JITEKH, Vol 7, No 2, Tahun 2019, 51-56

ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

PENERAPAN ALGORITMA RC4 DAN RAIL FENCE UNTUK ENKRIPSI DATABASE MAHASISWA PADA KAMPUS POLTEKKES KEMENKES MEDAN

Rananda Satia Siregar[1], Munjiat Setiani Asih[2], Nur Wulan[3]

[1, 2, 3)]Fakultas Teknik Informatika, Universitas Harapan, Medan

Email : [1)]siregarranandasatia@gmail.com, [2)]munjiat.stth@gmail.com, [3)]nurwulanstth@gmail.com

Abstrak

Pada saat ini setiap institusi telah menerapkan sistem *login* pada aplikasi pengolahan data, temasuk Poltekkes Kemenkes Medan. Sistem *login* tidaklah cukup untuk menjaga keamanan data. Jika orang yang tidak bertanggung jawab berhasil mengaskes *database* pada *sever local* (*locahost*), maka orang tersebut telah berhasil mengetahui isi dari *database* tersebut, hal ini tentunya dapat merugikan pihak Poltekkes Kemenkes Medan, karena data yang terdapat pada database dapat disalahgunakan. Untuk itu perlu dilakukan peningkatan terhadap kemanan data pada *database*. Pada penelitian ini peningkatan terhadap keamanan *database* akan dilakukan dengan teknik enkripsi, enkripsi akan dilakukan dengan menggunakan algoritma *RC4* dan *Rail Fence*. *RC4* merupakan algoritma simetris, dimana proses enkripsi dan dekripsi menggunakan kunci yang sama, dan dilakukan pada tiap *bite* dengan operasi biner *xor*. Sementara *Rail Fence* merupakan algoritma simetris dengan teknik transposisi dengan pola zig-zag. Kombinasi dari kedua algoritma ini akan menghasilkan ciphertext yang sulit untuk dipecahkan sehingga menjadi lebih kuat.

Kata Kunci: *RC4, Rail Fence, Database*

Abstract

At this time every institution has implemented a system login in data processing applications, especially the Poltekkes Kemenkes Medan. The login system is not enough to maintain data security. If an irresponsible person successfully to enter the database on local server (locahost), then the person has succeeded in knowing the contents of the database, this can certainly be detrimental to the Poltekkes Kemenkes Medan, because the data contained in the database can be misused. Therefore it is necessary to increase the security of data in the database. In this research an increase database security will be done with encryption techniques, encryption will be done using the algorithm RC4 and Rail Fence. RC4 is a symmetric algorithm, in which the encryption and decryption process uses the same key, and is performed on each byte bybinary xor operations. While Rail Fence is a symmetrical algorithm with transposition techniques with zig-zag patterns. The combination of these two algorithms will produce ciphertext which is difficult to solve so that it becomes stronger.

Keywords: *RC4, Rail Fence, Database*

1. PENDAHULUAN

Komputer pada saat ini digunakan sebagai sarana untuk mempermudah manusia dalam melakukan pekerjaan terutama dalam melakukan pengelolaan data, data-data yang tersimpan pada komputer disebut sebagai *database* (basis data). Data-data tersebut dapat diolah untuk menghasilkan informasi.

Poltekkes Kemenkes (Politeknik Kesehatan Kementrian Kesehatan) Medan merupakan salah satu institusi pendidikan kesehatan yang menghasilkan banyak tenaga kesehatan. Ada banyak data mahasiswa yang tersimpan pada *database* kampus Poltekkes Kemenkes Medan. Kampus Poltekkes Kemenkes Medan menggunakan MYSQL sebagai *software* DBMS *(Database Management System)* untuk mengelola *database* yang ada pada kampus tersebut. Poltekkes Kemenkes Medan menerapkan *system login* pada aplikasi pengelolaan data yang digunakan untuk menjaga keamanan data mahasiswanya.

