ENKRIPSI EMAIL FILE MENGGUNAKAN ALGORITMA DES DAN TEXT DENGAN ALGORITMA RC6 DAN VIGENERE CIPHER

Hasan Jalu Wiranto[1)], Rizky Tahara Shita[2)]
[1]*Teknologi Informasi, Fakultas Teknologi Informasi, Universitas Budi Luhur*
[1,2]*Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260*
E-mail : hasanjalu1845@gmail.com[1)], rizky.tahara@gmail.com[2)]

ABSTRAK

Salah satu fitur internet yang banyak digunakan untuk bertukar informasi adalah Email atau surat elektronik. Namun permasalahan keamanan informasi yang muncul dalam email yang sering dijumpai antara lain penyadapan pasif dan penipuan.. Secara umum, email tidak menjamin kerahasiaan dari pesan yang dikirim oleh pengguna. Karena teks pesan yang dikirim kadang-kadang adalah pesan rahasia dan pribadi. Oleh sebab itulah dibangun sebuah aplikasi yang memiliki keamanan yang cukup menjamin keamanan dan kerahasiaan data tersebut dan Salah satu cara mengamankan pertukaran informasi melalui email adalah dengan menggabungkan aplikasi email dengan teknik kriptografi, yakni teknik kriptografi yang berfungsi mengubah pesan dan file dokumen asli menjadi file dokumen yang tidak bisa dibaca atau dibaca(terenkripsi) dan untuk membaca pesan dan file yang sudah terenkripsi dengan melakukan proses dekripsi yaitu proses pengembalian pesan dan file seperti semula. Perancangan aplikasi pada tugas akhir ini menggunakan metode RC6 (Rivest Code 6), Vigenere Cipher dan DES (Data Encryption Standard). Didalam aplikasi ini terdapat fitur yang menyajikan semua pesan masuk dan pesan keluar. Dengan menggunakan aplikasi ini, diharapkan dapat membantu mengatasi ancaman atas keamanan informasi dan para pengguna bisa mengirimkan email yang bersifat privasi ini itu dengan aman dan terjaga kerahasiaannya.

Kata Kunci : Kriptografi, Email, RC6, Vigenere Cipher, DES

1. PENDAHULUAN

Era globalisasi saat ini hampir semua perusahaan telah melakukan transaksi melalui email yang bersifat privasi, tapi hal ini tidak menutup kemungkinan terjadinya peretasan email terebut. Karena sebuah transaksi adalah sebuah alur bisnis yang sangat penting bagi setiap prusahaan yang telah terkomputerisasi dimana kelancaran perkembangan perusahaan terletak dalam alur bisnis ini. didalam email sendiri telah terdapat sistem keamanan tersendiri, dimana hal ini tidak bisas dijadikan acuan umum untuk menjaga keamanan email tersebut. Maka akan lebih baik jika kita melakukan pengamanan ganda/berlapis yang keamananya bisa kita atur dan pantau sendiri email tersebut.

Salah satu cara untuk mencegah peretasan email adalah dengan penggabungan aplikasi email dengan algoritma kriptografi. Dengan menerapkan algoritma kriptografi pada email yang dikirim, maka isi email menjadi sulit untuk dibaca karena telah dienkripsi sehingga hanya dapat dibaca dengan menggunakan kunci enkripsi. Tujuan dari pembuatan aplikasi ini adalah membuat aplikasi enkripsi email berbasis dekstop. Dengan adanya aplikasi ini, pengguna dapat mengamankan isi email yang dikirim maupun yang diterima sehingga integritas email yang sifatnyanya personal atau rahasia dapat terjaga. Algoritma yang digunakan untuk mengamankan email ini adalah RC6 dan Vigenere cipher dan DES. Dimana aplikasi ini diamankan dengan tiga algoritma berbeda dan terpisah pada text atau body email dan file yang disipkan. Text atau body email di amankan dengan algoritma RC6 dan vigenere cipher, lalu untuk file diterapkan algoritma DES. Sehingga saat email diterima oleh si penerima maka email tersebut akan terenkripsi dengan tiga metode algoritma tersebut.

