:

Cipherteks : Ø©¯Ø®0£, Ú
Kunci : Ramah

1. Inisialisasi S-Box dengan panjang 256 byte, dengan S[0]=0, S[1]=1, S[2]=2, S[3]=3,....., S[255]=255 sehingga array S menjadi:

Tabel 3.2 Inisialisasi S-Box Dengan Panjang 256 byte

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

2. Inisialisasi kunci array Ki. Misalkan kunci terdiri dari 8 byte yaitu “Ramah” maka kalimat akan di ubah kedalam bentuk desimal. Setelah itu, ulang kunci sampai memenuhi seluruh array K sehingga array K menjadi:

Tabel 3.3 Inisialisasi Kunci Array Ki

i	0	1	2	3	4	...	255
Char	R	a	m	a	h	...	R
Dec	82	97	109	97	104	...	82

3. Berikutnya mencampur operasi dimana akan menggunakan variabel i dan j ke index array S[i] dan K[i]. Pertama diberi nilai inisial untuk i dan j dengan 0. Operasi pencampuran adalah pengulangan rumusan (j+S[i]+K[i]) mod 256 yang diikuti dengan penukaran S[i] dengan S[j].

i = 0 to 256
J = (j+S[i]+K[i]) mod 256
Swap S[i] dan S[j]
Iterasi ke-1:
i = 0, maka
j = (j+S[i]+K[i]) mod 256
= (j+S[0]+K[0]) mod 256
= (0+0+82) mod 256
= 82 mod 256
= 82
Swap S[0] dan S[82]
…
…
Iterasi ke-256:
i = 255, maka
j = (j+S[i]+K[i]) mod 256
= (j+S[255]+K[255]) mod 256
= (236+255+82) mod 256
= 573 mod 256
= 61
Swap S[255] dan S[61]

Hasil yang didapat setelah melakukan seluruh iterasi dari 0 sampai dengan iterasi ke 255 dan melakukan pertukaran S-Box (Swap) adalah sebagai berikut :

Tabel 3.4 Tabel hasil pertukaran S-Box (Swap)

82	180	35	135	243	74	177	37	142	255	91	199	64	174	36	133
246	116	231	98	200	62	193	57	185	36	159	39	164	41	153	25
166	40	178	39	172	62	197	84	206	88	239	123	15	142	29	185
74	227	103	251	256	50	208	89	242	152	51	214	100	2	173	77
245	136	43	219	128	45	197	109	34	204	126	27	200	130	49	232
138	60	251	175	107	18	201	141	70	7	179	111	56	246	188	109
46	252	191	138	64	6	217	161	113	44	247	207	156	113	49	1
222	176	138	79	36	6	221	188	134	96	71	35	7	214	181	161
130	107	63	35	20	250	232	193	170	160	139	126	92	74	69	53
45	16	3	3	248	245	221	213	218	212	214	195	192	202	201	208
194	196	211	215	227	218	225	245	254	15	11	23	48	62	84	85
102	132	151	178	184	206	241	9	41	52	79	119	148	185	201	233
22	56	98	119	156	206	245	36	62	104	159	203	255	30	77	137
186	243	23	75	140	194	0	41	98	168	227	38	84	146	221	29
101	152	219	43	112	189	245	61	146	220	46	107	184	18	97	184
250	76	171	255	91	162	249	93	182	23	99	191	40	134	236	61

4. Proses pembuatan pseudorandom byte:

i = (i+1) mod 256
j = (j+S[i]) mod 256
swap S[i] dan S[j]
t = (S[i]+S[j]) mod 256
K = S[t]
i = (i+1) mod 256
j = (j+S[i]) mod 256
swap S[i] dan S[j]
t = (S[i]+S[j]) mod 256
K = S[t]
Iterasi ke-1 : (Plainteks : Ø)
i = (i + 1) mod 256
= (0 + 1) mod 256
= 1
j = (j+S[i]) mod 256
= (0 + S[1]) mod 256
= (0 + 180) mod 256
= 180
swap S[1] dan S[180]
t = (S[i]+S[j]) mod 256
= (S[1]+S[180]) mod 256
= (180 + 184) mod 256
= 364 mod 256
= 108
Key[0] = S[108] = 156
…
…
Iterasi ke-10 : (Plainteks :)
i = (i + 1) mod 256
= (10 + 1) mod 256
= 11
j = (j+S[i]) mod 256
= (89 + S[11]) mod 256
= (89 + 199) mod 256
= 288 mod 256
= 32
swap S[11] dan S[32]
t = (S[i]+S[j]) mod 256
= (S[11]+S[32]) mod 256
= (199 + 166) mod 256
= 365 mod 256
= 109
Key[10] = S[109] = 113

