Aplikasi Enkripsi Email Dengan Algoritma Gost Dan Caesar Cipher Berbasis Web Pada Ppsdm Universitas Terbuka

Tulus Budiadji Syam[1)] , Wahyu Pramusinto[2)]
Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : tulus.syam@gmail.com 1) , wahyu.pramusinto@budiluhur.ac.id 2)

Abstrak

Bertukar informasi adalah hal umum yang hampir setiap orang pernah melakukannya. Kemajuan teknologi internet yang berkembang dengan sangat pesat telah menjadi manfaat besar. Salah satu media yang dapat digunakan adalah email. Dengan fasilitas email pengguna dapat bertukar informasi dalam bentuk pesan teks maupun mengirimkan file. Secara umum, email tidak menjamin kerahasiaan pesan yang dikirimkan oleh penggunanya. Karena pesan teks yang dikirimkan terkadang adalah pesan yang rahasia dan pribadi, sehingga kerahasiaannya menjadi sangat penting. Salah satu caranya adalah dengan menggabungkan aplikasi email dan kriptografi yang menggunakan algoritma kunci simetris, dimana kunci enkripsi dan dekripsi tersebut sama namun jika kunci dekripsi dikirimkan secara terpisah hal ini memungkinkan kunci dapat diketahui oleh penyadap. Algoritma yang digunakan adalah algoritma GOST dan Caesar Cipher. Algoritma GOST merupakan algoritma simetris blok cipher 64 bit yang dikenal cukup aman karena memiliki 32 putaran dalam proses enkripsi dan dekripsi nya. Algoritma GOST digunakan untuk enkripsi dan dekripsi File, Caesar Cipher digunakan untuk enkripsi dan dekripsi Message. Kesimpulannya adalah aplikasi ini mudah digunakan, pesan yang dikirim dan diterima melalui aplikasi ini aman karena sudah melalui proses enkripsi terlebih dahulu. Diharapkan dengan aplikasi ini pengguna dapat mengirimkan pesan dan data yang sifatnya rahasia tanpa takut akan ada orang lain yang membaca isi pesan dan data tersebut.

Kata kunci: *Email*, Kriptografi, Algoritma GOST, Caesar Cipher

1. PENDAHULUAN

Di era globalisasi ini, perkembangan teknologi komputer dan telekomunikasi telah mengalami kemajuan yang sangat pesat pada saat ini, bukan hanya di negara maju, di negara berkembangpun terjadi peningkatan terhadap penggunaan komputer dalam berbagai bidang, diantaranya untuk bisnis, pendidikan dan bidang-bidang yang lain. Hampir semua hal dapat disajikan dalam bentuk digital atau terkomputerisasi. Secara tidak langsung teknologi informasi telah menjadi bagian penting dari kehidupan manusia.

Salah satu fitur dari internet yang banyak dipakai saat ini ialah surat elektronik atau yang biasa disebut *email*. Karena masih banyak pemakai internet khususnya di Indonesia yang tidak memakai komputer pribadi miliknya, hal itu bisa mengakibatkan terjadinya *email* yang bersifat pribadi terbaca oleh pihak lain yang tidak berkepentingan, hal ini bisa terjadi karena tata cara kerja *email* harus melewati beberapa server sebelum akhirnya sampai ke alamat email yang tujuan. Dalam aspek pengiriman data terdapat suatu hal yang musti di perhatikan yakni keamanannya, agar data yang dikirimkan bias sampai ke tempat tujuan dengan utuh dan tidak berubah sedikitpun.

Permasalahan keamanan informasi dalam *email* yang sering dijumpai antara lain, penyadapan, pencurian data, dan sebagainya. Masalah keamanan yang paling sering terjadi ialah penyadapan. Penyadapan sendiri mengincar data-data penting milik korban yang bisa berdampak negative bagi si pemilikdata tersebut, serta seorang penyadap akan memantau tiap aktifitas.

Berdasarkan uraian latar belakang diatas penulis bermaksud membuat suatu aplikasi pengamanan informasi atau pesan melalui media *email* dengan algoritma GOST (*Gosudarstvennyi Standard*) dan algoritma caesar cipher pada PPSDM Universitas terbuka.