2. LANDASAN TEORI

2.1. Email

Email sendiri adalah metode untuk mengirimkan pesan dalam bentuk digital. Pesan ini berbasis internet dan dikirimkan melalui perangkat internet. Didalam email tersebut terdapat isi dari email tersebut lalu alamat penerima, alamat pengirim dan jika ada file yang disisipkan maka akan tampil sisipan file tersebut. Sistem *e-mail* sendiri beroperasi di atas jaringan berbasis pada model *store* and *forward*. Didalam sistem ini mengaplikasikan sebuah sistem *server* e*mail* yang berfungsi untuk mengirim, menerima, meneruskan, serta melakukan penyimpanan pesan-pesan yang sudah dikirim maupun yang diterima oleh pengguna. *E-mail* laksana sebuah kotak surat yang ada di kantor pos jadi sebuah *mail server* dapat memiliki banyak

account e*mail* yang ada didalamnya. Proses pengiriman *email* melalui beberapa *protocol*, yaitu:

1) *Simple Mail Transfer Protocol (SMTP)*
2) *Internet Message Access Protocol*
3) *Post Office Protocol 3 (POP3)* [1]

2.2. Kriptografi

Kriptografi hanya bersangkutan dengan kepercayaan pesan. Yakni dengan mengubah pesan menjadi biasa yang dapat dimengerti oleh hal layak menjadi pesan yang tidak dapat dimengerti, lalu mengubahnya kembali menjadi pesan yang dapat dimengerti. Ini memungkinkan pesan aman oleh orang yang tidak berkepentingan untuk melihat pesan tersebut. Algoritma yang dipakai untuk mengenkripsi dan mendekripsi sebuah *plaintext* melibatkan penggunaan suatu bentuk kunci. Pesan *plaintext* yang telah dienkripsi (atau dikodekan) dikenal sebagai *ciphertext* (teks sandi).[2]

Dalam penulisan metode kriptografi digunakan bahasa matematika. Secara garis besar atau pengertian umum enkripsi adalah merubah pesan asli (plaintext) dengan suatu teknik menjadi sebuah pesan tersandikan (ciphertext).

H = EN (P), dimana : P = pesan asli

EN = proses enkripsi

H = pesan tersandikan

Sedangkan dekripsi adalah proses mengembalikan pesan yang terenkripsi menjadi pesan asli tanpa adanya perubahan sedikitpun.

P = DE (C)

DE = proses dekripsi

Selain menggunakan rumus atau fungsi tertentu umunya enkripsi dan dekripsi menggunakan sebuah parameter tambahan yang dijadikan sandi atau kunci untuk menjalankan enkripsi dan dekripsi tersebut.

Kriptografi yang baik tidak ditentukan oleh kerumitan dalam mengolah data atau pesan yang akan disampaikan. Ada 4 syarat yang perlu dipenuhi, yaitu:

1) Kerahasiaan: Pesan (*plaintext*) hanya dapat dibaca oleh pihak yang memiliki kewenangan.
2) Autentikas*i:* Pengirim pesan harus dapat diidentifikasi dengan pasti, penyusup harus dipastikan tidak bisa berpura-pura menjadi orang lain.
3) Integritas*:* Penerima pesan harus dapat memastikan bahwa pesan yang dia terima tidak dimodifikasi saat dalam proses transmisi data.
4) Non-Repudiation*:* Pengirim pesan harus tidak bisa menyangkal pesan yang dia kirimkan.[3]

2.3. RC6

Metode rc6 dibangun dengan menitik utamakan untuk memenuhi syarat AES yang diantaranya adalah kemampuan untuk beroprasi pada mode blok 128 bit. Pada RC6 ini menggunakan 4 *register* 32 *bit*. Oleh karena itu maka akan terdapat 2 operasi rotasi pada setiap *half-round* yang ada dan juga akan lebih banyak *bit* yang digunakan dan ini akan mempengaruhi banyaknya *bit* yang melakukan rotasi. Algoritma RC6 adalah versi yang dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC6-w/r/b, dimana parameter w merupakan ukuran kata dalam satuan *bit*, r adalah bilangan bulat bukan negatif yang menunjukan banyaknya iterasi selama proses enkripsi, dan b menunjukan ukuran kunci enkripsi dalam *byte*.[4]