System login tidaklah cukup untuk menjaga keamanan data mahasiswa tersebut, jika orang yang tidak berkepentingan mengakses *localhost* secara langsung tanpa melewati *system login* pada aplikasi pengelolaan data untuk *database* mahasiswa Poltekkes Kemenkes Medan, maka orang tersebut telah berhasil melihat isi data mahasiswa yang tersimpan pada *localhost*. Hal ini dapat menyebabkan penyalahgunaan data seperti penggandaan terhadap data, tentunya hal tersebut dapat menyebabkan kesalahan terhadap informasi yang terkandung pada data-data tersebut dan dapat merugikan mahasiswa maupun kampus tersebut.

Untuk mencegah orang yang tidak bertanggung jawab melihat isi dari *database* maka perlu dilakukan peningkatan terhadap sistem keamanan *database*, dengan menyamarkan data pada *database* mahasiswa Poltekkes Kemenkes Medan. Penyamaran data pada *database* dapat dilakukan dengan menggunakan kriptografi.

ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

Kriptografi adalah ilmu mengenai teknik enkripsi, enkripsi sendiri adalah teknik yang digunakan untuk mengubah data kedalam bentuk kode-kode yang tidak dapat dimengerti lagi maknanya dengan menggunakan algoritma kriptografi, sehingga keamanan informasi dari data tersebut dapat terjaga dan tidak dapat dibaca tanpa didekripsi dahulu (kebalikan dari proses enkripsi).

Rumusan Masalah

Berdasarkan uraian dari latar belakang masalah diatas, adapun rumusan masalah yang diangkat dalam penelitian ini adalah :

1. Bagaimana merancang sistem keamanan untuk *database* mahasiswa kampus Poltekkes Kemenkes Medan ?
2. Bagaimana menerapkan algortima kriptografi RC4 dan *Rail Fence* untuk enkripsi data mahasiswa pada *database* Poltekkes Kemenkes Medan ?

Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Merancang dan membangun sebuah aplikasi yang dapat mengamankan data mahasiswa dengan kombinasi algoritma RC4 dan *Rail Fence*.
2. Dapat menerapkan aplikasi enkripsi dan dekripsi berbasis web dengan menggunakan algoritma RC4 dan *Rail Fence*.

2. METODE PENELITIAN

Penelitian ini dilakukan di Poltekkes Kemenkes Medan jurusan Keperawatan. Adapun metodologi pada penelitian ini adalah sebagai berikut :

1. Studi Literatur
   Pada tahap ini akan dilakukan pencarian data, informasi dan materi yang berhubungan dengan penelitian ini. Data, informasi dan materi didapat dari buku-buku serta artikel yang terkait dengan penelitian ini.
2. Observasi
   Pada tahap ini akan dilakukan pengamatan terhadap sistem kerja dan pengumpulan data-data agar mendapatkan informasi yang dibutuhkan untuk penelitian ini.
3. Analisis Metode
   Pada tahap ini akan dilakukan analisis terhadap metode yang digunakan seperti menghitung rumus yang terdapat pada algoritma yang digunakan pada penelitian ini.
4. Implemetasi Metode
   Pada tahap ini akan dilakukan penerapan algoritma RC4 dan *Rail Fence* pada bahasa pemrograman PHP (*Hypertext Preproccessor*) sebagai salahsatu bahasa pemrograman WEB dan MySQL sebagai bahasa pemrograman *database*.
5. Pengujian
   Pada tahap ini akan dilakukan pengujian penerapan algoritma RC4 dan *Rail Fence* menggunakan PHP dan MySQL sesuai pernyataan, dan mengarahkan pengujian untuk mendapatkan hasil yang dibutuhkan.

3. DASAR TEORI

3.2 Algoritma Kriptografi

Definisi terminologi algoritma adalah urutan langkah-langkah logis untuk menyelesaikan masalah yang disusun secara sistematis. Algoritma keriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut[1].