2. LANDASAN TEORI

2.1. Email

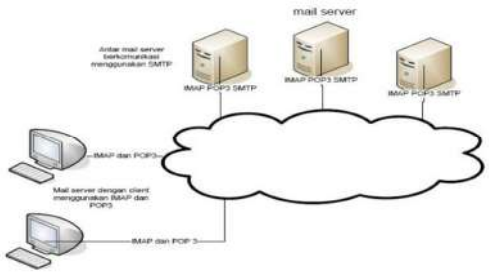
Email merupakan metode untuk mengirim sebuah pesan digital. Pesan digital ini biasanya dikirim melewati internet. Sebuah pesan elektronik terdiri dari isi, alamat si pengirim email, dan alamat si penerima email. Email beroprasi pada model store and forward. Dalam sistem ini menggunakan sebuah sistem *server* e*mail* yang mengirim, menerima, menyisipkan file, meneruskan, dan menyimpan pesan pengguna. *E-mail* laksana kotak POS yang bisa menyimpan dan menerima pesan, sedangkan *server* e*mail* dapat diibaratkan sebagai kantor POS. Maka dari itu, email server bisa memiliki banyak account didalamnya dan memiliki banyak pesan pesan dari para pengguna tersebut.

Untuk mengirimkan e*mail* dari alamat satu ke alamat yang lainnya digunakanlah sebuah *protocol* (aturan) yaitu *Simple Mail* Transfer *Protocol* SMTP. *Protocol* SMTP sudah menjadi aturan dasar yang telah disepakati untuk mengirimankan *e-mail*. Dengan kata lain semua software *e-mail server* pasti mendukung protokol ini.

SMTP adalah protokol yang digunakan untuk megirimkan *e-mail* (komunikasi antara *mail server*), dan tidak digunakan untuk berkomunikasi dengan pengguna. Sedangkan untuk pengguna, digunakan protokol imap imaps pop3 pop3s.

Agar *mail server* dapat di akses oleh pengguna, dikembangkan suatu aplikasi dimana pengguna dapat mengakses *e-mail* dari sebuah *e-mail server*. IMAP ialah sebuah aplikasi pada layer Internet protokol yang memungkinkan pengguna untuk mengakses e*mail* yang ada di *server*. Selain IMAP ada pula POP3 yang berfungsi sama dengan imap, tetapi memiliki beberapa karakteristik yang berbeda dalam cara mengaksesnya pada *server*.

Dalam menjalankan tugasnya, suatu *mail server* harus bisa melayani pengiriman e*mail* yang menggunakan protokol SMTP dan harus mampu melayani pengguna yang ingin mengakses e*mail* dengan menyediakan IMAP dan POP3. [3]



Gambar 1. ***Arsitektur E-mail***

2.2. Kriptografi

a. Pengertian Kriptografi

Kriptografi adalah ilmu tentang teknik enkripsi data yang diacak menggunakan suatu kunci menjadi sesuatu yang sulit dibaca oleh orang yang tidak memiliki kunci tersebut. Dapat disimpulkan, enkripsi adalah suatu proses pengacakan "naskah asli"*(plaintext)* menjadi "naskah acak" *(ciphertext)* yang "sulit untuk dibaca" oleh orang yang tidak mempunyai kunci dekripsi tersebut. Yang dimaksud dengan "sulit untuk dibaca" disini ialah kemungkinan untuk mendapatkan kembali naskah asli oleh seseorang yang tidak mempunyai kunci dekripsi dalam waktu singkat adalah sangat kecil. Jadi suatu proses enkripsi yang baik menghasilkan naskah acak yang memerlukan waktu yang lama untuk didekripsi oleh orang yang tidak mempunyai kunci dekripsi. [2]

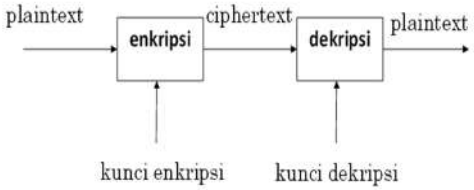
Pada awalnya kriptografi digunakan untuk merahasiakan naskah teks, sekarang kriptografi digunakan untuk semua data yang bersifat digital. Selain berdasarkan sejarah yang membagi kritografi menjadi dua, yaitu kriptografi klasik dan kriptografi modern, maka berdasarkan kunci yang dipergunakan untuk enkripsi dan dekripsi, kriptografi dapat dibagi lagi menjadi dua jenis, yaitu kriptografi Simetris dan kriptografi Asimetris. [1]

b. Jenis Algoritma Kriptografi

Algoritma Kriptografi terbagi menjadi dua berdasarkan kunci yang digunakannya, yaitu Algoritma Simetris (konvensional) dan Algoritma Asimetris (kunci publik).