Dalam pembangunan sistem yang akan dibuat setelah melalui proses analisa maka alur proses enkripsi menggunakan metode Rivert Code 6 (RC6) terhadap proses enkripsi pesan akan diterapkan pada saat *e-mail* akan dikirimkan. Berikut adalah garis besar langkah – langkah dalam proses pembuatannya :

1) Operasi Dasar :

1. A + B (penjumlahan bilangan integer)
2. A – B (pengurangan bilangan integer)
3. A ⊕ B (exclusive-OR (XOR))
4. A x B (perkalian bilangan integer)
5. A<<<B (A dirotasikan ke kiri sebanyak variabel kedua (B))
6. A>>>B (A dirotasikan ke kanan sebanyak variabel kedua (B))

2) Algoritma Enkripsi

```
1. 1Start
2. Input Text
3. Input key
4. Inisialisasi key
5. Int length = 16 – pesan
6. for(i=0;i<data.length+leng
   ht;i++)
```

```
7.  if(i>0 && i%16 == 0)
8.  B = B + S[0]
9.  D = D + S[1]
10. I = 0
11. 1 = i + 1
12. t = (Bx(2xB+1)5)
13. u = (Dx(2xD+1)5)
14. A = ((A XOR t)u ) + S[2xi]
15. C = ((C XOR )t ) +
    S[2xi+1]
16. i=r
17. A = A + S[2r+2]
18. C = C + S[2r+3]
19. Output Berkas Enkrip
20. END
```

3) Algoritma Dekripsi

```
1.  Start
2.  Input Text
3.  Input key
4.  Inisialisasi key
5.  for(i=0;i<data.length+lenght
    ;i++)
6.  if(i>0 && i%16 == 0)
7.  A = A – S[2r+2]
8.  C = C – S[2r+3]
9.  i = r + 1
10. i = i – 1
11. u = (Dx(2D + 1)) 5
12. t = (Bx(2B + 1))  5
13. C = ((C – S[2xi+1]) t) XOR u
14. A = ((A – S[2xi]) u) XOR t
15. I = 0
16. D = D – S[1]
17. B = B – S[0]
18. Output berkas  dekrip
19. END
```

2.4. DES

Algoritma DES masuk kedalam jenis simetris yang biasa disebut juga sebagai algoritma konvensional. Algoritma konvensional adalah algoritma yang menggunakan kunci enkrip dan dekrip yang sama. Banyak yang sudah melakukan penelitian menggunakan metode ini pada berbagai media seperti berkas digital, gambar, dan lain-lain. DES sendiri mentransformasi *input* 64 *bit* dalam beberapa tahap enkripsi ke dalam *output* 64 *bit.* Inilah sebabnya DES masuk kedalam *block cipher*. Dengan tahapan serta kunci yang sama DES digunakan untuk membalik hasil enkripsi tersebut.

Pada algoritma DES ini kunci internalnya dibankitkan dengan kunci eksternal (external key) 64 *bit*. Ini adalah Skema global dari proses algoritma DES yang biasa digunakan.

Dalam algoritma *DES*, terdapat kunci eksternal dan kunci internal. Setelah didapatkan 56 *bit* hasil permutasi, selanjutnya 56 *bit* ini akan dibagi menjadi 2 bagian, kiri dan kanan, yang masing-masing panjangnya 28 *bit*. Lalu ke-2 bagian tersebut akan disimpan ke dalam C_0 dan D_0.[5]

1. Algoritma Enkripsi

```
1.  Start
2.  Input Kunci
3.    Inisialisasi Kunci Internal
4.    Input Berkas Awal
5.    Baca Hingga Akhir File
6.    If EOF Then
7.      Ke Baris 20
8.  Else
9.      Read 64 Bit Berkas Awal
10.     If <64 Bit Then
11.       Padding Sampai 64 Bit
12.       Ke Baris 14
13. Else
14.     Proses Enkripsi
15.     Kembali Ke Baris 6
16.     Kembali Ke Baris 18
17. End If
18.       Output berkas Hasil
          Enkripsi
19. End If
20. End
```

