Vol. , No. , Juni 2021, pp.
P-ISSN :
E-ISSN :

Implementasi Pengamanan Data Pasien RSU Mitra Sejati Medan Dengan Menggunakan Metode Rivest Code (Rc4)

Deta Sari Gultom*, Badrul Anwar, Suardi Yakub,****
*Program Studi Sistem Informasi, STMIK Triguna Dharma
** Program Studi Sistem Informasi, STMIK Triguna Dharma

Article Info
Article history:

Keyword:
Cryptography,
RC4,
Encryption,
Decryption,
Data Pasien.

ABSTRACT

RSU Mitra Sejati rumah sakit yang berada di Medan Johor, kota medan. Merupakan salah satu tempat yang menyediakan atau menyelengaraan upaya kesehatan yaitu setiap kegiatan untuk memelihara dan meningkatkan kesehatan. Disamping itu RSU Mitra Sejati memiliki keluhan data atau belum adanya pengunci keamanan data pasien dalam sistem yang digunakan. Sehingga timbulnya, penyerangan dan faktor kesengajaan pencurian data. Akibat yang muncul dari ancaman tersebut yakni kepercayaan pasien terhadap perlindungan data menjadi berkurang. Salah satu solusi untuk mengatasi keamanan dan kerahasian data pasien adalah dengan cara melakukan enkripsi terhadap data pasien yang akan diamankan dan waktu yang dipergunakan untuk penyandian data tidak lama dan tergantung berapa besar data pasien yang akan diamankan. Kriptografi merupakan cara yang dilakukan dengan menyandikan pesan asli menjadi pesan acak yang sulit dipahami. Didalam kriptografi terdapat dua proses utama yaitu enkripsi dan dekripsi. Metode yang dipakai adalah metode Rivest Code 4 (RC4) karena metode ini merupakan salah satu metode yang hasil penyandian (ciphertext) memiliki ukuran panjang karakter yang sama dengan pesan aslinya (plaintext)..Hasil pengujian menunjukkan bahwa sistem apalikasi keamanan dalam algoritma RC4 data pasien RSU Mitra Sejati data yang tersimpan di dalam database pada data pasien berhasil di enkripsi sehingga terjamin kerhasiaan dan keamaman data nya. Ketika orang lain ingin membuka nya maka akan sulit membacanya karena sudah di enkripsi dalam bentuk algoritma RC4.

First Author
Nama : Deta Sari Gultom
Program Studi : Sistem Informasi
STMIK Triguna Dharma
E-Mail : detagultom98@gmail.com

1. PENDAHULUAN

Masalah Kerahasiaan dan keamanan saat melakukan pertukaran data adalah hal yang sangat penting dalam komunikasi data, baik untuk tujuan keamanan bersama, maupun untuk privasi individu. Mereka mengiginkan agar data nya tidak di ketahui oleh pihak yang tidak berkepentingan. Ketika pesan disadap pada saat pengiriman data melalui email atau melalui jaringan lain nya maka data tersebut tidak akan berguna lagi, sebab data tersebut tidak ada perlindungan yang di terapkan dalam data tersebut[1].

Keamanan dan kerahasian data pada jaringan komputer saat ini menjadi isu yang sangat penting dan terus berkembang. Beberapa kasus menyangkut keamanan jaringan komputer saat ini menjadi suatu pekerjaan yang membutuhkan biaya penanganan yang sedemikan besar seperti, sistem perbankan, sistem bandar udara dan sistem yang lain setingkat atau setara dengan yang dibahas, membutuhkan tingkat keamanaan yang sedemikan tinggi.

Hal ini di sebabkan karena kemajuan bidang jaringan komputer yang terbuka untuk umum sehinggasiapapun, kapanpun, dimanapun mempunyai kesempatan untuk mengakses nya melalui internet[2].

Disamping itu RSU Mitra Sejati Medan juga memiliki keluhan atau belum adanya pengunci keamanan data pasien dalam sistem yang digunakan sehingga timbulnya ancaman natur, techicial (teknis) dan faktor kesengajaan pencurian data. Akibat yang muncul dari ancaman tersebut yakni kepercayaan pasien terhadap perlindungan data menjadi berkurang.

Maka ancaman ini dapat diatasi dengan mendayagunakan keamanaan data pasien terhadap sistem tersebut berupa fitur-fitur keamanan data pasien dalam sistem informasi berbasis komputersasi. Untuk menjaga keamanan data pasien dan kerahasian data pasien yang sangat penting maka di perlukan enkripsi dan dekripsi, guna agar tidak mudah dicuri atau *Hack* oleh pihak yang tidak berkepentingan, kecuali pihak yang berhak yang ingin melihat data pasien tersebut.