1) Algoritma Simetris

Algoritma kriptografi simetris merupakan algoritma kriptografi yang menggunakan kunci enkripsi dan kunci dekripsinya sama. Bila mengirimkan pesan dengan menggunakan algoritma ini, si penerima harus diberitahukan kunci dari pesan tersebut supaya bisa mendekripsi pesan yang dikirimkan. Keamanan dari pesan yang digunakan algoritma ini tergantung pada kuncinya. Jika kunci tersebut diketahui orang lain maka orang tersebut dapat melakukan enkripsi serta dekripsi pesan.



*Gambar 2.****Proses Algoritma Simetris***

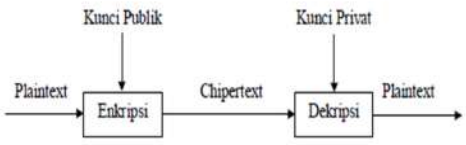
2) Algoritma Asimetris

Algoritma kriptografi asimetris sering juga disebut algoritma kunci publik. Kunci yang digunakan untuk melakukan enkripsi dan dekripsi berbeda. Pada algoritma asimetris kunci terbagi menjadi dua, yaitu:

a) Kunci umum (*public key*): kunci yang diperboleh untuk diberitahukan kepada orang lain (dipublikasikan).

b) Kunci rahasia (*private key*): kunci yang harus dirahasiakan dan tidak boleh tersebar luas (hanya boleh diketahui oleh satu orang).

Dengan menggunakan kunci public, orang dapat mengenkripsikan pesan tetapi tidak bisa untuk mendekripsi. Hanya orang yang mengetahui kunci rahasia yang dapat mendekripsikan pesan tersebut. Dengan kata lain Algoritma asimetris lebih aman daripada algoritma simetris.



*Gambar 3.****Proses Algoritma Asimetris***

2.3. Algoritma GOST

Algoritma GOST merupakan blok kode dari Uni Soviet, yang menjadi singkatan dari *Gosudarstvennyi Standard* atau standar pemerintah. GOST merupakan blok kode 64 bit dengan panjang

kunci 256 bit. Algoritma ini adalah perulangan dari algoritma enkripsi sederhana sebanyak 32 putaran.[4]

Untuk proses enkripsi pertama plainteks 64 bit dipecah menjadi 32 bit bagian kiri, L dan 32 bit bagian kanan, R. Subkunci untuk putaran i adalah Ki. Pada satu putaran ke-i rumushnya adalah seperti berikut :

$$L_i = Ri_{-1}$$
$$R_i = Li_{-1} \text{ xor } f(R_{i-1}, K_i)$$

a. Proses Pembangkitan Kunci

kunci internal pada algoritma GOST dibangkitkan dari kunci eksternal yang diberikan oleh pengguna. Pembangkitan kunci internal dilakukan dengan membagi kunci eksternal 256 bit (k1, k2, k3, k4, …, k256) ke dalam delapan bagian yang masing-masing mempunyai panjang 32 bit. Pembagiannya adalah seperti berikut:

K0 = (k32, ……. ,k1)
K1 = (k64, ……. ,k33)
K2 = (k96, ……. ,k65)
K3 = (k128, ……. ,k97)
K4 = (k160, ……. ,k129)
K5 = (k192, ……. ,k161)
K6 = (k224, ……. ,k193)
K7 = (k256, ……. ,k225)

b. Proses Enkripsi

Proses Enkripsi pada algoritma GOST untuk satu putaran (iterasi), adalah sebagai berikut :