Algoritma kriptografi dibagi menjadi tiga bagian berdasarkan kunci yang dipakainya[1]:

1) Algoritma simetris
   Algoritma ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsi. Algortima ini sudah ada sejak lebih dari 4000 tahun yang lalu. Bila mengirim pesan dengan menggunakan algoritma ini, si penerima pesan harus diberi tahu kunci dari pesan tersebut agar bisa mendekripsikan pesan yang dikirim. Keamanan dari pesan yang menggunakan algoritma ini tergantung pada kunci. Jika kunci tersebut diketahui oleh orang lain maka orang tersebut akan dapat melakukan enkripsi dan dekripsi terhadap pesan.
2) Algoritma Asimetris.
   Algoritma asimetri sering juga disebut dengan algoritma kunci public, dengan arti kata kunci yang digunakan untuk melakukan enkripsi dan dekripsi berbeda. Pada algoritma asimetri kunci terbagi menjadi dua bagian, yaitu[1] :
   a. Kunci umum *(Public Key):* Kunci yang boleh semua orang tahu (dipublikasikan).
   b. Kunci rahasia *(Private Key):* Kunci yang dirahasiakan (hanya boleh diketahui oleh satu orang).

   Kunci-kunci tersebut berhubungan satu sama lain. Dengan kunci publik orang dapat menenkrispi pesan tetapi tidak dapat mendekripsinya. Hanya orang yang memiliki kunci rahasia yang dapat mendekripsi pesan tersebut. Algoritma asimetri bisa mengirimkan pesan lebih aman daripada algoritma simetri.

3) Fungsi *Hash*
   Fungsi *Hash* sering disebut dengan fungsi hash satu arah (*one-way function*), *message digest*, *fingerprint*, fungsi kompresi dan *message authentication code* (MAC). Merupakan suatu fungsi matematika yang mengambil masukkan

ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap. Fungsi Hash biasanya diperlukan bila ingin membuat sidik jari dari suatu pesan. Sidik jari pada pesan merupakan suatu tanda bahwa pesan tersebut benar-benar berasal dari orang yang diinginkan[1].

3.3 Algoritma RC4 (Rivest Code 4)

Algoritma *RC4* (*Riverst Code 4*) merupakan *stream cipher* yang dirancang di *RSA (Rivest Shamir Adleman) Security* oleh *Ron Rivest* tahun 1987. Sifat kunci dalam algoritma *RC4* adalah simetris serta melakukan proses enkripsi *plain* per *digi*t atau *byte* per *byte* dengan operasi biner (biasanya XOR) dengan sebuah angka semiacak[3].

Sampai saat ini kode RC4 belum bisa dipecahkan kecuali menggunakan *exhaustive key search* atau *brute force search* yang merupakan suatu teknik dasar yang digunakan kriptanalis untuk mencoba setiap kunci yang mungkin hingga ditemukan kunci yang sebenarnya. Batas kunci RC4 yang bisa dipecahkan adalah 40 *bit*. Kini pemerintah AS sudah mengubah ketetapannya mengenai ekspor/penggunaa algoritma kriptografi yang lebih besar dari 40 *bit* diluar negeri. Batasnya adalah 128 *bit*[1].

3.4 Algoritma RC4 (Rivest Code 4)

Algoritma *RC4* (*Riverst Code 4*) merupakan *stream cipher* yang dirancang di *RSA (Rivest Shamir Adleman) Security* oleh *Ron Rivest* tahun 1987. Sifat kunci dalam algoritma *RC4* adalah simetris serta melakukan proses enkripsi *plain* per *digi*t atau *byte* per *byte* dengan operasi biner (biasanya XOR) dengan sebuah angka semiacak[3].

Sampai saat ini kode RC4 belum bisa dipecahkan kecuali menggunakan *exhaustive key search* atau *brute force search* yang merupakan suatu teknik dasar yang digunakan kriptanalis untuk mencoba setiap kunci yang mungkin hingga ditemukan kunci yang sebenarnya. Batas kunci RC4 yang bisa dipecahkan adalah 40 *bit*. Kini pemerintah AS sudah mengubah ketetapannya mengenai ekspor/penggunaa algoritma kriptografi yang lebih besar dari 40 *bit* diluar negeri. Batasnya adalah 128 *bit*[1].