2. Algoritma Dekripsi

```
1.   Start
2.   Input Kunci
3.     Inisialisasi Kunci Internal
4.     Input Berkas Awal
5.     Baca Hingga Akhir File
6.     If EOF Then
7.       If Has Padding Then
8.         Delete Padding
9.         Ke Baris 11
10.      Else
11.  Output berkas Hasil Enkripsi
12.      End If
13.    Else
14.        Read 64 Bit Berkas
           Awal
15.        Proses Dekripsi
16.        Kembali ke Baris 6
17.    End If
18.  End
```

2.5. Vigenere Cipher

Enkripsi dengan menggunakan metode *Vigenere Chiper* merupakan sebuah metode yang mengadaptasi prinsip kerja dari metode *caesar cipher* yakni melakukan enkripsi karakter pada *plainttext* menjadi karakter lain pada *ciphertext.* Dalam teknik subtitusi *vigenere cipher* ada dua cara yakni angka dan huruf. Contohnya dari penggunaan angka adalah dengan menggunakan tabel seperti pada gambar 1.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
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

Gambar 1 Tabel Vigenere Cipher Perubahan Menjadi Angka

Jadi jika ingin mengubahnya kedalam angka huruf " JALU" akan menjadi (9, 0, 11, 20)

Dan *plaintext* nya "SAYA JALU" akan menjadi :

P = (18, 0, 24, 0, 9, 0, 11, 20).

Chipertext yang dihasilkan:

Chipertext = (2, 0, 10, 20, 18, 0, 22, 15)

Chipertext yang dihasilkan dengan huruf menjadi "CAKU SAWP"

Untuk melakukan deskripsi, bisa juga digunakan modulo 26)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 2 Contoh Table Vigenere Cipher Dengan Huruf

Pada gambar 2 berisikan alfabet yang dituliskan dalam 26 bauh baris. Untuk menggunakan table diatas adalah dengan mengurutkan huruf *plainteks* dengan huruf kunci yang segaris lurus sehingga mendapatkan huruf yang akan menjadi *ciphertext*. Untuk melakukan proses dekripsinya itu menggunakan kebalikan dari proses enkripsinya.

Contohnya seperti dibawah ini :

Plantext: SAYA HASAN

Kunci: JALU

Dari *Plaintext* dengan kata kunci di tabel didapatkan *chipertext* sebagai berikut:

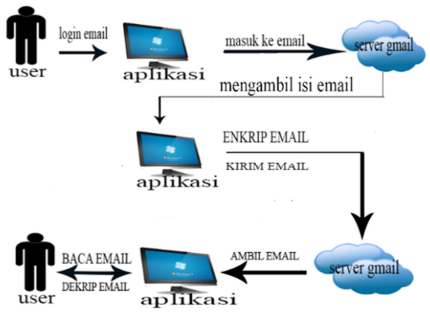
Chipertext: BAJU QADUW

Proses dekripsi, dilakukan dengan mencari huruf chipertext pada baris *plaintext* dari kata kunci.[6]

Setiap *vigenere cipher* berbeda beda sesuai dengan metode yang digunakan dan akan melalui berbagai macam perubahan.

3. Hasil Dan Pembahasan

Dalam program ini dilakukan proses penyadian pesan atau enkripsi pesan pada proses *compose* atau membuat pesan disana terdapat tombol untuk mengenkrip pesan dan pengguna juga bisa mengirim pesan biasa tanpa melakukan proses enkrip. Proses enkrip bisa dilakukan dengan memasukan pasword untuk pesan tersebut agar pesan tersebut terenkrip. Lalu proses dekrip terdapat pada saat *read* pesan atau baca pesan dimana disana akan ada pilihan untuk membuka pesan dengan memasukan password yang sama agar proses dekrip bisa dilakukan dengan benar. Bila password yang dimasukan berbeda dengan saat proses enkrip maka pesan akan tetap berubah menjadi pesan yang tidak bisa dibaca. Berikut arsitektur sistem yang dibuat:



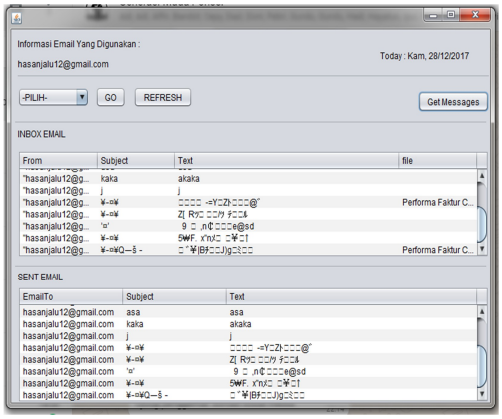
Gambar 3 Arsitektur Sistem

Berikut ini akan dipaparkan hasil dari pembuatan aplikasi beserta proses enkrip dan dekrip yang terdapat pada aplikasi yang dibuat. Dimana aplikasi ini menyediakan beberapa menu dan fungsi yang saling memiliki keterkaitan satu sama lain. Pada saat aplikasi pertama dijalankan maka akan muncul menu *mainframe* seperti pada gambar 4 ini. *User*

memiliki 3 pilihan menu yakni help, *login* dan *about us*.

Gambar 4 Tampilan Awal Aplikasi

Main program dari aplikasi ini adalah dengan menekan tombol login lalu user akan masuk ke menu login dan menginput nama email serta password email untuk masuk ke menu home yang berisikan inbox email, sent mail dan ada combo box yang berisikan untuk melakukan compose email dan logut. Jadi jika user ingin menmbuat email dia harus masuk ke combo box dan memilih menu compose email agar bisa mengakses form compose email. Didalam combo box juga terdapat menu logout yang berfungsi untuk keluar dari menu home dan akan kembali ke menu login. Didalam form home pula juga ada tombol refresh yang berfungsi untuk menrefresh table inbox serta table outbox jika terdapat email baru yang masuk. Menu home akan menampilkan tampilan seperti gambar 5 dibawah ini.



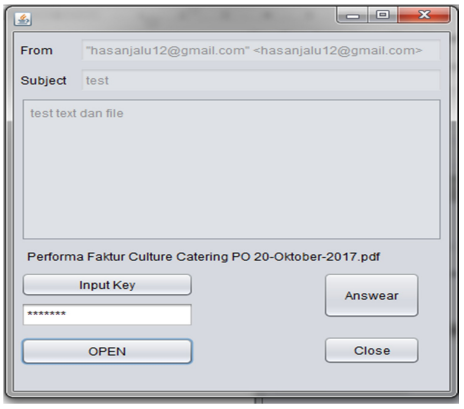
Gambar 5 Menu Home

Didalam program ini proses pengamanan pesan yang dilakukan ada pada saat kita melakukan compose email dimana kita akan menginput pesan dan alamat email tujuan beserta file bila ada file yang ingin di lampirkan.

Lalu si penerima akan menerima pesan yang sudah terenkrip. Seperti yang ditampilkan pada gambar 6 dibawah ini.

Gambar 6 Email yang diterima dan akan mendekrip email tersebut

Maka pesan yang telah diinput dengan password yang benar maka terbuka akan seperti gambar 7 dibawah ini.

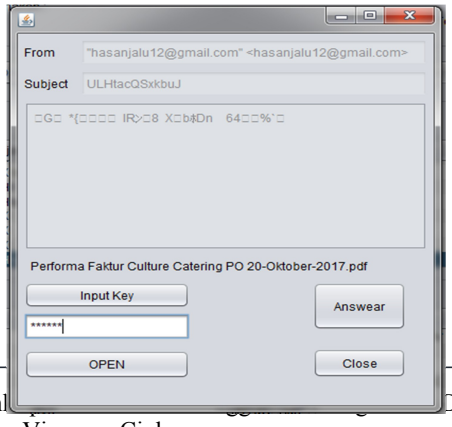


Gambar 7 Email yang Telah Terdekrip

Berikut ini adalah table pengujian yang telah dilakukan dengan beberapa file yang telah melalui proses enkrip dan dekrip.