2. KAJIAN PUSTAKA

2.1 Kriptografi

Kriptografi berasal dari bahasa Yunani, *kripto* yang berarti rahasia dan *graphia* yang berarti catatan atau tulisan. Sedangkan berdasarkan dari istilahnya, kriptografi yakni ilmu dan seni untuk mengamankan pesan pada saat dikirim dari satu tempat ketempat sejarah yaitu pada masa-masa awal mula sejarah kriptografi ada setiap orang mungkin mempunyai cara yang berbeda dan unik dalam melindungi pesannya.[3]

Kriptografi pada dasarnya merupakan metode yang digunakan untuk mengamankan berbagai jenis data-data. Kriptografi mendukung keperluan dari dua bagian keamanan, yaitu *secrecy* (keamanan dalam kerahasiaan data) dan *authenticity* (kamanan dalam pemalsuan dan manipulasi data yang tidak harapkan). Kriptografi tidak bermakna hanya memberikan keamanan data saja, namun lebih ke sisi teknik-tekniknyayang lain. Kata “seni” tersebut berawal dari kenyataan.

Kriptografi terdiri atas dua proses, yaitu proses enkripsi yang merupakan proses penyandian pesan yang bisa dibaca *(plaintext)* menjadi pesan rahasia (*ciphertext*), dan proses dekripsi mengembalikan pesan rahasia *(ciphertext)* ke dalam bentuk semula *(plaintext)* sehingga pesan tadi dapat dibaca kembali oleh penerima pesan.

2.2 Algoritma Rivest Code (RC4)

RC4 merupakan salah satu jenis *stream cipher*, yaitu memproses unit atau *input* data pada satu saat. Unit atau data pada umumnya sebuah *byte* atau bahkan bisa berupa bit. Dengan cara ini *enkripsi* atau *dekripsi* dapat dilaksanakan pada panjang variabel. Algoritma ini tidak harus menunggu sejumlah *input* data tertentu sebelum diproses, atau menambahkan *byte* tambahan untuk meng*enkripsi*[4].

Cara kerja algoritma RC4 yaitu inisialisasi S-Box pertama, S[0], S[1], ..., S[255], dengan bilangan 0 sampai 255. Pertama isi secara berurutan S[0]=0, S[1]=1, ..., S[255]=255. Kemudian inisialisasi *arrray* lain (S-Box), misal *array* K dengan panjang 256. Isi *array* K dengan kunci diulangi sampai seluruh *array* K[0], K[1],...., K[255] terisi seluruhnya.

Pada algoritma RC4 memakai S-*Box* (kotak subsitusi) dengan *array* 256 *byte* dengan ukuran 8 x 8. Walaupun RC4 menggunakan S-*Box*, tapi operasi yang terjadi di dalam S-*Box* adalah operasi permutasi.

Initial S-*Box*

For r = 0 to 255 do
S[i] = I
T[i] = K [I*modkeylen*]

Kunci mempunyai peran utama dalam merubah isi dari S-*box* dengan operasi sebagai berikut:

```
j = 0
For I = 0 to 255 do
J = (J+ S[i] = K[i]) mod 256
Swap (S[i], S[j])
```

Untuk membangkitkan kunci *enkripsi* dengan proses:

```
I=j=0
While (true)
    I = ( i + 1 ) mod 256;
J = (j+S[i] ) mod 256;
Swap ( S[i] dan S[j] );
t = S (S[i]+S[j]) mod 256;
k = S[t];
```

Catatan "K" merupakan kunci yang beroperasi terhadap *plaintext,* sedangkan "K" merupakan kunci utama.

Algoritma dari *Rivest Code 4* (RC4) terus berlanjut. Hal ini terus diteliti untuk mendapatkan algoritma yang kuat dan tangguh. Algoritma *enkripsi Rivest Code 5* (RC5) merupakan kelanjutan dari algoritma *Rivest Code 4* (RC4) *yang dibuat oleh Ron Rivest* dari *Massacchusrtts Intitute of tecnology* (MIT), RSA data *Security inc*. Berhasil membuat algoritma baru setelah *Rivest Code 4* (RC4), yaitu *Rivest Code 5* (RC5). *Rivest Code 5* (RC5) dianalisis *(RSA Laborator)* untuk menghasilkan kelanjutan dari *Rivest Code 5* (RC5), yaitu *Rivest Code 6* (RC6) yang mungin bisa menggantikan DES.