Algoritma RC4 cukup mudah untuk dijelaskan RC4 mempunyai sebuah *S-Box*, S0, S1,…S255, yang berisi permutasi dari bilangan 0 sampai 255, dan permutasi merupakan fungsi dari kunci dengan panjang yang variabel. Terdapat dua indeks yaitu i dan j, yang diinisialisasi dengan bilangan nol[4]. Menurut Setianingsih dan Agung H. & Budiman dalam[3] Algortima RC4 bekerja dengan tiga tahap utama yaitu *Key Scheduling Algorithm* (KSA), *Pseudo Random Generation Algorithm* (PRGA) dan proses enkripsi dan dekripsi.

1) *Key Scheduling Algorithm (*KSA*)*
   Proses KSA merupakan proses pembentukan tabel *S-Box* (Tabel *Array* S) dan Kunci (Tabel *array* T) yang di permutasi sebanyak 256 iterasi[3].
   Pada tahap ini ada tiga proses yaitu :

a. Inisialisasi *array S-box* pertama, S[0], S[1],.....,S[255]. Diisi dengan bilangan 0 sampai 255, sehingga array S-box array S berbentuk S[0] = 0, S[1] = 1,...., S[255] = 255 [5].
b. Inisialisasi *array* kunci (*S-box* lain), misal *array* kunci K dengan panjang 256. Jika panjang kunci K < 256, maka di lakukan *padding* yaitu penambahan *byte* sehingga panjang kunci menjadi 256 *byte*. Misalnya K = "abc" yang hanya terdiri dari tiga *byte* (tiga huruf), maka lakukan *padding* dengan penambahan *byte* (huruf) semu, misalnya K = "abcabcabcabc....." sampai panjang K mencapai 256 *byte*, sehingga *S-Box array* kunci K terbentuk. Adapun cara pembentukan tabel K yaitu, i diberi nilai awal 0 maka lakukan perhitungan sebagai berikut[5] :
   $K(i) = Kunci\ (i\ mod\ keylength)$
c. Setelah dua proses diatas dilakukan, lalu acak tabel S. "Maka Set indeks j dan i dengan nol, kemudian lakukan langkah berikut"[4] :
   $j = (j + Si + Ki)\ mod\ 256$
   $Tukarkan\ nilai\ (Si\ dan\ Sj)$

2) *Pseudo Random Generation Algorithm* (PRGA)
   Tabel *array S-Box* akan digunakan pada proses ini untuk menghasilkan *key stream* yang jumlahnya sama dengan jumlah banyaknya karakter *plaintext* kemudian akan di XOR dengan *plaintext*[3]. Terdapat dua indeks yaitu i dan j, yang diinisialisasi dengan bilangan nol. Untuk menghasilkan random *byte* langkahnya adalah sebagai berikut Slamet Maryono dalam[4] :
   $i = (i + 1)\ mod\ 256$
   $j = (j + Si)\ mod\ 256$
   $Tukarkan\ nilai\ (Si\ dan\ Sj)$
   $t = (Si + Sj)\ mod\ 256$
   $k = St$
3) Proses enkripsi atau dekripsi dengan operasi *XOR*. Proses enkripsi atau dekripsi di awali dengan merubah setiap nilai *plaintext* ke biner *Formula* untuk melakukan proses enkripsi dan dekripsi adalah[3] :
   Enkripsi : $Ci = Pi\ XOR\ Ki$ (1)
   Dekripsi : $Pi = Ci\ XOR\ Ki$ (2).

3.5 Algoritma Rail Fence (Zig-Zag) Cipher

ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

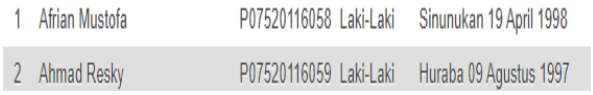
Metode zig-zag *cipher* merupakan salah satu algoritma kriptografi klasik dengan teknik transposisi. Metode transposisi adalah metode yang enkripsi dengan menyusun *plaintext* pada matriks secara baris, lalu dari hasil susunan tersebut menghasilkan sebuah *ciphertext* dengan mengambil rangkaian karakter secara kolom[2].