Tabel 3.6 Dekripsi Ciphertext

Ciphertext	Dec	Bin	*Key*		XOR		*Plaintext*
			Dec	Bin	Bin	Dec	
Ø	216	11011000	156	10011100	01000100	68	D
©	169	10101001	200	11001000	01100001	97	a
¯	175	10101111	219	11011011	01110100	116	t
Ø	216	11011000	185	10111001	01100001	97	a
®	174	10101110	174	10101110	01011111	95	_
0	48	00110000	96	01100000	01010000	80	P
£	163	10100011	194	11000010	01100001	97	a
,	184	10111000	203	11001011	01110011	115	s
	2	00000010	107	01101011	01101001	105	i
Ú	218	11011010	191	10111111	01100101	101	e
	30	00011110	113	01110001	01101110	110	n

2.5 Proses Penyisipan Pesan Dengan Metode *Least Significant Bit (LSB)*

Tabel 3.7 Contoh penyisipan Bit Biner ke Piksel Citra

Nilai Piksel Citra	Konversi ke Biner	Bit Biner	Hasil Penyisipan	Hasil Stegano
Piksel 1				
R = 201	11001001	-	201	11001001
G = 196	11000100	1	197	11000101
B = 192	11000000	1	193	11000001
Piksel 2				
R = 94	01011110	1	95	01011111
G = 57	00111001	-	57	00111001
B = 49	00110001	-	49	00110001
Piksel 3				
R = 224	11100000	1	225	11100001
G = 224	11100000	1	225	11100001
B = 226	11100010	1	227	11100011
Piksel 4				
R = 222	11011110	1	223	11011111
G = 226	11100010	1	227	11100011
B = 225	11100001	-	115	11100001
Piksel 5				
R = 223	11011111	-	223	11011111
G = 221	11011101	-	221	11011101
B = 222	11011110	1	223	11011111
Piksel 6				
R = 222	11011110	1	223	11011111
G = 222	11011110	1	223	11011111
B = 224	11100000	1	225	11100001
Piksel 7				
R = 190	10111110	1	191	10111111
G = 187	10111011	-	187	10111011
B = 182	10110110	1	183	10110111
Piksel 8				
R = 187	10111011	-	187	10111011
G = 182	10110110	1	183	10110111
B = 179	10110011	-	179	10110011
Piksel 9				
R = 174	10101110	1	175	10101111
G = 168	10101000	1	169	10101001
B = 168	10101000	1	169	10101001
Piksel 10				
R = 188	10111100	1	189	10111101
G = 148	10010100	1	149	10010101
B = 136	10001000	1	137	10001001
Piksel 11				
R = 225	11100001	-	225	11100001
G = 229	11100101	-	229	11100101
B = 232	11101000	1	233	11101001
Piksel 12				
R = 223	11011111	-	223	11011111
G = 225	11100001	-	225	11100001
B = 224	11100000	1	225	11100001

Tabel 3.8 Citra setelah disisipkan bit cipher

201 197 193	95 57 49	225 225 227
223 227 115	223 221 223	223 223 225
191 187 183	187 183 179	175 169 169
189 149 137	225 229 233	223 225 225

4. ANALISA DAN HASIL

Merupakan kegiatan akhir dari proses penerapan sistem, dimana sistem ini akan dioperasikan secara menyeluruh. Sebelum sistem benar-benar bisa digunakan dengan baik, sistem harus melalui tahap pengujian analisa dan hasil terlebih dahulu untuk menjamin tidak ada kendala yang muncul pada saat sistem digunakan. Hasil dari penelitiaan yang dilakukan adalah sebagai berikut :