XOR adalah operasi *Exclusive*-OR yang dilambangkan dengan tanda "⊕". Hasil dari operasi XOR akan bernilai bit "0" (nol) jika dua buah bit *input* memiliki nilai yang sama dan akan menghasilkan nilai bit "1" (satu) jika dua buah bit *input* memiliki nilai bit yang berbeda[5].

Penjumlahan bit *modulo* yang digunakan dalam penjumlahan dua buah bit bilangan yang sama panjang dan menghasilkan bilangan dengan panjang bit yang sama pula. Jika panjang bit bilangan lebih besar, maka bit bilangan yang berlebihan akan dibuang.

Contoh : 10111101 + 10010101 = **1** 01010010

3. METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Metode Penelitian merupakan proses atau cara ilmiah untuk mendapatkan data yang akan digunakan untuk menyelesaikan masalah dengan dengan mengadakan studi langsung kelapangan untuk mengumpulkan data.

Dalam penelitian ini, ada beberapa cara yang dilaksanakan untuk menyelesaikan penelitian. Adapun cara yang dilakukukan adalah :

1. Pengumpulan Data
   Dalam Teknik Pengumpulan Data terdapat beberapa yang dilakukan di antaranya yaitu:
   a. Observasi
      Dalam penelitian ini dilakukan observasi pra-riset telebih dahulu untuk mencari masalah yang terjadi di RSU Mitra Sejati Medan terkhusus dalam pengamanan data pasien, dari masalah tersebut masalah akan dirumusakan dalam penelitian ini sehingga dapat menemukan rumusan apa saja yang perlu dipersiapkan untuk bagaimana cara menyelesaikan masalah tersebut.
   b. Wawancara
      Sedangkan dalam mendapatkan data yang baik, dalam penelitian ini dilakukan wawancara kepada admin atau pihak-pihak yang terlibat langsung dalam penelitian ini. Selain itu juga, penelitian mencoba mencari data sekunder dengan melakukan surfing di mesin pencarian terkait hal-hal penting dalam pengamanan data pasien.
2. Studi Kepustakaan
   Pada bagian ini dijelaskan mengenai tahapan penelitian yang akan dilakukan. Tahapan-tahapan ini disesuaikan dengan kebutuhan secara berurutan. Metode penelitian ini

meliputi pengumpulan data, penerapan perhitungan, implementasi algoritma RC4 dan pengujian hasil enkripsi dan dekripsi *database*.

3.2 Algoritma Sistem

Algoritma sistem merupakan penjelasan langkah-langkah dalam penyelesaian masalah dalam perancangan sistem keamanan data pasien dengan menggunakan Algoritma RC4. Hal ini dilakukan untuk meningkatkan keamanan data pasien.

3.2.1 Penyelesaian

Tabel 3.2 Sampel Data Pasien Dari Perusahaan

NO	KODE PASIEN	NAMA PASIEN	TGL.MASUK	JENIS PENYAKIT	DOKTER
1	JMP101	RENO PURBA	23/12/2001	HIPERTENSI	dr.Anita
2	ASL201	ARIANTO	15/09/1998	DIABETES	dr.Anita
3	JMP202	RATNA	13/10/1978	DIABETES	dr.Agus
4	UMP201	JUMIHAR	30/12/1985	DIABETES	dr.Agus
5	UML303	MULYANTI	10/10/1966	TIPES	dr.Danang
6	ASP103	INTAN	23/12/1977	HIPERTENSI	dr.Anita
7	ASP304	ROHMA	20/05/1967	TIPES	dr.Danang
8	UMP202	EKO	19/02/1998	DIABETES	dr.Agus
9	JML103	KUSYUDI	11/11/1989	HIPERTENSI	dr.Anita
10	JML204	SANTO	18/10/1998	DIABETES	dr.Agus
11	JMP105	ARIS	10/11/1997	DIABETES	dr.Danang
12	ASL203	SANJAYA	17/02/1989	DIABETES	dr.Agus
13	ASL305	DENIIS	20/08/1996	HIPERTENSI	dr.Anita
14	ASP206	NITAYA	08/09/1889	TIPES	dr.Agus
15	JML104	KENJO	19/10/1995	TIPES	dr.Danang

3.2.2 Proses Enkripsi Dan Dekripsi RC4

Algoritma RC4 adalah algoritma kriptografi simetri. Disebut algoritma kriptografi simetri karena menggunakan kunci yang sama untuk meng*enkripsi* ataupun men*dekripsi* suatu pesan, data, ataupun informasi. Kunci *enkripsi* didapat dari sebuat 256 bit *state-array* yang diinisialisasi dengan sebuah *key* tersendiri dengan panjang 1-256 bit.