1) 64 bit plainteks dibagi menjadi 2 bagian 32 bit, yaitu L_i dan R_i.Caranya adalah:
   Input a1(0), a2(0), ….. ,a32(0) ; b1(0), b2(0), ….. ,b32(0)
   R0 = a32(0), a31(0), ….. ,a1(0)
   L0 = b32(0), b31(0), ….. ,b1(0)
2) $(R_i + K_i)$ mod 2^{32}. Hasil dari penjumlahan modulo 2^{32} berupa 32 bit.
3) Hasil dari jumlah modulo 2^{32} dibagi menjadi 8 bagian, dimana tiap bagian terdiri dari 4 bit. Setiap bagian dimasukkan ke dalam table *S-box* yang berbeda, 4 bit pertama menjadi input dari *S-Box* 0, 4 bit kedua menjadi *S-Box* 1 dan seterusnya.

Table 1. Tabel S-Box Algoritma GOST

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0x	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
1x	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
2x	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
3x	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
4x	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
5x	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
6x	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
7x	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

4) Hasil yang didapatkan dari substitusi ke S-Box kemudian digabungkan kembali menjadi 32 bit setelah itu dilakukan RLS (*Rotate Left Shift*) pergeseran ke kiri sebanyak 11 bit.
5) R_{i+1} = RLS XOR L_i
6) $L_{i+1} = R_i$ sebelum dilakukannya proses
   Langkah nomor 2 sampai 6 dilakukan sebanyak 32 kali (putaran). Pada langkah nomor 2 penggunaan kunci dijadwalkan penggunaannya sesuai dengan putarannya.

Table 2. Penjadwalan Kunci Internal Enkripsi

Putaran	0	1	2	3	4	5	6	7
Kunci Internal	K0	K1	K2	K3	K4	K5	K6	K7
Putaran	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Kunci Internal	K0	K1	K2	K3	K4	K5	K6	K7
Putaran	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
Kunci Internal	K0	K1	K2	K3	K4	K5	K6	K7
Putaran	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
Kunci Internal	K7	K6	K5	K4	K3	K2	K1	K0

Untuk putaran ke-31, langkah nomor 5 dan 6 terdapat sedikit perberbedan. Langkah 5 dan 6 untuk putaran 31 adalah sebagai berikut ini :

R32 = R31 sebelum dilakukannya proses
L32 = L31 XOR R31

Sehingga menghasilkan cipherteks adalah,
L32 : b(32), b(31), ….. ,b(1)
R32 : a(32), a(31), ….. ,a(1)
Chiperteks = a(1), ….. ,a(32); b(1), ….. ,b(32).

c. Proses Dekripsi

Proses dekripsi adalah kebalikan dari proses enkripsi. Penggunaan kunci masing-masing putaran pada proses dekripsi adalah sebagai berikut :

Table 3. Penjadwalan Kunci Internal Dekripsi

Putaran	0	1	2	3	4	5	6	7
Kunci Internal	K0	K1	K2	K3	K4	K5	K6	K7
Putaran	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Kunci Internal	K0	K1	K2	K3	K4	K5	K6	K7
Putaran	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
Kunci Internal	K0	K1	K2	K3	K4	K5	K6	K7
Putaran	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**
Kunci Internal	K7	K6	K5	K4	K3	K2	K1	K0

Di dalam proses dekripsi terdapat aturan sama dengan proses enkripsi yaitu untuk langkah ke-5 dan ke-6 pada putaran ke-31 seperti berikut :

R32 = R31 sebelum dilakukan proses
L32 = L31 XOR R31

Sehingga, plainteks yang dihasilkan pada proses dekripsi adalah,
L32 : b(32), b(31), ….. ,b(1)
R32 : a(32), a(31), ….. ,a(1)
Plainteks = a(1), ….. ,a(32); b(1), ….. ,b(32).

2.4. Caesar Cipher

Caesar Cipher pada awalnya berasal dari kaisar Roma, yaitu *Julius Caesar*, dia mempergunakan *cipher* substitusi untuk mengirimkan perintah ke panglima perang. *Caesar*

Chiper juga dikenal dengan nama lain seperti: *Shift Cipher, Caesar's Code, atau Caesar Cipher Shif*.