3.1. Tampilan *Form Login*

Tampilan menu *login* adalah tampilan untuk masuk ke menu utama. *User* harus melakukan *login* terlebih dahulu dengan cara menginput *username* dan *password* dengan benar sesuai dengan *database*. Berikut ini adalah tampilan menu *login* :

Gambar 3.1 Tampilan *Form Login*

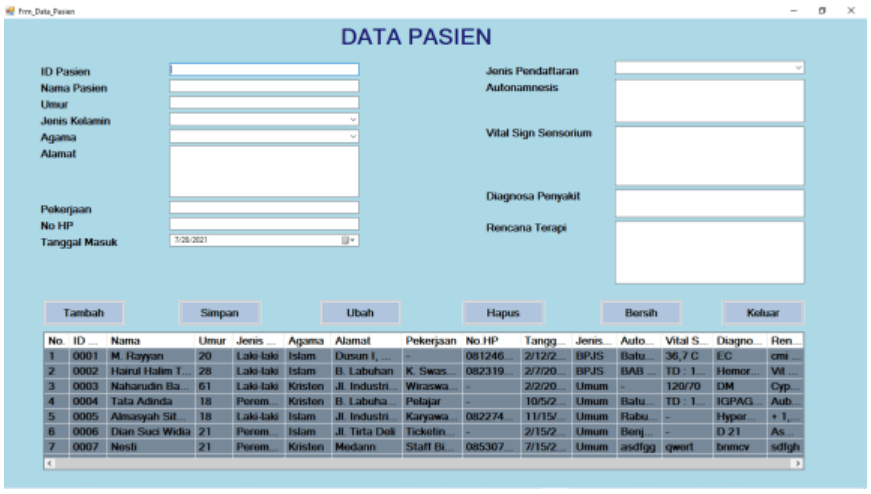
3.2 Tampilan *Form* Menu Utama

Tampilan *form* menu utama adalah tampilan awal sistem untuk melakukan pengolahan data dalam pengamanan data pasien. Berikut ini adalah tampilan menu utama :

Gambar 3.2 Tampilan *Form* Menu Utama

3.3 Tampilan *Form* Data Pasien

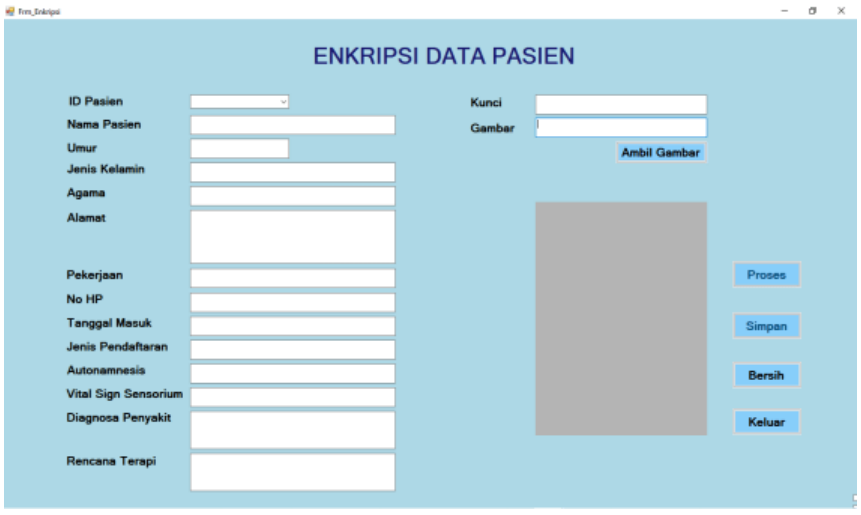
Tampilan *form* menu data pasien adalah *form* untuk menginput data pasien. Berikut ini adalah tampilan *form* data pasien :