3.2.3 Perhitungan Metode *RC4*

Dalam penyelesaian penerapan metode enkripsi Rc4, dalam contoh hal studi kasus ini kita mengambil sampel data nama pasien diatas untuk diterapkan dalam metode enkripsi dan dekripsi rc4.

Plaintext *: RENO PURBA*
Kunci *: MITRA*

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

Tabel 3.3 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
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 5 byte dengan kunci array Ki, dimana kunci terdiri dari 5 byte ialah ”MITRA” jadi kalimat yang akan diubah ke dalam bentuk dan diulangi kunci sampai memenuhi seluruh array K sehingga array K menjadi:

Tabel 3.4 Inisialisasi *Array* K

77	73	84	82	65	77	73	84	82	65	77	73	84	82	65	77
73	84	82	65	77	73	84	82	65	77	73	84	82	65	77	73
84	82	65	77	73	84	82	65	77	73	84	82	65	77	73	84
82	65	77	73	84	82	65	77	73	84	82	65	77	73	84	82
65	77	73	84	82	65	77	73	84	82	65	77	73	84	82	65
77	73	84	82	65	77	73	84	82	65	77	73	84	82	65	77
73	84	82	65	77	73	84	82	65	77	73	84	82	65	77	73
84	82	65	77	73	84	82	65	77	73	84	82	65	77	73	84
82	65	77	73	84	82	65	77	73	84	82	65	77	73	84	82
65	77	73	84	82	65	77	73	84	82	65	77	73	84	82	65
77	73	84	82	65	77	73	84	82	65	77	73	84	82	65	77
73	84	82	65	77	73	84	82	65	77	73	84	82	65	77	73
84	82	65	77	73	84	82	65	77	73	84	82	65	77	73	84
82	65	77	73	84	82	65	77	73	84	82	65	77	73	84	82
65	77	73	84	82	65	77	73	84	82	65	77	73	84	82	65
77	73	84	82	65	77	73	84	82	65	77	73	84	82	65	77

3. Selanjutnya tahap Key Scheduling Algorithm (KSA) yaitu proses pengacakan kunci yang terjadwal, dengan pemberian nilai awal berdasarkan kunci enkripsi dengan mencampur operasi dimana akan menggunakan variabel i dan j ke index array S[i] dan K[i]. Langkah pertama di beri nilai inisial untuk i dan j dengan 0. Operasi pencampuran adalah pengulangan rumusan (j+S[i]+K[i] mod 256) yang diikuti dengan pertukaran S[i] dengan S[j]. Untuk contoh ini, karena menggunakan array dengan panjang 256 byte maka algoritma menjadi:

For i = 0 to 256
j = (j+S[i]+K[i]) mod 256
Swap S[i] dan S[j]

Algoritma yang seperti diatas maka dengan nilai awal i = 0 sampai i = 255 akan menghasilkan array S seperti dibawah ini :

Iterasi ke-1
i=0
j=0,maka
j = (j+S[i]+K[i]) mod 256
j = (0+S[0]+K[0]) mod 256
j = (0+0+77) mod 256
j = (77) mod 256
j = 77
Swap S[0] dan S[77]

Iterasi ke-2
i=1
j=77,maka
j = (j+S[i]+K[i]) mod 256
j = (77+S[1]+K[1]) mod 256
j = (77+1+73) mod 256
j = (151) mod 256
j = 151
Swap S[1] dan S[151]

Iterasi ke-3
i=2
j=151,maka
j = (j+S[i]+K[i]) mod 256
j = (151+S[2]+K[2]) mod 256
j = (151+2+84) mod 256
j = (237) mod 256
j = 237
Swap S[2] dan S[237]

Iterasi ke-4
i=3
j=237,maka
j = (j+S[i]+K[i]) mod 256
j = (237+S[3]+K[3]) mod 256
j = (237+3+82) mod 256
j = (322) mod 256
j = 66
Swap S[3] dan S[66]
……
……
……
Iterasike-256
i=255
j=69,maka
j = (j+S[i]+K[i]) mod 256
j = (69+S[255]+K[255]) mod 256
j = (69+255+77) mod 256
j = (401) mod 256
j = 145
Swap S[255] dan S[145]