Caesar Cipher adalah metodologi enkripsi pertama. Metode enkripsi *Caesar Cipher* ini adalah *cipher* subtitusi, dimana setiap huruf yang ada pada *plaintext*-nya digantikan dengan huruf lain. Misalnya dengan pergeseran 3 langkah, A akan digantikan oleh D, B akan menjadi E, dan seterusnya. [5]

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
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

Gambar 4. ***Contoh pergeseran tiga langkah***

Untuk merahasiakan sebuah pesan, cukup mencari setiap huruf yang hendak dirahasiakan di alfabet biasa, kemudian tuliskan huruf yang sesuai pada alfabet kode. Untuk memecahkan kode tersebut gunakan cara sebaliknya. Contoh merahasiakan sebuah pesan adalah sebagai berikut.

Plaintext: kirim pasukan ke sayap kiri

Ciphertext: NLULP SDVXNDQ NH VDBDS NLUL

Proses pengkodean (enkripsi) didapat secara matematis mempergunakan operasi modulus dengan mengganti setiap huruf menjadi angka, A = 0, B = 1,..., Z = 25. Sandi (En) dari "huruf" x dengan pergeseran n secara matematis yang dituliskan dengan,

$$E_n(x) = (x+n) \bmod 26$$

Sedangkan pada proses memecahan kode (dekripsi), hasil dari dekripsi (Dn) adalah,

$$D_n(x) = (x+n) \bmod 26$$

Tiap huruf yang sama digantikan dengan huruf yang sama sepanjang pesan, sehingga kode *Caesar* digolongkan kepada, substitusi monoalfabetik, yang berlawanan dengan substitusi polialfabetik

3. ANALISA DAN PERANCANGAN PROGRAM

3.1. Analisa Masalah dan Solusi

Dokumen merupakan data yang sangat penting baik itu berupa dokumen pribadi, perusahaan atau organisasi dan lain sebagainya. *Email* adalah salah satu sarana untuk mengirimkan pesan maupun lampiran *file* seperti dokumen, gambar, video, suara dan lain-lain. Saat ini hampir setiap orang mempunyai akun *email,* baik email pribadi, tempat bekerja dan lain sebagainya. Pada proses pengiriman *email* pada dasarnya hanya melakukan pengiriman tanpa melakukan pengamanan terhadap *file* dari data yang dikirim, sehingga ketika data disadap pada saat pengiriman. Data akan langsung terbaca oleh penyadap. Hal ini dapat menimbulkan masalah, terutama masalah keamanan data pada PPSDM Universitas terbuka yang sebagian besar data atau informasi yang dikirim bersifat rahasia. Untuk memecahkan masalah tersebut, maka dibutuhkan sebuah aplikasi yang dapat menjaga kerahasiaan dari sebuah dokumen atau data dalam pengiriman. Aplikasi ini menggunakan algoritma GOST dan Caesar cipher. Pesan yang dikirimkan melalui aplikasi ini akan dienkripsi terlebih dahulu dan pesan teks yang dikirim akan terdekripsi secara otomatis saat pesan dibuka oleh penerima. Dengan adanya aplikasi ini diharapkan file dokumen tidak dapat dimanipulasi oleh pihak yang tidak bertanggung jawab.

3.2. Perangkat Yang Dibutuhkan

Perangkat yang dibutuhkan dalam membangun aplikasi ini terdiri dari perangkat keras, dan perangkat lunak. Adapun penjelasan dari masing-masing perangkat adalah sebagai berikut:

a. Perangkat Keras

Perangkat keras yang dibutuhkan dalam membangun perangkat lunak ini memiliki spesifikasi, spesifikasi perangkat keras dapat dilihat pada tabel berikut:

Table 4. Spesifikasi Perangkat Keras

No	Perangkat	Kebutuhan
1	CPU	Intel(R) Core(TM) i3-2350M CPU @2.30GHZ
2	Hardisk	500 GB
3	RAM	4.00 GB
4	Monitor	14.0"
5	Keyboard	Internal Keyboard Laptop

b. Perangkat Lunak

Perangkat lunak yang digunakan untuk menunjang kelancaran dalam pembuatan aplikasi ini. Spesifikasi perangkat lunak, dapat dilihat pada tabel berikut:

Table 5. Spesifikasi Perangkat Lunak

No	Perangkat	Kebutuhan
1	Sistem Operasi	Windows 10
2	Bahasa Pemograman	PHP, HTML, *Java script*

4. IMPLEMENTASI DAN UJI COBA PROGRAM

a. Tampilan Layar Halaman Login

Tampilan layar halaman *login* merupakan layar yang akan tampil pertama kali ketika aplikasi dijalankan yang menjadi penghubung ke menu utama. Berikut adalah gambar tampilan layar halaman *login*.

