

Implementasi Keamanan File Dokumen Menggunakan Metode RC4 Dan BASE64

Implementation of Document File Security Using the RC4 and BASE64 Methods

Ananda Perdana^{a,1}, Wahyu Saptha Negoro^{b,2}

^aMahasiswa Universitas Potensi Utama, Jl.K.L Yos Sudarso KM 6.5 Tj.Mulia, Medan 20241, Indonesia

^bDosen Universitas Potensi Utama, Jl.K.L Yos Sudarso KM 6.5 Tj.Mulia, Medan 20241, Indonesia

nandakeleng101@gmail.com¹, wahyusaptha1707@gmail.com²

ABSTRAK

Penelitian ini berfokus pada penerapan keamanan dokumen untuk PT Agri Marina Inspeksi dengan menggunakan algoritma kriptografi RC4 dan Base64 dalam aplikasi berbasis Android untuk melindungi data sensitif karyawan. Studi ini menyoroti kebutuhan peningkatan keamanan dokumen, mengingat file penting perusahaan saat ini disimpan dalam format .docx tanpa perlindungan memadai, sehingga rentan terhadap kebocoran data, modifikasi, dan akses yang tidak sah. Metodologi penelitian mengadopsi pendekatan Waterfall, mencakup tahap analisis kebutuhan, desain sistem, pengembangan kode, pengujian aplikasi, hingga evaluasi akhir. Data dikumpulkan melalui observasi lapangan langsung dan tinjauan literatur untuk mendukung desain sistem. Algoritma RC4, sebagai metode enkripsi simetris, secara efektif mengenkripsi data, sementara Base64 melakukan encoding tambahan untuk mengamankan data dalam format yang dapat disimpan dan ditransmisikan dengan andal. Temuan menunjukkan bahwa penggunaan RC4 dan Base64 secara bersamaan secara signifikan meningkatkan keamanan dokumen, memungkinkan akses hanya bagi pihak yang berwenang. Kesimpulannya, pendekatan ini memberikan solusi praktis dan efektif untuk melindungi informasi sensitif di PT Agri Marina Inspeksi, mengurangi risiko kebocoran data dan memastikan integritas data.

Kata Kunci : Aplikasi Android, Base64, Keamanan Dokumen, Kriptografi, RC4

ABSTRACT

This research focuses on implementing document security for PT Agri Marina Inspeksi using the RC4 and Base64 cryptographic algorithms in an Android-based application to safeguard sensitive employee data. The study addresses the need for increased document security, as critical company files are currently stored in .docx format without sufficient protection, making them susceptible to data leakage, tampering, and unauthorized access. The research methodology adopts a Waterfall approach, encompassing stages from requirements analysis, system design, code development, and application testing to final evaluation. Data was gathered through direct field observations and a literature review to support the system's design. The RC4 algorithm, a symmetric encryption method, effectively encrypts the data, while Base64 encoding further secures the data in a format that can be reliably stored and transmitted. The findings indicate that using RC4 and Base64 together significantly enhances document security, allowing only authorized access. In conclusion, this approach provides a practical and effective solution to protect sensitive information at PT Agri Marina Inspeksi, reducing the risk of data breaches and ensuring data integrity.

Keywords : Android Application, Base64, Cryptography, Document Security, RC4

Info Artikel :

Disubmit: 05 Agustus 2024

Direview: 11 September 2024

Diterima : 19 Oktober 2024

Copyright © 2024 – senadimu. All rights reserved.

1. PENDAHULUAN

Pesatnya perkembangan teknologi informasi dan juga komunikasi pada masa sekarang telah menunjukkan dampaknya bagi semua kalangan terutama dalam perihal memudahkan berbagai pekerjaan manusia seperti penggunaan dokumen digital yang memudahkan dalam mengelola data dan informasi,

penggunaan jaringan internet yang memudahkan komunikasi ataupun pertukaran informasi dari jarak yang sangat jauh dan masih banyak lagi. Perkembangan tersebut di sisi lain juga menyebabkan munculnya ancaman kejahatan komputer atau kejahatan dunia maya yaitu kebocoran informasi, modifikasi, penyadapan hingga penyalahgunaan informasi oleh pihak yang tidak memiliki otoritas atas suatu data atau informasi yang bisa merugikan siapapun sehingga membuat aspek keamanan menjadi faktor penting dan lebih diperhatikan. Proses keamanan data pegawai diterapkan dengan teknik kriptografi [1].

Kriptografi merupakan teknik pengamanan informasi yang dilakukan dengan cara mengolah informasi awal (plaintext) dengan suatu kunci tertentu menggunakan suatu metode enkripsi tertentu sehingga menghasilkan suatu informasi baru (ciphertext) yang tidak dapat dibaca secara langsung. Enkripsi juga merupakan bagian dari keamanan informasi, yaitu kerahasiaan adalah layanan yang dirancang untuk melindungi konten informasi dari semua kecuali mereka yang memiliki izin atau kunci rahasia untuk membuka/mengekstraksi informasi yang dienkripsi.

Algoritma RC4 adalah salah satu algoritma kunci simetris berupa enkripsi aliran yang memproses unit atau input data dalam satu waktu. Unit atau data biasanya berupa byte, atau terkadang bit, dan dapat memiliki panjang variabel untuk enkripsi atau dekripsi [2]. Sedangkan Algoritma Base64 adalah salah satu algoritma untuk Encoding dan Decoding suatu data ke dalam format ASCII, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metoda yang digunakan untuk melakukan encoding (penyandian) terhadap data binary yang diterapkan dengan aplikasi berbasis android [3].