Format matrix yang membentuk baris dan kolom dapat diterapkan menjadi pola kerja dari zig-zag dalam melakukan transposisi teks asli[6].

4. HASIL DAN PEMBAHASAN

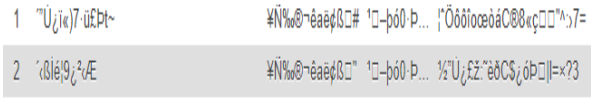
4.1 Hasil

Tamplian data pada *database* sebelum dienkripsi menggunakan algoritma RC4 dan *Rail Fence.* Dapat dilihat pada gambar 1 data yang belum melalui tahap enkripsi masi memiliki makna (informasi), jika data tidak dienkripsi data dapat saja dicuri oleh orang yang tidak bertanggung jawab maka dari itu data perlu melalui tahap enkripsi untuk menyembunyikan maknanya.



Gambar 1. Data Sebelum Dienkripsi

Tampilan data pada *database* setelah dienkripsi menggunakan algoritma RC4 dan *Rail Fence.* Pada gambar 2 dapat dilihat hasil enkripsi dari data pada gambar 1, data-data tersebut sudah berubah menjadi simbol-simbol yang tidak memiliki makna lagi.



Gambar 2. Data Setelah Dienkripsi

4.2 Pembahasan

Proses enkripsi data mahasiswa akan dirancang dengan algoritma RC4 dan *Rail Fence*. Pertama proses enkripsi data akan dilakukan oleh algoritma RC4, hasil *ciphertext* proses algoritma RC4 akan dienkripsi kembali dengan menggunakan algoritma *Rail Fence*. Kemudian hasil dari enkripsi dari *Rail Fence* akan disimpan kembali kedalam *database* dan akan ditampilkan ke aplikasi, setelah *user* melakukan enkripsi *user* tidak dapat melakukan pengolahan data.

Pada proses enkripsi ini *Plaintext* yang akan dienkripsi adalah “Afrian Mustofa”, dan kunci yang digunakan adalah “Bangko Sempurna”.

Adapun proses enkripsi dengan menggunakan algoritma RC4 yaitu melalui tiga tahap, adapun tahapnya yaitu sebagai berikut :

1) KSA*(Key Scheduling Algorithm)*

KSA merupakan tahap pembentukan tabel S dan tabel K. Pada tabel S berisikan biliangan 0 sampai 255, dan tabel K nantinya akan diisi dengan desimal dari kunci.

a. Sebelum melakukan inisialisasi terhadap tabel K, karakter kunci diubah kedalam bentuk desimal. Pada saat inisialisasi tabel K akan dilakukan penambahan panjang *byte* pada kunci *(padding)* jika panjang kunci < 256, hasil padding nantinya akan menghasilkan tabel K, *flowchart* inisialisasi tabel K dapat dilihat pada tabel 1.

Tabel 1. Tabel Desimal Kunci

Karakter	B	a	n	g	k	o	space	S	e	m	p	u	r	n	a
Desimal	66	97	110	103	107	111	32	83	101	109	112	117	114	110	97

Setelah Kunci diubah ke desimal maka kunci akan dilakukan penambahan panjang *byte* (inisialisasi terhadap tabel K) sehingga panjang kunci 256

Tabel 2. Hasi Inisialisasi Tabel K

66	97	110	103	107	111	32	83	101	109	112	117	114	110	97	66
97	110	103	107	111	32	83	101	109	112	117	114	110	97	66	97
110	103	107	111	32	83	101	109	112	117	114	110	97	66	97	110
103	107	111	32	83	101	109	112	117	114	110	97	66	97	110	103
107	111	32	83	101	109	112	117	114	110	97	66	97	110	103	107
111	32	83	101	109	112	117	114	110	97	66	97	110	103	107	111
32	83	101	109	112	117	114	110	97	66	97	110	103	107	111	32
83	101	109	112	117	114	110	97	66	97	110	103	107	111	32	83
101	109	112	117	114	110	97	66	97	110	103	107	111	32	83	101
109	112	117	114	110	97	66	97	110	103	107	111	32	83	101	109
112	117	114	110	97	66	97	110	103	107	111	32	83	101	109	112
117	114	110	97	66	97	110	103	107	111	32	83	101	109	112	117
114	110	97	66	97	110	103	107	111	32	83	101	109	112	117	114
110	97	66	97	110	103	107	111	32	83	101	109	112	117	114	110
97	66	97	110	103	107	111	32	83	101	109	112	117	114	110	97
66	97	110	103	107	111	32	83	101	109	112	117	114	110	97	66

b. Tabel S yang sudah diacak.