Gambar 5. ***Tampilan Layar Halaman Login***

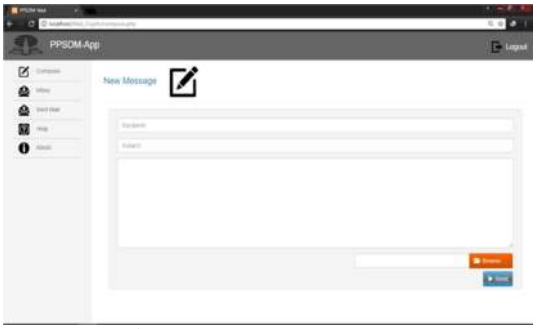
b. Tampilan Layar Menu Utama

Menu utama ini akan tampil ketika *user* sudah berhasil melakukan *login*. Pada halaman ini terdapat beberapa menu yang dapat dipilih *user*, yaitu menu *compose*, menu *inbox*, menu *Sent mail,* menu *help*, menu *about*, serta 1 *button logout*. Untuk lebih jelasnya berikut adalah gambar tampilan layar halaman utama :

Gambar 6. ***Tampilan Layar Menu Utama***

c. Tampilan Layar Halaman Compose

Halaman *compose* adalah halaman yang digunakan *user* untuk mengirimkan *email* menggunakan aplikasi ini dengan cara mengklik menu *compose* yang ada di halaman utama.

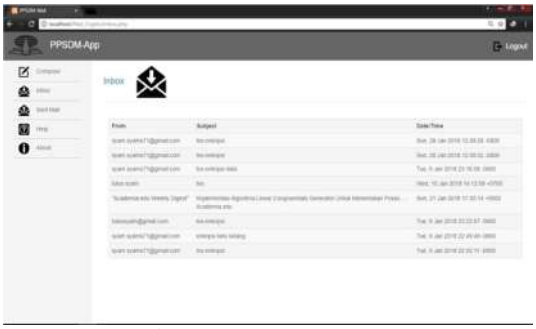


Gambar 7. ***Tampilan Layar Halaman Compose***

d. Tampilan Layar Halaman Inbox

Pada halaman *inbox*, *user* yang baru pertama kali menggunakan aplikasi ini, harus mengklik authorize agar aplikasi dapat membaca inbox yang ada di email.

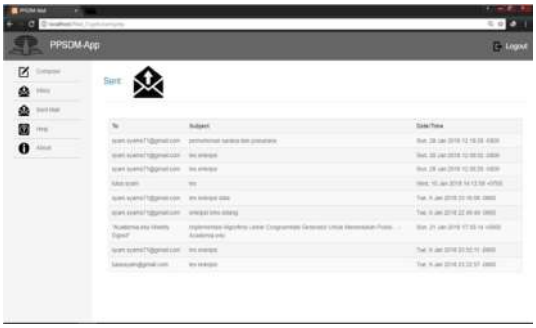
Setelah *user* mengklik *authorize*, maka secara otomatis aplikasi akan meminta perizinan hak akses *email* seperti saat *login*. Selanjutnya *user* baru dapat melihat *form list inbox*. Untuk membaca *email*, *user* dapat melakukannya dengan cara mengklik *subject email* dan kemudian isi *email* akan ditampilkan. Untuk membaca *email*, *user* dapat melakukannya dengan cara mengklik *subject email* dan kemudian isi *email* akan ditampilkan.