Enkripsi adalah sebuah proses mengubah plaintext menjadi Ciperteks yaitu kode kode yang tidak dapat dimengerti. Metode Enkripsi yang digunakan pada penelitian ini adalah menggunakan Algoritma RC4 dan Base64 seperti yang sudah dijelaskan pada paragraf sebelumnya diatas. Sedangkan Dekripsi adalah proses mengembalikan Ciperteks menjadi plainteks[4]. Metode Dekripsi yang digunakan adalah menggunakan metode gabungan RC4 dan Base64 seperti yang sudah dijelaskan pada paragraph sebelumnya diatas.

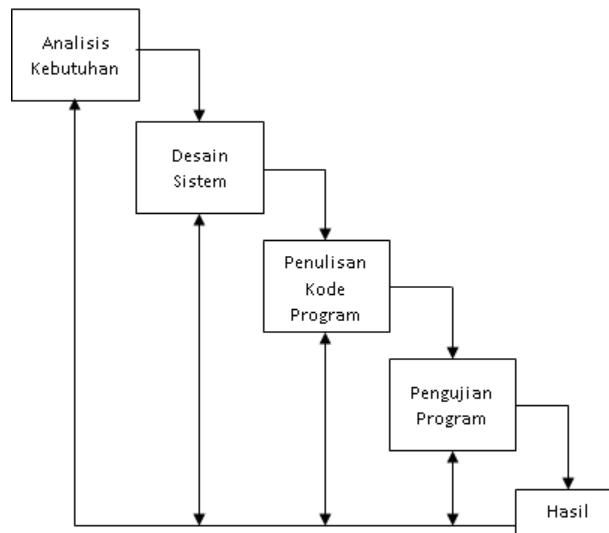
Android merupakan sebuah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi, middleware, dan aplikasi [5]. Android adalah software platform yang open source untuk mobile device. Android berisi sistem operasi, middleware dan aplikasi-aplikasi dasar. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi. Aplikasi berbasis android dapat menyimpan dokumen perusahaan [6]. Dokumen adalah informasi rahasia yang sangat berharga dan penting bagi otoritas individu atau perusahaan manapun dan hanya boleh diketahui oleh pihak berkepentingan saja Keamanan informasi merupakan topik yang luas, termasuk keamanan komputer dari serangan hacking. Masalah keamanan adalah bagian penting dari sistem informasi. Salah satu hal terpenting dalam komunikasi komputer dan jaringan komputer untuk menjamin keamanan pesan, data atau informasi apapun [7].

PT Agri Marina Inspeksi adalah perusahaan yang bergerak di bidang surveyor yang mencakup agrikultur, kelautan, pemeriksaan tangki kapal, pemeriksaan kelayakan kargo, sertifikasi pengapalan, dan konsultasi. Perusahaan ini memiliki informasi yang harus dijaga keamanannya. Dalam kegiatan yang terjadi pada PT Agri Marina Inspeksi, informasi masih disimpan dalam bentuk file dokumen yang berekstensi .docx. Saat ini PT Agri Marina Inspeksi masih menggunakan Microsoft Word untuk mengolah data yang kemudian disimpan di harddisk, dengan demikian dapat menimbulkan kehilangan data dan dapat menimbulkan pencurian data oleh pihak yang tidak berwenang. PT Agri Marina Inspeksi memiliki dokumen yang bersifat penting serta rahasia, yang tidak boleh diketahui oleh pihak yang tidak berkepentingan seperti dokumen data pegawai. Perusahaan ini bergerak di bidang surveyor yang tenaga kerjanya masih sedikit ditemukan sehingga diperlukan upaya untuk menjaga identitas pegawai sehingga perusahaan pesaing tidak dapat mengetahui pegawai yang bekerja di PT. Agri Marina Inspeksi. Permasalahan yang dihadapi adalah file dokumen dapat diketahui dan dimengerti oleh pihak yang tidak berkepentingan dikarenakan perusahaan masih belum memiliki sistem pengamanan untuk file dokumen. Upaya untuk mengatasi permasalahan yang ada yaitu mengimplementasikan mekanisme keamanan seperti algoritma kriptografi yang bisa menyandikan isi file dokumen sehingga membuat file dokumen menjadi tidak bisa dipahami dan tidak bisa diketahui oleh pihak yang tidak berkepentingan.

2. METODE

A. Metodologi Penelitian

Penelitian ini menggunakan metode studi literatur untuk merangkum dan menganalisa sumber-sumber terpercaya dan valid tentang penggunaan kriptografi menggunakan kombinasi algoritma RC4 dan Base64 untuk mengenkripsi dan mendekripsi data pegawai. Penelitian ini akan melalui beberapa tahapan. Tahapan dalam penelitian ini dapat dimodelkan pada diagram waterfall. Adapun beberapa tahapan yang digunakan dalam penelitian ini adalah sebagai berikut :



Gambar 1. Diagram Waterfall

Keterangan :

1. Analisis Kebutuhan

Pada tahapan ini peneliti mengumpulkan kebutuhan sistem yang diperlukan dalam proses implementasi keamanan file dokumen data pegawai. Serta hal-hal yang dibutuhkan untuk perancangan sistem dengan menggunakan pemrograman android yang nantinya dapat dijalankan dalam mengamankan data dengan menggunakan gabungan algoritma RC4 dan Base64.

2. Desain Sistem

Desain sistem yang digunakan dalam teori adalah pemodelan UML yaitu *use case diagram*, *class diagram*, *activity diagram*, dan *sequence diagram*.

3. Penulisan Kode Aplikasi

Dalam penulisan kode aplikasi, peneliti menggunakan Bahasa pemrograman java

4. Pengujian

Pengujian aplikasi dilakukan untuk mengetahui kekurangan sistem. Apabila terdapat kekurangan sistem atau program tidak berjalan dengan baik, maka akan dilakukan perbaikan sampai seluruh program berjalan dengan baik.