Tabel 3. Tabel S Yang Sudah Diacak Pada Tahap KSA

2) PRGA *(Peseudo Random Genetration Algorithm)*

Pada tahap ini tabel S yang sudah proses pada tahap KSA akan diacak kembali pada tahap ini. Iterasi disesuaikan dengan panjang *plaintext* nya (jumlah *keystream* yang dihasilkan pada proses ini sama dengan jumlah karakter *Plaintext*).

Tabel 4. Tabel S Yang Sudah Diacak Pada Tahap PRGA

ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

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

Keterangan tabel 4 :
Abu-abu = indeks tabel S.
Hijau = isi dari tabel S yang diacak.
Biru = isi tabel S yang digunakan sebagai *keystream.*
Dari tahap PRGA diatas maka didapatlah *keystream* yaitu, 245, 81, 230, 222, 187, 146, 159, 238, 154, 173, 223, 27, 79, dan 31.
Pada tahap ini tabel S yang sudah proses pada tahap KSA akan diacak kembali pada tahap ini. Iterasi disesuaikan dengan panjang *plaintext* nya (jumlah *keystream* yang dihasilkan pada proses ini sama dengan jumlah karakter *Plaintext*). Pada proses ini i dan j diberikan nilai awal 0.

Untuk mendapatkan nilai *keystream* maka akan dilakukan penjumlahan terhadap Si dan Sj. Maka didapatlah *Keystream* yang akan di xor kan untuk mendapatkan *ciphertext*.

3) Tahap Xor
Pada tahap ini akan di xorkan biner dari *plaintext* dan *Keystream*, tahap ini merupakan tahap terakhir pada algoritma RC4 untuk mendapatkan *ciphertext.* Tahap ini juga digunakan pada proses dekripsi untuk mendapatkan kembali *plaintext.*

Tabel 5. Hasil Xor *Plaintext* Dan *Keystream*

Keystream	Xor	Plaintext	Hasil	Ciphertext
245 = 11110101	⊕	65 = 01000001	10110100	´
81 = 01010001	⊕	102 = 01100110	110111	7
230 = 11100110	⊕	114 = 01110010	10010100	”
222 = 11011110	⊕	105 = 01101001	10110111	·
187 = 10111011	⊕	97 = 01100001	11011010	Ú
146 = 10010010	⊕	110 = 01101110	11111100	ü
159 = 10011111	⊕	32 = 00100000	10111111	¿
238 = 11101110	⊕	77 = 01001101	10100011	£
154 = 10011010	⊕	117 = 01110101	11101111	ï
173 = 10101101	⊕	115 = 01110011	11011110	Þ
223 = 11011111	⊕	116 = 01110100	10101011	«
27 = 00011011	⊕	111 = 01101111	1110100	t
79 = 01001111	⊕	102 = 01100110	101001	)
31 = 00011111	⊕	97 = 01100001	1111110	~

Setelah melalui proses enkripsi dengan RC4 maka selanjutnya enkripsi akan dilanjutkan oleh *Rail Fence* berikut adalah proses enkripsi menggunakan *Rail Fence* :

1) Karakter teks yang akan dienkripsi (**´7”·Úü¿£ïÞ«t)~**). Proses enkripsi pada algoritma ini menggunakan kunci k = 2, offset = 0 dimana proses enkripsi menggunakan tabel dengan 2 buah baris dan dimulai dari baris ke 0 atau baris paling atas. Jumlah karakter teks yang akan dienkripsi adalah 14, maka masukkan karakter degan pola zig-zag dimulai dari baris yang paling atas (baris 0) seperti pada tabel 5 :