Gambar 8. ***Tampilan Layar Halaman Inbox***

e. Tampilan Layar Halaman Sent mail

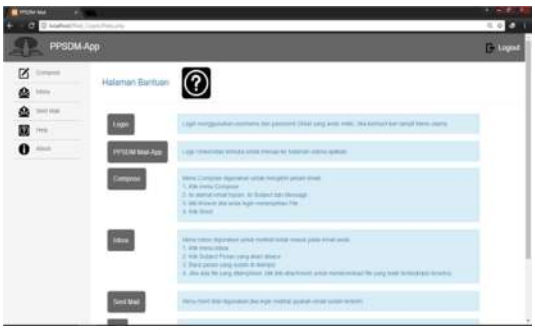
Pada halaman ini, *user* dapat melihat email yang sudah terkirim. Berikut adalah gambar tampilan layar halaman *sent mail* :



Gambar 9. ***Tampilan Layar Halaman Sent mail***

f. Tampilan Layar Halaman Help

Pada halaman ini, *user* dapat mengetahui cara penggunaan aplikasi PPSDM-Mail. Berikut adalah gambar tampilan layar halaman *help* :



Gambar 10. ***Tampilan Layar Halaman Help***

g. Tampilan Layar Halaman About

Tampilan layar halaman *about* merupakan tampilan layar yang berisi informasi mengenai pembuat aplikasi ini.

Gambar 11. ***Tampilan Layar Halaman About***

5. PENUTUP

5.1. KESIMPULAN

Berdasarkan proses perancangan, pembuatan, dan pengujian aplikasi PPSDM-Mail yang digunakan untuk mengamankan isi *email,* maka dapat diambil suatu kesimpulan, yaitu :

a. Aplikasi ini telah diatur oleh sistem sehingga isi pesan atau data yang terkandung dalam akun *email* tersebut sudah terenkripsi secara otomatis.
b. Meminimalisir kemungkinan kebocoran informasi yang terdapat di akun *email* apabila menjadi korban dari *hacker*, karena pesan tersebut terenkripsi.
c. Semakin sedikit teks dan semakin kecil ukuran *file*nya maka semakin cepat pula waktu pengirimannya.

5.2. SARAN

Untuk pengembangan lebih lanjut agar aplikasi ini menjadi lebih baik lagi, adapun saran yang diberikan antara lain:

a. Aplikasi ini hanya dapat menyisipkan file *.txt, *.doc, .docx, *.pdf, *.ppt, *.pptx, *.xls, *.xlsx dan untuk itu kedepannya perlu dikembangkan untuk menambahkan *file extension* lainnya.
b. fitur aplikasi masih sederhana, diharapkan dapat ditambahkan beberapa fitur seperti *forward message, delete message*, dll.
c. Dalam pengembangannya, aplikasi ini dapat menggunakan metode kompresi sehingga ukuran file yang dienkrip atau didekrip dapat lebih diminimalisir.
   Aplikasi diharapkan dapat dikembangkan lagi sehingga dapat mengenkrip format data digital seperti gambar(.*jpg, *.jpeg, *.png, dll) dan suara (*.mp3, *.mp4, dll).

6. DAFTAR PUSTAKA

[1] Busran. dan Mandarani, Putri., Maret 2012, Analisa Komputasi Enkripsi Dan Dekripsi Data Gambar, Teks Dan Audio Dengan Menggunakan Algoritma RC4, Berbasis Visual Basic 6.0, *Jurnal Teknologi Informasi & Pendidikan*.

[2] Kromodimoeljo, Sentot., 2010, *Teori dan Aplikasi Kriptografi*, Jakarta, SPK IT Consulting, 458.

[3] Novasandro, Ridzky. Dan Kautsar, Alvian, Edo., 2013, Prinsip Kerja Protokol - Protokol Electronic Mail, *Jurnal Ilmiah*, pp.1–18.

[4] Siregar, Juleha., Juni 2016, Implementasi Keamanan Data Teks Dengan Algoritma Gost Dan Rot13, *1*(2), 68–73.

[5] Puspita, Khairani. dan Wayahdi, M, Rhifky., Februari 2015, Analisis Kombinasi Metode Caesar Cipher, Vernam Cipher, Dan Hill Cipher Dalam Proses Kriptografi, *Seminar Nasional Teknologi Informasi Dan Multimedia*, 43–48.