5. Hasil

Pada tahapan akhir penelitian ini memastikan bahwasanya program atau aplikasi sudah sesuai dengan perancangan. Kemudian pada program atau aplikasi yang sudah selesai tetap dilakukan pemeliharaan disesuaikan dengan kebutuhan pengguna.

B. Landasan Teori

- Base64

Transformasi Base64 merupakan salah satu algoritma untuk Encoding dan Decoding suatu data ke dalam format ASCII, yang didasarkan pada bilangan dasar 64 atau bisa dikatakan sebagai salah satu metoda yang digunakan untuk melakukan encoding (penyandian) terhadap data binary. Cara Kerja Algoritma Base 64 adalah Carilah kode ASCII dari masing-masing teks lalu cari bilangan biner 8 bit dari kode ASCII yang ada dan gabungkan 8 bit tadi menjadi 24 bit.

Kemudian, pecah 24 bit tadi menjadi 6 bit maka akan menghasilkan 4 pecahan masing-masing pecahan diubah ke dalam nilai decimal terakhir, jadikan nilai – nilai decimal yang akan menjadi indeks untuk memilih karakter penyusun daripada base64 dan juga maksimal adalah 63 atau indeks ke 64 ternyata didalam proses encoding juga terdapat sisa pembagi, maka tambahkan menjadi penggenap sisa karakter tersebut ‘=’. Maka sering kali pada base64 akan muncul satu atau dua karakter ‘=’ [8].

- RC4

RC4 merupakan jenis dari aliran kode yang yang berarti operasi enkripsinya dilakukan per karakter 1 byte untuk sekali operasi. Secara resmi RC4 tidak pernah dipublikasikan, tetapi pada bulan september tahun 1994 seseorang yang tidak dikenal mengirim detail algoritmanya ke internet sehingga di ketahui publik. RC4 memiliki panjang kunci 2048 bit (256 byte) [9].

- Android

Android adalah software platform yang open source untuk mobile device. Android berisi sistem operasi, middleware dan aplikasi-aplikasi dasar. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi. Basis OS Android adalah kernel linux 2.6 yang telah dimodifikasi untuk mobile device. Android versi 1.0 dikeluarkan tanggal 23 September 2008, versi 1.1 pertama kali digunakan di mobile phone (T1) [6].

- Firebase

Firebase adalah BaaS (Backend as a Service) yang saat ini dimiliki oleh Google. Firebase pertama kali didirikan pada tahun 2011 oleh Andrew Lee dan James Tamplin. Produk yang pertama kali dikembangkan adalah Realtime Database, dimana developer dapat menyimpan dan melakukan sinkronasi data ke banyak user. Kemudian berkembang menjadi layanan penyedia pengembangan aplikasi. Pada Oktober 2014, perusahaan tersebut diakusisi oleh Google [10].

3. HASIL DAN PEMBAHASAN

A. Penerapan Metode

Algoritma RC4 menggunakan dua buah S-Box array dengan panjang 256 dan berisi permutasi dari bilangan 0 hingga 255, dan S- Box kedua, yang berisi hasil permutasi merupakan fungsi dari kunci dengan panjang yang bervariasi. Cara kerja algoritma RC4 yaitu inialisasi S-Box pertama, S[0], S[1], S[2],..., S[255], dengan bilangan 0 hingga 255. Pertama S-Box diisi secara berurutan dengan S[0] = 0, S[1] = 1, S[2] = 2,...,S[255] = 255. Kemudian menginisialisasi array kedua (S-Box kedua), misalkan array K dengan panjang 256. Isi array K dengan kunci yang diulangi sampai seluruh array K[0], K[1],...,K[255] telah terisi seluruhnya [11].

1) Proses inialisasi S-Box (Array S) inialisasi dengan persamaan berikut:

for i = 0 to 255

S[i] = i

2) Selanjutnya proses inialisasi S-Box (Array K) dengan persamaan berikut:

Array kunci dengan panjang kunci “length” for i = 0 to 255

$K[i] = \text{Kunci}[i \bmod \text{length}]$

3) Kemudian langkah pengacakan S-Box dirumuskan dengan persamaan berikut:

$i = 0 ; j = 0$

for $i = 0$ to 255

$j = (j + S[i] + K[i]) \bmod 256$

swap $S[i]$ dan $S[j]$

4) Lalu membuat pseudo random code dengan rumus persamaan berikut:

$i = (i + 1) \bmod 256$

$j = (j + S[i]) \bmod 256$

swap $S[i]$ dan $S[j]$ t

$t = (S[i] + S[j]) \bmod 256$

$K = S[t]$

B. Studi Kasus

Berikut ini adalah contoh perhitungan dari implementasi metode RC4 dan Base64 dalam konteks pengamanan file dokumen. Dalam contoh ini akan berfokus pada bagaimana pengamanan dengan mengambil sampel dari nama pegawai yang di enkripsi menggunakan RC4 dan Base64 dan akan di dekripsi kembali dengan gabungan metode tersebut untuk membuktikan bahwa metode tersebut dapat mengamankan data, berikut gambaran Langkah-langkah yang diambil dan prosesnya :

- Enkripsi RC4 dan Base64

Penyelesaian penerapan metode enkripsi Rc4 dan Base64, dalam contoh hal studi kasus ini kita mengambil sampel adalah kata nama pegawai untuk diterapkan dalam metode enkripsi dan dekripsi Rc4.

