IMPLEMENTASI ALGORITMA BLOWFISH PADA APLIKASI PENGAMANAN SURAT ELEKTRONIK (*MAIL CLIENT*) BERBASIS WEB

Selvy Megira
Institut Teknologi dan Bisnis (ITBis) Lembah Dempo
selvymegi96@gmail.com

ABSTRAK

Bertukar informasi adalah hal umum yang hampir setiap orang pernah melakukannya, salah satu media komunikasi yang dapat digunakan adalah email. Karena kemudahan pengaksesan media komunikasi oleh semua orang, membawa dampak bagi keamanan informasi yang menggunakan media tersebut. Kemudian banyak orang berusaha mencari cara bagaimana mengamankan informasi dalam melakukan pertukaran informasi. salah satu caranya adalah dengan menggabungkan aplikasi email dan kriptografi yang menggunakan algoritma kunci simetri, dimana kunci enkripsi dan dekripsi tersebut sama namun jika kunci dekripsi dikirimkan secara terpisah hal ini memungkinkan kunci dapat diketahui oleh penyadap. Pada tugas akhir ini dirancang suatu aplikasi email dankriptografi, dalam perancangan aplikasi ini menggunakan metode pengembangan sistem SDLC (Software Development Life Cycle) dengan model waterfall. Aplikasi menggunakan algoritma blowfish dengan bahasa pemrograman php, dimana kunci enkripsi dan dekripsi sudah diatur oleh sistem hal ini dilakukan agar informasi kunci tidak dapat diketahui dengan mudah oleh penyadap. Algoritma blowfish merupakan algoritma blok cipher 64 bit dimana kunci blowfish harus dikomputasikan pada saat awal, sebelum dilakukannya proses enkripsi dan dekripsi.

Kata kunci : Email, Kriptografi, Kunci Simetri, Blowfish, Enkripsi, Dekripsi

PENDAHULUAN

Dalam perkembangan teknologi informasi yang sangat cepat, hampir semua hal disajikan dalam bentuk digital dan terkomputerisasi. Kemajuan teknologi yang sangat cepat memberikan banyak kemudahan bagi kehidupan manusia terutama dalam komunikasi. Media komunikasi yang digunakan tentu harus merupakan media yang digunakan semua orang dan mudah dijangkau, contoh media komunikasi yang banyak digunakan adalah telepon, jaringan internet bahkan *email.*

Email memanfaatkan suatu *text area* untuk mengirimkan informasi namun akibat dari banyaknya orang yang menggunakan *email* tersebut sebagai alat penyimpan dan pengirim informasi. Informasi menjadi sangat rentan untuk diketahui oleh orang yang tidak berkepentingan dan tidak sedikit orang yang mencari tahu mengenai informasi tersebut dengan melakukan *hack* ke *email* korban.

Karena itulah dibutuhkan suatu metode yang dapat menjaga kerahasiaan informasi. Kerahasiaan informasi dapat dilakukan dengan

teknik enkripsi terhadap informasi sehingga sulit untuk dipahami dengan kata lain disebut dengan kriptografi. Kriptografi yaitu sebuah seni dan bidang keilmuan penyandian informasi dengan tujuan informasi yang dikirim tidak dibaca oleh orang yang tidak berhak melihatnya dengan kata lain walaupun orang tersebut dapat melihat informasi tersebut namun isi dari informasi tersebut tidak dapat dimengerti oleh orang yang tidak berkepentingan melihatnya.

Aplikasi pengirim *email* telah banyak dikembangkan namun masih sedikit aplikasi yang menggabungkan kriptografi dan email dalam satu aplikasi, dimana pengguna dapat mengirim *email* dan disaat yang sama informasi tersebut diamankan dengan proses enkripsi dan dekripsi. Maka dari itu dibuatlah aplikasi Pengamanan Mail berbasis web yang menggunakan algoritma blowfish yang dapat mengenkripsi dan dekripsi informasi. Diharapkan dengan algoritma ini proses enkripsi dan dekripsi informasi dapat dilakukan dengan lebih cepat dibandingkan dengan algoritma kriptografi asimetris.

METODE PENELITIAN

Algoritma Kriptografi

Definisi *terminology* algoritma adalah urutan langkah-langkah logis untuk menyelesaikan masalah yang disusun secara sistematis. Algoritma kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut Algoritma kriptografi terdiri dari tiga fungsi [2], yaitu :

a. *Enkripsi* : merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiaanya. Pesan asli disebut *plaintext,* yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan sebagai chipper atau kode. Sama halnya dengan kita tidak mengerti akan sebuah kata, maka kita akan melihatnya didalam kamus atau daftar istilah. Beda halnya dengan enkrisi, untuk mengubah teks asli kebentuk tek kode kita menggunakan algoritma yang dapt mengkodekan data yang kita inginkan.

b. *Dekripsi* : merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (teks asli) disebut dengan dekripsi pesan. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan algoritma yang digunakan untuk enkripsi.

c. *Kunci* : yang dimaksud disini adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci dibagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*).

Algoritma Blowfish

a. Sejarah Algoritma Blowfish

Blowfish diciptakan oleh seorang *cryptnalyst* bernama *Bruce Schneiner* Presiden perusahaan *Counterpane Internet Security, Inc* (Perusahaan konsultan tentang kriptgrafi dan keamanan computer) pada tahun 1993 dan dipublikasikan tahun 1994. Blowfish dibuat untuk digunakan pada komputer yang mempunyai *microprocessors* besar atau 32 bit keatas dengan *cache* data yang besar .