Tabel 3.5 Tabel *Key Scheduling Algorithm* (KSA)

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

4. *Pseudo-random Generation Algorithm (PRGA)* adalah proses pembangkitan kunci, Enkripsi RC4 Berikut adalah proses enkripsi yaitu meng XOR-kan byte dengan plaintext “RENO PURBA”.Plaintext terdiri dari 9 karakter maka terjadi 9 iterasi. Sebelumnya iterasi harus di ubah menjadi bentuk bilangan biner

Tabel 3.6 Hasil Karakter ke Bilangan Biner

Plainteks	Decimal	Biner
R	82	01010010
E	69	01000101
N	78	01001110
O	79	01001111

	32	00100000
P	80	01010000
U	85	01010101
R	82	01010010
B	66	01000010
A	65	01000001

Inisialisasi i dan j dengan i = 0 ; j = 0
Iterasi ke-1
i = (i+1) mod 256
i = (0+1) mod 256
i =1
j = (j+S[i]) mod 256
j = (0+S[1]) mod 256
j =36
Swap S[1] dan S[36]
t = (S[i]+S[j]) mod 256
t = (S[1]+S[36]) mod 256
t = (155+36) mod 256
t =191
Key[0] = S[191]= 190Iterasi ke-2
i = (i+1) mod 256
i = (1+1) mod 256
i =2
j = (j+S[i]) mod 256
j = (36+S[2]) mod 256
j =17
Swap S[2] dan S[17]
t = (S[i]+S[j]) mod 256
t = (S[2]+S[17]) mod 256
t = (250+237) mod 256
t =231
Key[1] = S[231]= 186
Iterasi ke-3
i = (i+1) mod 256
i = (2+1) mod 256
i =3
j = (j+S[i]) mod 256
j = (17+S[3]) mod 256
j =42
Swap S[3] dan S[42]
t = (S[i]+S[j]) mod 256
t = (S[3]+S[42]) mod 256
t = (57+25) mod 256
t =82
Key[2] = S[82]= 177
Iterasi ke-4
i = (i+1) mod 256
i = (3+1) mod 256
i =4
j = (j+S[i]) mod 256
j = (42+S[4]) mod 256
j =14
Swap S[4] dan S[14]
t = (S[i]+S[j]) mod 256
t = (S[4]+S[14]) mod 256

t = (88+228) mod 256
t =60
Key[3] = S[60]= 92

Iterasi ke-5
i = (i+1) mod 256
i = (4+1) mod 256
i =5
j = (j+S[i]) mod 256
j = (14+S[5]) mod 256
j =144
Swap S[5] dan S[144]
t = (S[i]+S[j]) mod 256
t = (S[5]+S[144]) mod 256
t = (21+130) mod 256
t =151
Key[4] = S[151]= 8
Iterasi ke-6
i = (i+1) mod 256
i = (5+1) mod 256
i =6
j = (j+S[i]) mod 256
j = (144+S[6]) mod 256
j =170
Swap S[6] dan S[170]
t = (S[i]+S[j]) mod 256
t = (S[6]+S[170]) mod 256
t = (118+26) mod 256
t =144
Key[5] = S[144]= 130
Iterasi ke-7
i = (i+1) mod 256
i = (6+1) mod 256
i =7
j = (j+S[i]) mod 256
j = (170+S[7]) mod 256
j =60
Swap S[7] dan S[60]
t = (S[i]+S[j]) mod 256
t = (S[7]+S[60]) mod 256

t = (92+146) mod 256
t =238
Key[6] = S[238]= 78

Iterasi ke-8
i = (i+1) mod 256
i = (7+1) mod 256

i =8
j = (j+S[i]) mod 256
j = (60+S[8]) mod 256
j =131
Swap S[8] dan S[131]
t = (S[i]+S[j]) mod 256
t = (S[8]+S[131]) mod 256
t = (212+71) mod 256
t =27
Key[7] = S[27]= 213

Iterasi ke-9
i = (i+1) mod 256
i = (8+1) mod 256
i =9
j = (j+S[i]) mod 256
j = (131+S[9]) mod 256
j =170
Swap S[9] dan S[170]
t = (S[i]+S[j]) mod 256
t = (S[9]+S[170]) mod 256
t = (26+39) mod 256
t =65
Key[8] = S[65]= 182
Iterasi ke-10
i = (i+1) mod 256
i = (9+1) mod 256
i =10
j = (j+S[i]) mod 256
j = (170+S[10]) mod 256
j =40
Swap S[10] dan S[40]
t = (S[i]+S[j]) mod 256
t = (S[10]+S[40]) mod 256
t = (41+126) mod 256
t =167
Key[9] = S[167]= 164