Table 1 Hasil Pengujian File Yang Terenkrip

InputFile	Password	size File Input	Ukuran Hasil Enkrip	Output File



PEMASUKAN.docx	test	15 KB	20 KB	ENK_PEMASUKAN.docx
Bahan.xlsx	Test12	11 KB	15 KB	ENK_Bahan.xlsx
Faktur pengiriman.pdf	Test123	20	27	Enk_faktur pengirimian.pdf

Table 2 Hasil Pengujian File Yang Terdekrip

InputFile	Password	size *File Input*	Ukuran hasil	*Output File*
ENK_PEMASUKAN.docx	test	20 KB	15 KB	PEMASUKAN.docx
ENK_Bahan.xlsx	Test12	15 KB	11 KB	Bahan.xlsx
Enk_faktur penjualan.pdf	Test123	27	20	Faktur penjualan.pdf

4. KESIMPULAN

Berdasarkan penjelasan diatas dan penyelesaian masalah dari permasalahan dan telah dilihat dari aplikasi yg telah dibuat untuk menyelesaikan permasalahan tersebut maka bisa disimpulkan beberapa hal yakni:

a. Aplikasi pengamanan email ini setiap proses pengirimian email menjadi lebih aman dari pihak yang tidak berkepentingan.
b. Proses dekripsi tetap berjalan meskipun password yang dimasukan tidak sesuai, namun email dan file yang terdekripsi tidak bias dibaca dan file tidak bisa dibuka.
c. Proses dekripsi dengan password yang benar akan mengembalikan email dan file menjadi email dan file semula tanpa mengalami perubahan.
d. Waktu dalam proses enkripsi dan dekripsi yang dibutuhkan berbanding lurus dengan ukuran file yang dipakai.

Selain membuat sejumlah kesimpulan, dapat pula dianjurkan beberapa saran-saran yang mungkin bisa dijadikan pertimbangan dalam pengembangan sistem, antara lain:

a. Keterbatasan file yang bisa disisipkan hanya file *.doc, *.docx, *.pdf, *.xls, *.xlsx dan untuk itu kedepannya perlu dikembangkan untuk menambahkan *file extension* lainnya.
b. Interface masih sangat sederhana diharapkan dapat di implementasikan kedalam aplikasi berbasis *web* ataupun *mobile*.
c. Dalam proses dekripsi *file* yang berukuran besar diharapkan dapat berjalan dengan lebih cepat.
d. Didalam pengembanganya aplikasi ini bisa menggunakan metode kompresi agar lebih efisien dalam penyimpanan jika terdapat file-file besar yang masih bisa lebih diminimalisir.

5. DAFTAR PUSTAKA

[1] Hamid, "Uji Keamanan Aplikasi Email Bawaan Android Pada Jaringan Nirkabel," *J. Cybermatika*, vol. 2, no. 1, hal. 13–19, 2014.

[2] F. N. Pabokory, I. F. Astuti, dan A. H. Kridalaksana, "Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard," *J. Inform. Mulawarman*, vol. 10, no. 1, hal. 20–31, 2015.

[3] A. Rohmanu, "Implementasi Kriptografi dan Steganografi Dengan Metode Algoritma Des dan Metode End Of File Ajar Rohmanu," vol. 1, no. 2, hal. 1–11, 2017.

[4] M. Zulham, H. Kurniawan, dan I. F. Rahmad, "Perancangan Aplikasi Keamanan Data Email Menggunakan Algoritma Enkripsi Rc6," *Semin. Nas. Inform.*, hal. 96–101, 2014.

[5] R. Primartha, J. T. Informatika, F. I. Komputer, dan U. Sriwijaya, "Penerapan Enkripsi Dan Dekripsi File Menggunakan Algoritma Data Encryption Standard (DES)," *Enkrip dan Dekrip dengan DES*, vol. 3, no. 2, hal. 371–387, 2013.

[6] P. H. Arjana, T. P. Rahayu, Yakub, dan Hariyanto, "Implementasi Enkripsi Data Dengan Algoritma Vigenere Chiper," *Sentika*, vol. 2013, no. Sentika, hal. 164–169, 2013.