Tabel 6. Tabel Enkripsi Zig-Zag Dengan K = 2 Dan Offset = 0

Baris 0	´		”		Ú		¿		ï		«		)	
Baris 1		7		·		Ü		£		Þ		t		~

Mengeluarkan setiap karakter dari tabel untuk mendapatkan *ciphertext.* Keluarkan setiap karakter

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

pada baris 0 (**´”Úü¿ï«)**) dan keluarkan seluruh karakter yang terdapat pada baris 1 (**7·Ü£Þt~**), letakan seluruh

ISSN 2338-5677 Cetak
ISSN 2549-6646 Online

karakter yang terdapat pada baris 0 didepan dan seluruh karakter yang terdapat pada baris 1 dibelakang baris 0, maka didapatlah ciphertext dari algoritma *Rail Fence* yaitu (**´”Ú¿ï«)7·Ü£Þt~**).

5. PENUTUP

5.1 Simpulan

Berdasarkan dari hasil implementasi dan pengujiana sistem maka penulis dapat mengambil kesimpulan sebagai berikut :

1. Aplikasi yang dirancang pada penelitian ini dapat melakukan enkripsi dan dekripsi dengan baik.
2. Algoritma RC4 dan *Rail Fence* dapat digunakan untuk mengamankan database mahasiswa Kampus Poltekkes Kemenkes Medan dengan tingkat keamanan yang memadai.
3. Algoritma RC4 dan *Rail Fence* dapat di terapkan pada perograman Web sebagai aplikasi enkripsi dan dekripsi.

5.2 Saran

Adapun saran dari hasil penelitian ini agar dapat dikembangkan dengan lebih baik lagi, yaitu :

1. Diharapkan pada pengembang selanjutnya, agar dapat menambah algoritma kriptografi modern asimetris pada aplikasi untuk meningkatkan keamanan data, terutama data teks.
2. Diharapkan pada pengembang selanjutnya, agar dapat meningkatkan fitur pada aplikasi.
3. Pada aplikasi ini, sistem enkripsinya masih dilakukan secara manual yaitu admin harus melakukan enkripsinya dengan memasukkan kunci pada halaman enkripsi, untuk kedepanya diharapkan aplikasi ini dapat dibangun dengan sistem enkripsi otomatis, tanpa harus memasukkan kunci.

6. DAFTAR PUSTAKA

[1] D. Ariyus, *Pengantar Ilmu Kriptografi: Teori Analisis dan Implementasi*. Yogyakarta: ANDI OFFSET, 2008.

[2] R. K. Hondro, “Aplikasi Enkripsi Dan Dekripsi SMS Dengan Algoritma Zig Zag Cipher Pada Mobile Phone Berbasis Android,” *Pelita Inform. Budi Darma*, vol. 10, no. 3, 2015.

[3] T. Zebua and E. Ndruru, “Pengamanan Citra DIgital Berdasarkan Modifikasi Algoritma RC4,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 4, no. 4, pp. 275–282, 2017.

[4] H. Pandiangan and S. Sijabat, “Perancangan Media Pengiriman Pesan Teks Dengan Penyandian Pesan Menggunakan Algoritma RC4 Berbasis WEB,” *J. Matik Penusa*, vol. 19, no. 1, pp. 63–71, 2016.

[5] Nurhardian and A. Pudoli, “Implementasi Keamanan File dengan Kompresi Huffman dan Kriptografi menggunakan Algoritma RC4 serta Steganografi menggunakan End of File Berbasis Desktop pada SMK Negeri 3 Kota Tangerang,” *J. TICOM*, vol. 5, no. 1, pp. 39–46, 2016.

[6] A. Hariati, K. Hardiyanti, and W. E. Putri, “Kombinasi Algoritma Playfair Cipher Dengan Metode Zig-zag Dalam Penyandian Teks,” *J. Penelit. Tek. Inform.*, vol. 2, no. 2, pp. 13–17, 2018.