Gambar 3.3 Tampilan *Form* Data Pasien

3.4 Tampilan *Form* Enkripsi

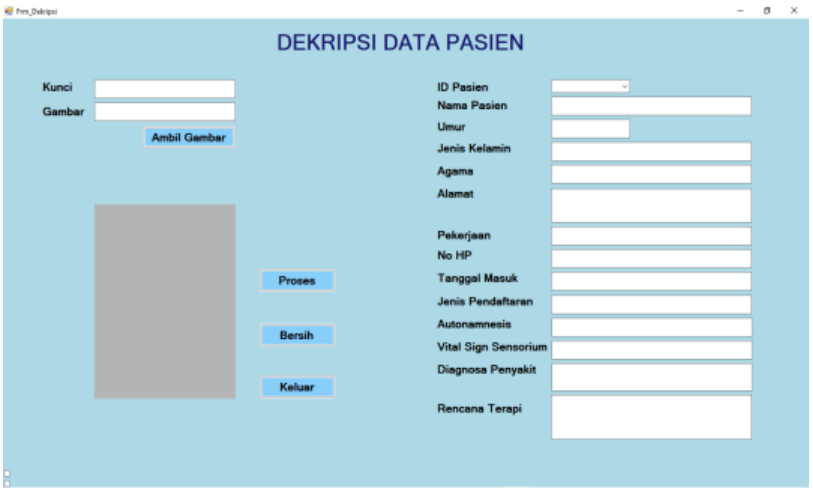
Tampilan *form* enkripsi adalah tampilan untuk proses penyandian dan penyisipan pesan serta penyimpanan hasil *encode*. Berikut ini adalah tampilan dari *form* enkripsi :



Gambar 3.4 Tampilan *Form* Enkripsi

3.5 Tampilan *Form* Dekripsi

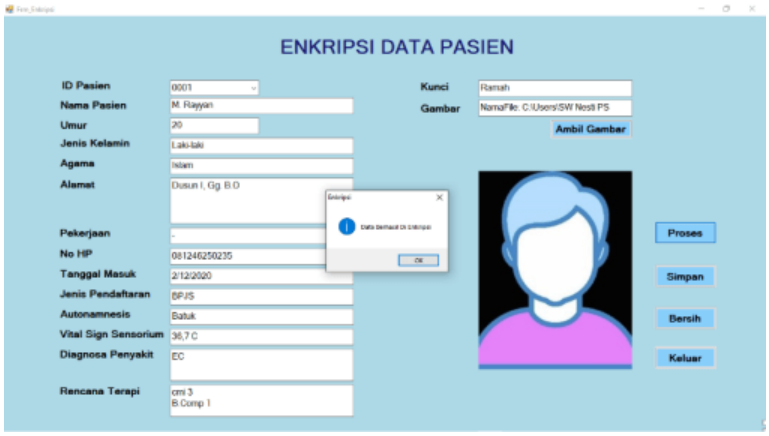
Tampilan *form* dekripsi adalah tampilan untuk menampilkan isi pesan dari hasil *encode* dan mengubah kembali data dalam bentuk semula. Berikut ini adalah tampilan dari *form* dekripsi :



Gambar 3.5 Tampilan *Form* Dekripsi

3.6 Tampilan Proses Enkripsi

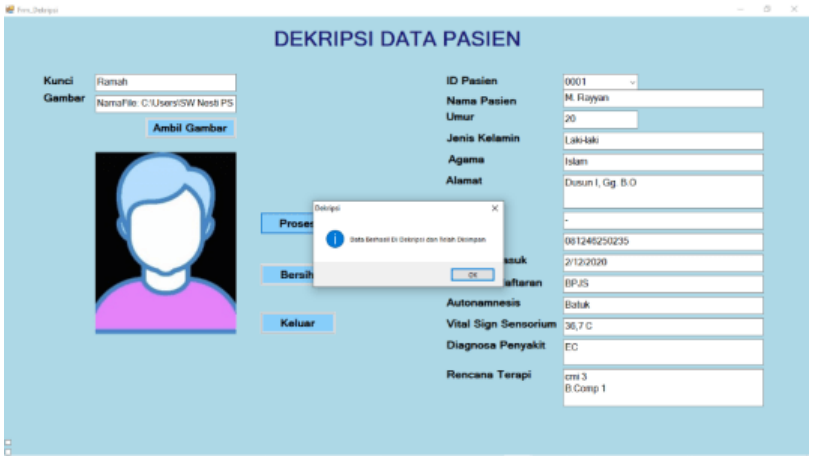
Berikut ini adalah tampilan dari hasil proses enkripsi mengunakan algoritma RC4 dan menyisipkannya kedalam gambar menggunakan metode LSB :



Gambar 3.6 Tampilan Proses Enkripsi

3.7 Tampilan Proses Dekripsi

Berikut ini adalah tampilan dari hasil proses dekripsi mengunakan algoritma RC4 dan mengeluarkan data dalam gambar menggunakan metode LSB :



Gambar 3.7 Tampilan Proses Dekripsi

5. KESIMPULAN

Berdasarkan masalah yang telah dipaparkan pada pembahasan sebelumnya maka dapat ditarik kesimpulan bahwa dibangunnya sistem pengamanan data pasien di klinik Pratama Siti Rahmah menggunakan algoritma RC4 dan metode LSB berhasil diterapkan.