T*abel 3.7 Pseudo-random Generation Algorithm (PRGA)*

Indec	Kunci	Biner
0	190	10111110
1	186	10111010
2	177	10110001
3	92	01011100
4	8	00001000
5	130	10000010
6	78	01001110
7	213	11010101
8	182	10110110
9	164	10100100

5. Enkripsi

Proses enkripsi yaitu meng XOR kan pseudorandom byte dengan Plaintext. Berikut merupakan proses enkripsi :

T*abel 3.8 XOR Proses Enkripsi*

Index	*Plainteks*	*Key*	*Ciphertext = P XOR K*	*Des (C)*	*ASCI*
0	01010010	10111110	11101100	236	ì
1	01000101	10111010	11111111	255	ÿ
2	01001110	10110001	11111111	255	ÿ
3	01001111	01011100	00010011	19	U
4	00100000	00001000	00101000	40	(
5	01010000	10000010	11010010	210	Ò
6	01010101	01001110	00011011	27	
7	01010010	11010101	10000111	135	‡
8	01000010	10110110	11110100	244	ô
9	01000001	10100100	11100101	229	å

Maka Enkripsi bilangan biner menggunakan RC4 adalah chiperteks = i y y U (O ‡ ô å

6. Dekripsi RC4

Proses dekripsi yaitu meng XOR kan pseudorandom byte dengan ciphertext nya adalah i y y U (O ‡ ô å. Ciphertext terdiri dari 9 iterasi yang akan diubah menjadi karakter-karakter bentuk bilangn biner.

Tabel 3.9 XOR Proses Dekripsi

Index	*Chipertext*	*Key*	*Plaintext = C XOR K*	*Des (C)*	*ASCI*
0	11101100	10111110	01010010	190	R
1	11111111	10111010	01000101	186	E
2	11111111	10110001	01001110	177	N
3	00010011	01011100	01001111	92	O
4	00101000	00001000	00100000	8	
5	11010010	10000010	01010000	130	P
6	00011011	01001110	01010101	78	U
7	10000111	11010101	01010010	213	R
8	11110100	10110110	01000010	182	B
9	11100101	10100100	01000001	164	A

Maka hasil pengembalian dari Enkripsi menjadi Dekripsi adalah plainteks RENO PURBA

4. IMPLEMENTASI DAN PENGUJIAN

4.1 Pengujian

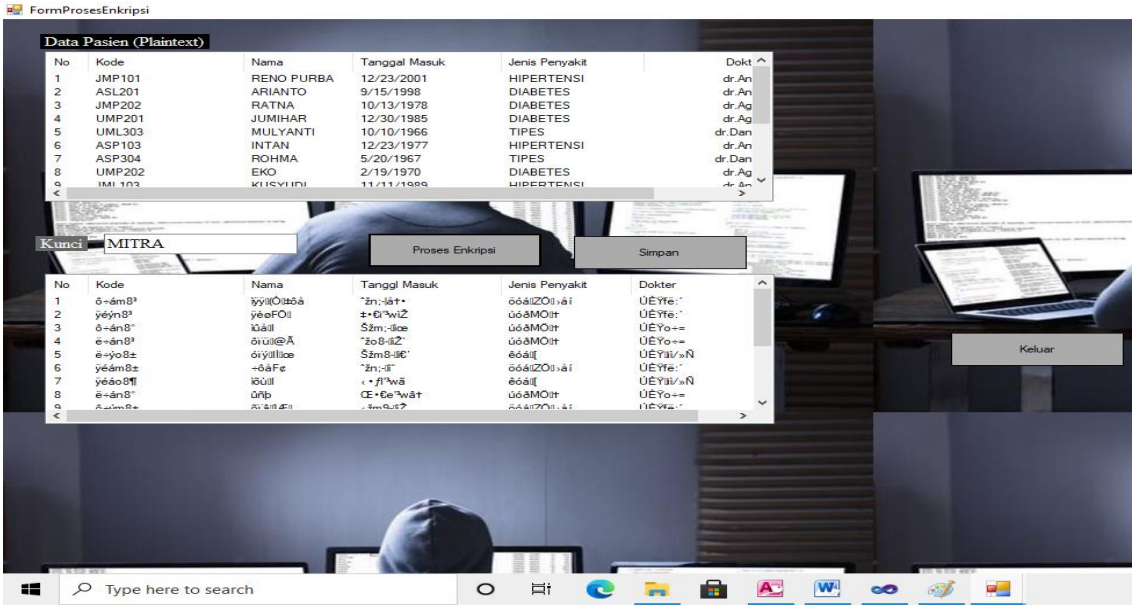
Dalam tahap ini akan dilakukan uji coba terhadap aplikasi Kriptografi metode *RC4* sebagai berikut :

Pengujian terhadap hasil enkripsi juga melihat apakah hasil enkripsi (*ciphertext*) mempunyai ukuran data yang sama dengan data yang asli atau tidak.