Plaintext : Sazqya

Kunci : 12345

- 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 1. 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 "12345" jadi kalimat yang akan diubah ke dalam bentuk desimal dan diulangi kunci sampai memenuhi seluruh array K sehingga array K menjadi:

Tabel 2. Inisialisasi Array K

49	50	51	52	53	49	50	51	52	53	49	50	51	52	53	49
50	51	52	53	49	50	51	52	53	49	50	51	52	53	49	50
51	52	53	49	50	51	52	53	49	50	51	52	53	49	50	51
52	53	49	50	51	52	53	49	50	51	52	53	49	50	51	52
53	49	50	51	52	53	49	50	51	52	53	49	50	51	52	53
49	50	51	52	53	49	50	51	52	53	49	50	51	52	53	49
50	51	52	53	49	50	51	52	53	49	50	51	52	53	49	50
51	52	53	49	50	51	52	53	49	50	51	52	53	49	50	51
52	53	49	50	51	52	53	49	50	51	52	53	49	50	51	52
53	49	50	51	52	53	49	50	51	52	53	49	50	51	52	53
49	50	51	52	53	49	50	51	52	53	49	50	51	52	53	49
50	51	52	53	49	50	51	52	53	49	50	51	52	53	49	50
51	52	53	49	50	51	52	53	49	50	51	52	53	49	50	51
52	53	49	50	51	52	53	49	50	51	52	53	49	50	51	52
53	49	50	51	52	53	49	50	51	52	53	49	50	51	52	53
49	50	51	52	53	49	50	51	52	53	49	50	51	52	53	49

Berdasarkan Tabel III.2 nilai desimal dari Kunci 12345 adalah 49, 50, 51, 52, 53 yang kemudian diulangi sampai memuni array K yaitu 255.

- 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 $(j+S[i]+K[i] \bmod 256)$ yang diikuti dengan pertukaran S[i] dengan S[j]. Untuk contoh ini, karena menggunakan array dengan panjang 256 byte maka algoritma menjadi:

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

$$j = (j+S[i]+K[i]) \bmod 256$$

$$j = (0+S[0]+K[0]) \bmod 256$$

$$j = (0+0+49) \bmod 256$$

$$j = (49) \bmod 256$$

$$j = 49$$

Swap S[0] dan S[49]

Iterasi ke-2

$i=1$

$j=49$, maka

$$j = (j+S[i]+K[i]) \bmod 256$$

$$j = (49+S[1]+K[1]) \bmod 256$$

$$j = (49+1+50) \bmod 256$$

$$j = (100) \bmod 256$$

$$j = 100$$

Swap S[1] dan S[100]

Iterasi ke-3

$i=2$

$j=100$, maka

$$j = (j+S[i]+K[i]) \bmod 256$$

$$j = (100+S[2]+K[2]) \bmod 256$$

$$j = (100+2+51) \bmod 256$$

$$j = (153) \bmod 256$$

$$j = 153$$

Swap S[2] dan S[153]

Iterasi ke-4

$i=3$

j=153,maka
 $j = (j+S[i]+K[i]) \bmod 256$
 $j = (153+S[3]+K[3]) \bmod 256$
 $j = (153+3+52) \bmod 256$
 $j = (208) \bmod 256$
 $j = 208$
 Swap S[3] dan S[208]
Iterasi ke-5
 $i=4$
 j=208,maka
 $j = (j+S[i]+K[i]) \bmod 256$
 $j = (208+S[3]+K[3]) \bmod 256$

$j = (208+4+53) \bmod 256$
 $j = (265) \bmod 256$
 $j = 9$
 Swap S[4] dan S[9]
Iterasike-256
 $i=255$
 j=78,maka
 $j = (j+S[i]+K[i]) \bmod 256$
 $j = (78+255+49) \bmod 256$
 $j = (282) \bmod 256$
 $j = 26$
 Swap S[255] dan S[145]

Tabel 3. Tabel Key Scheduling Algorithm (KSA)

49	100	153	208	9	63	119	177	237	43	102	163	226	291	102	166
232	127	114	186	255	70	143	218	39	113	30	11	91	173	252	77
160	245	76	160	246	78	168	4	93	184	21	116	213	51	147	245
89	191	34	135	238	87	194	42	148	0	110	80	75	186	43	158
19	133	249	111	231	97	216	81	204	73	200	68	194	66	196	72
201	76	209	56	225	103	239	121	261	147	189	171	58	203	94	238
128	20	170	66	215	110	7	162	63	217	117	19	179	85	244	149
56	221	132	40	206	118	32	204	117	32	205	124	45	219	139	61
241	167	90	15	198	44	58	242	172	104	38	230	163	27	35	230
171	109	49	247	191	137	222	25	228	177	128	76	26	234	188	144
97	52	9	224	185	143	103	65	29	251	214	179	146	115	86	54
24	252	226	202	53	150	127	106	87	65	45	98	11	253	236	221
208	197	188	176	166	158	152	148	141	136	133	132	133	131	131	133
137	143	146	151	158	167	178	186	196	208	222	238	251	10	27	46
67	85	105	127	151	177	200	225	252	25	88	84	114	146	180	216
249	28	65	104	145	183	223	9	175	99	142	187	234	27	78	26

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

Tabel 4. Hasil Karakter ke Bilangan Biner

Plainteks	Decimal	Biner
S	83	0101 0011
a	97	0110 0001
z	122	0111 1010
q	113	0111 0001
y	121	0111 1001
a	97	0110 0001

Berdasarkan Tabel III.4 diatas diperoleh bilang biner dari plaintext Sazqya.