Dalam penerapan sistem keamanan data pasien di klinik Pratama Siti Rahmah khususnya terkait data rekam medis pasien dapat digunakan dan kebutuhan pada sistem telah sesuai dengan kebutuhan dalam pengamanan data pasien di klinik Pratama Siti Rahmah.

UCAPAN TERIMA KASIH

Pada kesempatan ini penulis mengucapkan banyak terimakasih kepada kedua orang tua yang telah banyak memberikan dukungan moril dan materil, tidak terkecuali doa yang senantiasa dipanjatkan sehingga penulis dapat menyelesaikan penelitian ini.

Penyusunan jurnal ini juga tidak terlepas dari bantuan berbagai pihak. Oleh karena itu dengan segala kerendahan hati, diucapkan terimakasih yang sebesar-besarnya kepada: Bapak Azanuddin,S.Kom., M.Kom selaku Dosen Pembimbing I, kepada Ibu Vina Winda Sari S.E., M.Ak. selaku Dosen Pembimbing II yang telah banyak membantu dalam memberikan arahan dan bimbingan.

REFERENSI

[1] E. Helmud, "Kombinasi Kriptografi Rc4 Dan Steganografi Lsb Pada Citra Digital Dengan Format Bitmap Untuk Menjaga Keamanan Pesan," *Techno Xplore J. Ilmu Komput. dan Teknol. Inf.*, vol. 2, no. 2, pp. 20–27, 2018, doi: 10.36805/technoxplore.v2i2.304.

[2] A. K. Harsa, "Keamanan Data Dengan Menggunakan Algoritma Rivest Code 4 (Rc4) Dan Steganografi Pada Citra Digital," *Inform. Mulawarman Februari*, vol. 9, no. 1, 2014.

[3] M. M. Assyahid, R. Rihartanto, and D. S. B. Utomo, "Implementasi Steganografi Pesan Text ke Dalam Audio Dengan Metode Spread Spectrum," *Juristek*, vol. 3, no. 2, pp. 27–34, 2018.

[4] 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.

[5] N. Rismawati, "Analisis dan Perancangan Simulasi Enkripsi dan Dekripsi pada Algoritma Steganografi untuk Penyisipan Pesan Text pada Image menggunakan Metode Least Significant Bit (LSB) Berbasis Cryptool2," *Fakt. Exacta*, vol. 12, no. 2, pp. 132–144, 2019.

BIBLIOGRAFI PENULIS

	Nama : Sri Winda Nesti Putri Santi Mendrofa NIRM : 2017020717 Program Studi : Sistem Informasi Deskripsi : Mahasiswa Stambuk 2017 pada Program Studi Sistem In*form*asi yang memiliki minat dan fokus dalam bidang keilmuan Kriptografi dan Desain Grafis. Email : sw.nesti@gmail.com
	Nama : Azanuddin, S.Kom., M.Kom Tempat/Tgl Lahir : Klambir Lima, 26 Juni 1989 Alamat : Dusun XI Gg. Mardisan Agama : Islam Jenis Kelamin : Laki-Laki No.Hp : 0813-7683-7222 Prestasi Dosen : - Bidang Keilmuan : Jaringan, Mobile, dan Sistem Terdistribusi Email : azdin.bpc@gmail.com
	Nama : Vina Winda Sari, SE., M.Ak. Tempat/Tgl : Medan, 19 Juni 1984 Alamat : Jalan Pinang Baris Gg.Keluarga No.145 H Lk. I Agama : Islam Jenis Kelamin : Perempuan No.Hp : 0813-7625-6397 Prestasi Dosen : - Bidang Keilmuan : Akuntansi Email : winda_vina@yahoo.co.id