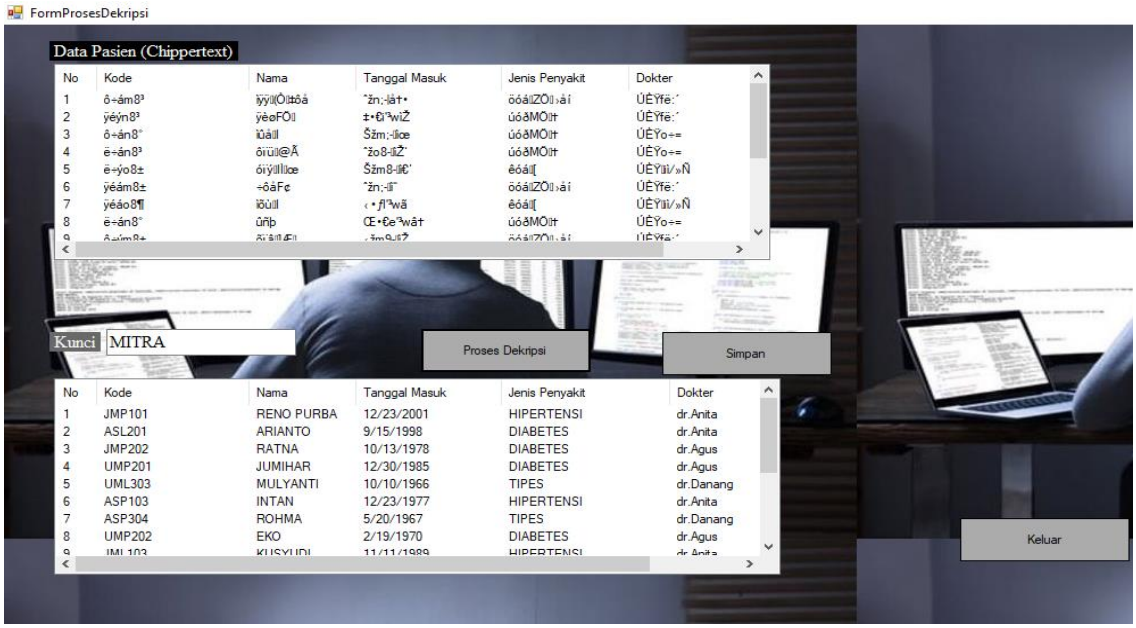
Tabel 5.1 Data Pasien

NO	KODE PASIEN	NAMA PASIEN	TGL.MASUK	JENIS PENYAKIT	DOKTER
1	JMP101	RENO PURBA	23/12/2001	HIPERTENSI	dr.Anita
2	ASL201	ARIANTO	15/09/1998	DIABETES	dr.Anita
3	JMP202	RATNA	13/10/1978	DIABETES	dr.Agus
4	UMP201	JUMIHAR	30/12/1985	DIABETES	dr.Agus
5	UML303	MULYANTI	10/10/1966	TIPES	dr.Danang
6	UMP203	ULINA	14/12/1999	DIABETES	dr.Agus
7	ASP304	ROHMA	20/05/1967	TIPES	dr.Danang
8	UMP202	EKO	19/02/1998	DIABETES	dr.Agus
9	JML103	KUSYUDI	11/11/1989	HIPERTENSI	dr.Anita
10	JML204	SANTO	18/10/1998	DIABETES	dr.Agus
11	JMP105	ARIS	10/11/1997	DIABETES	dr.Danang
12	ASL203	SANJAYA	17/02/1989	DIABETES	dr.Agus
13	ASL305	DENIIS	20/08/1996	HIPERTENSI	dr.Anita
14	ASP206	NITAYA	08/09/1889	TIPES	dr.Agus
15	JML104	KENJO	19/10/1995	TIPES	dr.Danang

Adapun hasil pengujian ditampilkan kedalam sebuah program pengamanan data pasien sebagai berikut.