Inisialisasi i dan j dengan $i = 0$; $j = 0$

Iterasi ke-1

$i = (i+1) \bmod 256$

$i = (0+1) \bmod 256$

$i = 1$

$j = (j+S[i]) \bmod 256$

$j = (0+S[1]) \bmod 256$
 $j = 100$
 Swap $S[1]$ dan $S[100]$
 $t = (S[i]+S[j]) \bmod 256$
 $t = (S[1]+S[100]) \bmod 256$
 $t = (215+100) \bmod 256$
 $t = (315) \bmod 256$
 $t = 59$
 $\text{Key}[0] = S[59] = 80$

Iterasi ke-2

$i = (i+1) \bmod 256$
 $i = (1+1) \bmod 256$
 $i = 2$
 $j = (j+S[i]) \bmod 256$
 $j = (100+S[2]) \bmod 256$
 $j = (100+153) \bmod 256$
 $j = 253$
 Swap $S[2]$ dan $S[253]$
 $t = (S[i]+S[j]) \bmod 256$
 $t = (S[2]+S[253]) \bmod 256$
 $t = (27+153) \bmod 256$
 $t = 180$
 $\text{Key}[1] = S[180] = 53$

Iterasi ke-3

$i = (i+1) \bmod 256$
 $i = (2+1) \bmod 256$
 $i = 3$
 $j = (j+S[i]) \bmod 256$
 $j = (253+S[3]) \bmod 256$
 $j = (253+208) \bmod 256$
 $j = 205$
 Swap $S[3]$ dan $S[205]$
 $t = (S[i]+S[j]) \bmod 256$
 $t = (S[3]+S[205]) \bmod 256$
 $t = (131+208) \bmod 256$
 $t = (339) \bmod 256$
 $t = 83$
 $\text{Key}[2] = S[83] = 56$

Iterasi ke-4

$i = (i+1) \bmod 256$
 $i = (3+1) \bmod 256$

$i = 4$
 $j = (j+S[i]) \bmod 256$
 $j = (205+S[4]) \bmod 256$
 $j = (205 + 9) \bmod 259$
 $j = 214$
 Swap $S[4]$ dan $S[214]$
 $t = (S[i]+S[j]) \bmod 256$
 $t = (S[4]+S[214]) \bmod 256$
 $t = (178+9) \bmod 256$
 $t = 187$
 $\text{Key}[3] = S[187] = 98$

Iterasi ke-5

$i = (i+1) \bmod 256$
 $i = (4+1) \bmod 256$
 $i = 5$
 $j = (j+S[i]) \bmod 256$
 $j = (214+S[5]) \bmod 256$
 $j = (214+63) \bmod 256$
 $j = (277) \bmod 256$
 $j = 21$
 Swap $S[5]$ dan $S[21]$
 $t = (S[i]+S[j]) \bmod 256$
 $t = (S[5]+S[21]) \bmod 256$
 $t = (70+63) \bmod 256$
 $t = 133$
 $\text{Key}[4] = S[133] = 44$

Iterasi ke-6

$i = (i+1) \bmod 256$
 $i = (5+1) \bmod 256$
 $i = 6$
 $j = (j+S[i]) \bmod 256$
 $j = (21+S[6]) \bmod 256$
 $j = (21+119) \bmod 256$
 $j = 140$
 Swap $S[6]$ dan $S[140]$
 $t = (S[i]+S[j]) \bmod 256$
 $t = (S[6]+S[140]) \bmod 256$
 $t = (163+119) \bmod 256$
 $t = (282) \bmod 256$
 $t = 26$
 $\text{Key}[5] = S[26] = 30$

Tabel 5. Pseudo-random Generation Algorithm (PRGA)

Index	Kunci	Biner
0	80	0101 0000
1	53	0011 0101
2	56	0011 1000

3	98	0110 0010
4	44	0010 1100
5	30	0001 1110

Berdasarkan Tabel III.5 diatas diperoleh hasil PRGA yang akan digunakan sebagai kunci untuk proses XOR dengan Plainteks.

5) Enkripsi RC4

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

Tabel 6. XOR Proses Enkripsi

<i>Index</i>	<i>Plainteks</i>	<i>Key</i>	<i>Ciphertext = P XOR K</i>	<i>Des (C)</i>	<i>ASCII</i>
0	0101 0011	0101 0000	0000 0011	3	☒
1	0110 0001	0011 0101	0101 0100	84	T
2	0111 1010	0011 1000	0100 0010	66	B
3	0111 0001	0110 0010	0001 0011	19	☒
4	0111 1001	0010 1100	0101 0101	85	U
5	0110 0001	0001 1110	0111 1111	127	☒

Maka Enkripsi bilangan biner menggunakan RC4 adalah chiperteks = TB U ☒.

Hasil dari enkripsi RC4 akan dikombinasikan dengan metode Base64. Maka hasil enkripsi dari RC4 menjadi plaintext untuk dilanjut ke Base64.

6) Kombinasi Enkripsi RC4 dan Base64

Pada enkripsi RC4 diperoleh Ciphertext yaitu = TB U ☒ yang akan dijadikan plaintext, pada tahap encoding base64 akan dilakukan beberapa tahap sebagai berikut;

- Konversi plaintext menjadi bilangan biner 8 bit. Dapat dilihat pada tabel III.7 berikut ini.

Tabel 7. ASCII ke Biner 8 Bit

Kode ASCII	Desimal	Biner 8Bit
☒	3	0000 0011
T	84	0101 0100
B	66	0100 0010
☒	19	0001 0011
U	85	0101 0101
☒	127	0111 1111

- Setelah mengkonversikan ASCII ke bilangan 8bit maka selanjutnya gabungkan Biner 8bit menjadi Biner 24bit, Ketika ada bilangan yang tidak cukup maka ditambahkan 0 sampai semua bilangan menjadi genap 24bit. 00000011 01010100 01000010 / 00010011 01010101 01111111.
- Setelah mengkonversikan Bilangan 8bit ke bilangan 24bit maka selanjutnya pecah Biner 24bit sebelumnya menjadi Biner 6bit 000000 / 110101/ 010001 / 000010 / 000100 / 110101 / 010101 / 111111. Setelah mengkonversikan Bilangan 24bit ke 6Bit maka selanjutnya bilangan biner 6bit tersebut dikonversikan ke kode ASCII
- Diperoleh hasil kombinasi enkripsi RC4 dan Base64 yaitu A1RCE1V/

- Dekripsi RC4 dan Base64

1) Decoding Base64

Pada encoding base64 diperoleh hasil yaitu w6zDv8O/EyjDkhvCh8O0w6U=, pada tahap decoding base64 akan dilakukan beberapa tahap sebagai berikut.

- Konversikan Kode ASCII hasil encoding ke bilangan Biner 6bit
 $A = 000000$, $1 = 110101$, $R = 010001$, $C = 000010$, $E = 000100$,
 $1 = 110101$, $V = 010101$, $/ = 111111$
- Gabungkan bilangan Biner 6bit menjadi Biner 24bit
 $00000011\ 01010100\ 01000010\ / \ 00010011\ 01010101\ 01111111$.
- Pecah bilangan Biner 24bit menjadi Biner 8bit
 $00000011\ / \ 01010100\ / \ 01000010\ / \ 00010011\ / \ 01010101\ / \ 01111111$.
- Konversi bilangan biner 8bit menjadi ASCII dapat dilihat pada table 8 berikut ini

Tabel 8. Konversi Biner 8bit ke ASCII

Biner 8 Bit	Kode ASCII / Plaintext
0000 0011	☒
0101 0100	T
0100 0010	B
0001 0011	☒
0101 0101	U
111 1	☒

Berdasarkan Tabel III.8 diperoleh hasil Decoding Base64. Hasil dari decoding base64 akan dikombinasikan dengan metode RC4. Maka Decoding dari Base64 menjadi Chipertext untuk dilanjut ke RC4.

2. Kombinasi Dekripsi Base64 dan RC4

Proses dekripsi yaitu meng XOR kan pseudorandom byte dengan ciphertext nya terdiri dari 9 iterasi yang akan diubah menjadi karakter-karakter bentuk bilangan biner.

Tabel 9. XOR Proses Dekripsi

Index	ASCII	Chipertext	Key	Plaintext = C XOR K	Des (C)	ASCI
0	☒	0000 0011	0101 0000	0101 0011	83	S
1	T	0101 0100	0011 0101	0110 0001	97	a
2	B	0100 0010	0011 1000	0111 1010	122	z
3	☒	0001 0011	0110 0010	0111 0001	113	q
4	U	0101 0101	0010 1100	0111 1001	121	y
5	☒	0111 1111	0001 1110	0110 0001	97	a

Berdasarkan Tabel III.9 diperoleh hasil dekripsi dari gabungan metode Base64 dan RC4 yang yaitu **Sazqya**.

C. Perancangan Unified Modelling Language (UML)

- Use Case Diagram

Rancangan sistem yang diusulkan ini dimodelkan ke dalam use case diagram yang dimana menjelaskan secara visual konteks dari interaksi antara aktor dengan sistem [12]

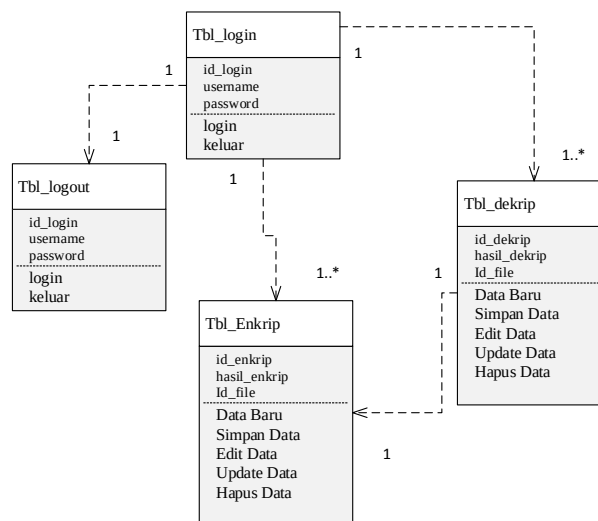


Gambar 2. Use Case Diagram

Gambar 2 diatas merupakan use case diagram dari aplikasi keamanan file dokumen dengan menerapkan metode rc4 dan base64 yang menggambarkan interaksi antara admin dan user. Pada diagram diatas admin dapat mengakses menu untuk mengenkripsi file dokumen dan menu dekripsi file dokumen sedangkan user hanya dapat mengakses menu dekripsi file.

- Class Diagram

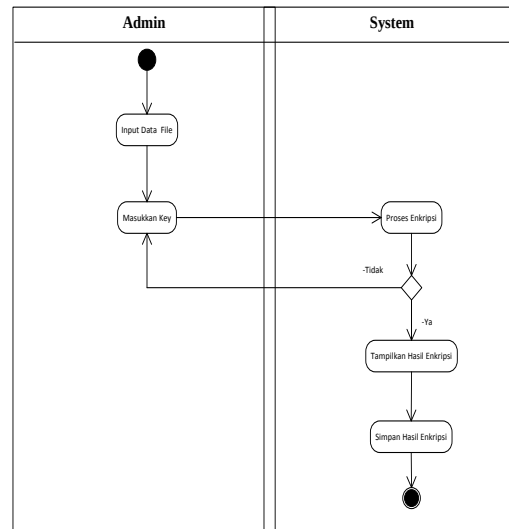
Class Diagram merupakan salah satu jenis diagram UML yang digunakan untuk menggambarkan koneksifitas basis data [13]. Class diagram struktur sistem dari segi pendefinisian kelas-kelas atau kelas objek dan memperlihatkan aturan serta tanggung jawab entitas serta berisikan metode-metode atau aturan tertentu [14]