Gambar 5.6 Hasil Enkripsi RC4



Gambar 5.7 Hasil Dekripsi RC4

5. KESIMPULAN

Berdasarkan analisa pada permasalahan yang terjadi dalam kasus yang diangkat tentang mengamankan data Pasien pada RSU Mitra Sejati, maka dapat ditarik kesimpulan sebagai berikut :

1. Berdasarkan hasil dari penelitian yang telah dilakukan dalam menerapkan metode RC4 untuk mengamankan data informasi pasien dapat digunakan dengan cara mengolah data pasien sesuai dengan algoritma RC4 dengan menggunakan frasa sandi yang kunci keamanan data pasien hanya diketahui oleh pihak yang berwenang saja khususnya bagian administrasi rumah sakit mitra sejati atau pengguna aplikasinya.
2. Dalam merancang dan membangun aplikasi pengamanan data pasien dapat menggunakan algoritma RC4 (*Rivest Code*) dilakukan dengan mengimplementasikan seluruh rancangan yang ada kedalam bahasa pemrograman *visual studio.*
3. Berdasarakan keseluruhan pengujian yang telah dilakukan. Data yang tersimpan di dalam database pada data pasien berhasil di enkripsi sehingga terjamin kerahasiaan dan keamanannya.
4. Tingkat keamanan aplikasi data pasien pada RSU. Mitra Sejati adalah 90 % aman karena telah di uji

UCAPAN TERIMA KASIH
Terima kasih kepada dosen pembimbing 1 Bapak Badrul Anwar, SE., S.kom., M.kom dosen pembimbing 2 Bapak Suardi Yakub, SE., MM dan pihak-pihak yang mendukung penyelesaian jurnal skripsi ini.

REFERENSI

[1] M. Marbun, "Implementasi Sistem Pengamanan Data Barang Pada Pt . Matahari Putra Prima, Tbk," *J. Mantik Penusa*, vol. 18, no. 2, pp. 1–10, 2015, [Online]. Available: http://e-jurnal.pelitanusantara.ac.id/index.php/mantik/article/view/128.

[2] M. K. Ruri Hartika Zain, S. Kom, "Perancangan Dan Implementasi Cryptography Dengan Metode Algoritma Rc4 Pada Type File Document Menggunakan Bahasa Pemrograman Visual Basic 6 . 0," *J. Momentum ISSN 1693-752X*, vol. 12, no. 1, pp. 71–81, 2012, [Online]. Available: https://ejournal.itp.ac.id/index.php/momentum/article/view/91.

[3] M. Steganografi and A. Pada, "IMPLEMENTASI ALGORITMA KRIPTOGRAFI RC4 DAN," vol. 04, no. 02, pp. 81–88, 2017.

[4] P. Aplikasi and P. Planing, "IMPLEMENTASI RC4 STREAM CIPHER UNTUK KEAMANAN BASIS DATA," no. January 2012, 2017.

[5] J. H. Lubis, "Implementasi Keamanan Data Dengan Metode Kriptografi XOR," *J. Sist. Inf. Kaputama*, vol. 2, no. 2, pp. 1–4, 2018.

BIOGRAFI PENULIS

NIRM : 2016020699
Nama : Deta Sari Gultom
Tempat/Tanggal Lahir : Janjimaria 04, Januari 1998
Jenis Kelamin : Perempuan
Agama : Kristen Protestan
Kewarganegaraan : Indonesia
E-mail : detagultom98@mail.com
N0/Hp : 082273573277
Bidang Keahlian : Pemrograman Berbasis Dekstop

NIDN : 0126017501
Nama : Badrul Anwar, S.E., S.Kom.,M.Kom
Tempat/Tanggal Lahir : Medan, 26 Januari 1975
Jenis Kelamin : Laki - Laki
Agama : Islam
Kewarganegaraan : Indonesia
E-mail : badrul.anwar@yahoo.com
N0/Hp : 08126086799
Bidang Keahlian : Komputer Akuntansi, Pemrograman Visual, dll

NIDN : 0106046601
Nama : Suardi Yakub, S.E., S.Kom.,MM
Tempat/Tanggal Lahir : Pariaman 06 April 1966
Jenis Kelamin : Laki - Laki
Agama : Islam
Kewarganegaraan : Indonesia
E-mail : yakubsuardi@gmail.com
N0/Hp : 085359587766
Bidang Keahlian : Manajemen, Sistem Informasi, dll