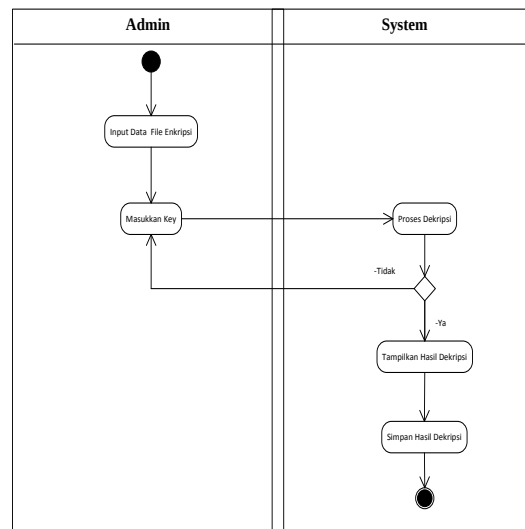
Gambar 3. Class Diagram

- Activity Diagram

Activity Diagram berikut menjelaskan tentang alur dari aktivitas didalam sistem yang sedang dibuat, bagaimana masing-masing aktivitas dimulai, kejadian yang mungkin terjadi, serta bagaimana berakhirnya sistem tersebut [15].

a. *Activity Diagram* File EnkripsiGambar 4. *Activity Diagram* File Enkripsi

Gambar 4 *activity diagram* file enkripsi menjelaskan aktivitas admin dalam proses enkripsi file dengan melakukan *input* berupa *data file* yang dimiliki, selanjutnya masukkan *key* dan sistem akan melakukan proses enkripsi yang diinputkan. Jika *key* yang dimasukkan benar, maka akan diarahkan ke tampilan hasil enkripsi dan menyimpan hasil nya, namun jika *key* salah, maka akan diarahkan untuk mengulangi pengisian *key*.

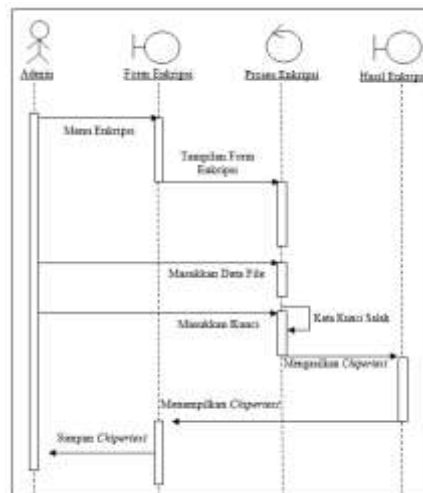
b. *Activity Diagram* File DekripsiGambar 5. *Activity Diagram* File Dekripsi

Gambar 5 *activity diagram* file dekripsi menjelaskan aktivitas admin dalam proses dekripsi file dengan melakukan *input* berupa *data file* yang dimiliki, selanjutnya masukkan *key* dan sistem akan melakukan proses dekripsi yang diinputkan. Jika *key* yang dimasukkan benar, maka akan diarahkan ke tampilan hasil dekripsi dan menyimpan hasil nya, namun jika *key* salah, maka akan diarahkan untuk mengulangi pengisian *key*.

- Sequence Diagram

Sequence diagram untuk menggambarkan interaksi antara sejumlah objek dalam urutan waktu. tahapan ini mengurutkan rincian proses jalannya sistem agar mencapai tujuan pada use case, interaksi antar kelas, operasi mana yang terkait, pengurutan operasi, dan informasi untuk setiap operasi [15].

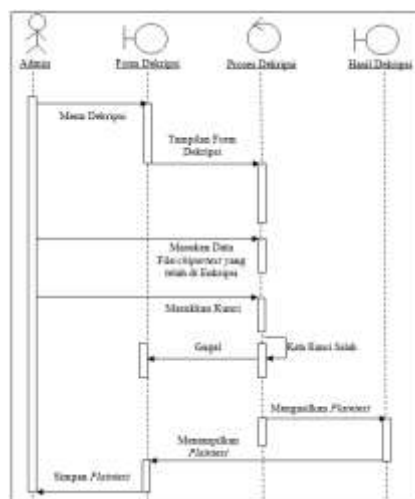
a. *Sequence Diagram* File Enkripsi



Gambar 6. *Sequence Diagram* File Enkripsi

Gambar 6 *sequence diagram* file enkripsi yang dilakukan oleh admin menjelaskan tentang *diagram sequence* file enkripsi. Dimulai dengan admin membuka menu enkripsi dan sistem akan menampilkan halaman enkripsi lalu admin melakukan inputan data berupa file dan setelah itu menginputkan kunci nya, jika *key* benar maka sistem akan menghasilkan *chiphertext* dan akan menampilkan nya pada admin, namun jika *key* salah, maka sistem akan diarahkan untuk mengulangi pengisian *key*.

b. *Sequence Diagram* File Dekripsi



Gambar 7. *Sequence Diagram* File Dekripsi

Gambar 7 *sequence diagram* file dekripsi yang dilakukan oleh admin menjelaskan tentang *diagram sequence* file dekripsi. Dimulai dengan admin membuka menu dekripsi dan sistem akan menampilkan halaman enkripsi lalu admin melakukan inputan berupa file *chiphertext* yang sebelumnya telah di enkripsi dan diikuti memasukkan kunci nya. Jika kunci nya benar maka akan menghasilkan *plaintext* dan

menampilkannya ke admin, namun jika kuncinya salah maka akan gagal dan diarahkan untuk mengulangi pengisian kunci.

4. KESIMPULAN

Berdasarkan hasil pembahasan dan uji coba yang telah dilakukan, maka penulis menarik beberapa kesimpulan adalah sebagai berikut :

- a. Aplikasi yang dibangun dapat menjaga keamanan file dokumen berupa data pegawai dengan algoritma kriptografi kombinasi Base64 dan RC4
- b. Dengan menggunakan kombinasi metode RC4 dan Base64 dapat melindungi data pegawai dengan melakukan penerapan berupa enkripsi dan dekripsi terhadap file dokumen data pegawai.

Adapun penulis menyadari bahwa aplikasi yang telah dihasilkan belum sempurna, untuk itu penulis memberikan usulan atau saran kepada pembaca untuk dapat mengembangkan aplikasi ini agar dapat mengurangi kekurangan yang terdapat pada aplikasi ini. Adapun saran yang dapat peneliti berikan bagi pengembangan aplikasi selanjutnya :

- a. Diharapkan untuk aplikasi selanjutnya agar dalam menggunakan kata kunci hal ini dilakukan agar saat pemilihan kata kunci, *user* bebas membuat kata kunci sesuai dengan keinginan *user*.
- b. Diharapkan untuk dikembangkan agar proses enkripsi (*chiphertext*) dengan sistem yang dibuat dapat menghasilkan data yang berbeda.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Dosen Pembimbing telah membimbing dan membantu dalam proses pembuatan Karya Tulis Ilmiah ini dan penulis juga mengucapkan terima kasih banyak kepada Universitas Potensi Utama yang telah memberikan kesempatan kepada penulis agar menyelesaikan Karya Tulis Ilmiah ini. Penulis berharap karya ilmiah ini dapat diambil ilmu dan manfaatnya.

REFERENSI

- [1] H. Saputra Djong and S. Siswanto, "Implementasi Kriptografi Dengan Menggunakan Metode RC4 Dan AES-256 Untuk Mengamankan File Dokumen Pada PT Varnion Technology Semesta," *Seminar Nasional Fakultas Teknologi Informasi*, pp. 149–158, 2022.
- [2] M. S. Syukur and M. Hardjianto, "Penerapan Algoritma RC4 Untuk Pengamanan Dokumen Pada PT.Linard Power Kontraktor," *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi*, vol. 2, no. 2, pp. 233–241, 2023.
- [3] F. C. Wahyu, A. P. Rahangiar, and F. de Fretes, "Penerapan Algoritma Gabungan RC4 dan Base64 Pada Sistem Keamanan E-Commerce," *Seminar Nasional Aplikasi Teknologi Informasi*, pp. 47–52, 2021.
- [4] R. Wardhani, S. Rahmah Nurshiami, and N. Larasati, "Komputasi Enkripsi Dan Dekripsi Menggunakan Algoritma Hill Cipher," *Jurnal Ilmiah Matematika dan Pendidikan Matematika (JMP)*, vol. 14, no. 1, pp. 45–58, 2022.
- [5] K. F. Ikhsan, F. Fahurian, and A. Hafiz, "Rancang Bangun Aplikasi Cloud Storage Dengan Angular Dan Firebase Berbasis Android," *Jurnal Management Sistem Informasi dan Teknologi*, pp. 43–49, 2022.

- [6] K. Abi Perbawa and Diana, "Penerapan Algoritma Linear Congruential Generator (LCG) dalam Game Edukasi Matematika Berbasis Android," *JURNAL KOMITEK*, vol. 2, no. 1, pp. 47–56, 2022, doi: 10.53697/jkomitek.v2i1.
- [7] Z. Basim and Painem, "Implementasi Kriptografi Algoritma RC4 Dan 3DES Dan Steganografi Dengan Algoritma EOF Untuk Keamanan Data Berbasis Desktop Pada Smk As-Su'udiyah," *Skanika*, vol. 3, no. 4, pp. 54–60, 2020.
- [8] A. A. Cherniaieva, "Frequency of asymptomatic hyperuricemia among adults with type 1 and type 2 diabetes," *Miznarodnij Endokrinologichnij Zurnal*, vol. 16, no. 4, pp. 327–332, 2020, doi: 10.22141/2224-0721.16.4.2020.208486.
- [9] I. Anas, G. L. Ginting, E. Ndruru, A. S. Sembiring, and T. Zebua, "Perancangan Aplikasi Keamanan Data Dengan Kombinasi Algoritma Kriptografi RC4 dan One Time Pad," *JURIKOM (Jurnal Riset Komputer)*, vol. 8, no. 1, pp. 20–27, 2021, doi: 10.30865/jurikom.v7i5.2541.
- [10] Arisetiaji, F. Piliang, and N. Sariana, "Rancang Bangun Aplikasi Pemesanan Desain Jersey Berbasis Android Dengan Menggunakan Teknologi Firebase (Studi Kasus : Konfeksi Minister)," *Jurnal Sistem Informasi dan Sains Teknologi*, vol. 2, no. 2, 2020.
- [11] S. Setiono Puteri and S. Waluyo, "Aplikasi Pengamanan Surat Dengan Metode RC4 Berbasis Web Di Keluarahan Pakujaya Tangerang Selatan," *Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)*, vol. 287, no. 294, 2022.
- [12] E. B. Pratama and U. Saparingga, "Pemodelan UML Sistem Informasi Administrasi Kependudukan Untuk Kantor Desa," *Jurnal Ilmiah Media Sisfo*, vol. 15, no. 2, pp. 107–118, Oct. 2021, doi: 10.33998/mediasisfo.2021.15.2.1085.
- [13] M. Sumiati, R. Abdillah, and A. Cahyo, "Pemodelan UML untuk Sistem Informasi Persewaan Alat Pesta," *Jurnal Fasilkom*, vol. 11, no. 2, pp. 79–86, 2021.
- [14] Suendri, "Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan)," *ALGORITMA: Jurnal Ilmu Komputer dan Informatika*, vol. 3, no. 1, pp. 1–9, 2018, [Online]. Available: <http://www.omg.org>
- [15] Z. Zulkifli, "Rancang Bangun Website E-Learning Dengan Pemodelan UML," *INTECOMS: Journal of Information Technology and Computer Science*, vol. 1, no. 2, pp. 159–167, Jul. 2018, doi: 10.31539/intecom.v1i2.291.