Blowfish dirancang dan diharapkan mempunyai kriteria perancangan sebagai berikut:

1) Cepat, blowfish melakukan enkripsi data pada *microprocessors*

32-bit dengan rate 26 *clock cycles* per byte.
2) *Compact,* Blowfish dapat dijalankan pada memori kurang dari 5K.
3) Sederhana, blowfish hanya menggunakan operas-operasi sederhana: penambahan, XOR, dan *lookup table* pada operan 32-bit.
4) Memiliki tingkat keamanan yang bervariasi, panjang kunci yang digunakan oleh blowfish dapat bervariasi dan bisa sampai sepanjang 448 bit.

Dalam penerapannya algoritma ini sering kali menjadi tidak optimal. Dikarenakan strategi implementasi yang tidak tepat. Algoritma blowfish akan lebih optimal jika digunakan untuk aplikasi yang tidak sering berganti kunci, seperti jaringan komunikasi atau enkripsi file otomatis. Selain itu, algoritma ini tidak dapat diterapkan untuk aplikasi yang memiliki memori kecil seperti *smart card* dikarenakan algoritma ini membutuhkan memori yang besar. Keamanan penerapan algoritma blowfish juga ditentukan oleh panjang kunci yang digunakan.

Algoritma blowfish terdiri atas dua bagian, yaitu ekspansi kunci dan enkripsi data.

1) Ekspansi Kunci (Key-expansion)

Berfungsi mengubah kunci (Minimum 32-bit, Maksimum 448-bit) menjadi beberapa array subkunci (*subkey*) dengan total 4168 *byte* (18x 32-bit untuk P-*array* dan 4x256x32-bit untuk S-*box* sehingga totalnya 33344 bit atau 4168 *byte)*. Kunci disimpan dalam K-*array*:

K1, K2, … Kj $1 \leq j \leq 14$

Kunci-kunci ini dibangkitkan (*generate*) dengan menggunakan subkunci yang harus dihitung terlebih dahulu sebelum enkripsi atau dekripsi data. Sub-sub kunci yang digunakan terdiri dari:

P-*array* yang terdiri dari 18 buah 32-bit subkunci, P1, P2,….. P18

S-*box* terdiri dari 4 buah 32-bit, masing-masing memiliki 256 entri :
S1,0, S1,1,….,S1,255
S2,0, S2,1,….,S2,255
S3,0, S3,1,….,S3,255
S4,0, S4,1,….,S4,255

2) Enkripsi Data

Terdiri dari iterasi fungsi sederhana (*feistel network*) sebanyak 16 kali setiap putaran terdiri dari permutasi kunci-*dependent,* subsitusi-kunci dan data-*dependent.* semua operasi adalah penambahan (*addition*) dan XOR pada variable 32-bit. Operasi tambahan lainnya hanyalah empat tabel (*table lookup*) array berindeks untuk setiap putaran.

Kunci blowfish harus dihitungsebelum enkripsi atau dekripsi data karena blowfish menggunakan subkunci yang besar. Blowfish adalah algoritma yang menerapkan jaringan feistel yang terdiri dari 16 putaran (iterasi), masukannya adalah 64-bit elemen data X [3].

b. Enkripsi algoritma blowfish

Kunci blowfish harus dikomputasikan pada saat awal, sebelum pengkomputasian enkripsi dan dekripsi data karena blowfish menggunakan subkunci berukuran besar. Langkah-langkah penghitungan atau pembangkitan subkunci tersebut adalah sebagai berikut:

1) Inisialisasi P-*array* yang pertama dan juga empat S-*box*, berurutan dengan *string* yang telah pasti. Contoh
P1= 0x243f6a88
P2= 0x85a308d3
P3= 0x13198a2e
P4= 0x03707344

Dan seterusnya sampai S-*box* yang terakhir.

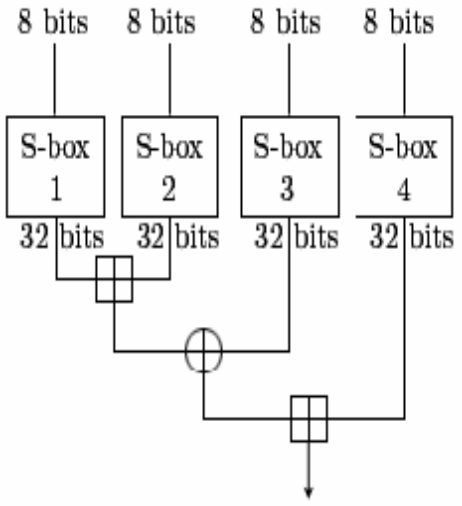
2) XOR-kan P1 dengan 32-bit awal kunci, XOR-kan P2 dengan 32-bit berikutnya dari kunci, dan seterusnya untuk semua bit kunci. Ulangi siklus seluruh bit kunci secara berurutan sampai seluruh P-*array* ter-XOR-kan dengan bit-bit kunci, jika disimbolkan adalah : P1 = P1 ⊕ K1, P2 = P2 ⊕ K2, P3 = P3 ⊕ K3, …. P14 = P14 ⊕ K14, P15 = P15 ⊕ K1, ….. P18 = P18 ⊕ K4.
3) Enkripsikan *string* yang seluruhnya no dengan algoritma blowfish, menggunakan subkunci yang telah didekripsikan pada langkah 1 dan 2.
4) Gantikan P1 dan P2 dengan keluaran dari langkah 3.
5) Enkripsikan keluaran langkah 3 dengan langkah 2 kembali menggunakan algoritma blowfish dengan subkunci yang berbeda (sebab langkah 2 menghasilkan subkunci baru).
6) Gantikan P3 dan P4 dengan keluaran dari langkah 5
7) Lakukan seterusnya, gantikan seluruh elemen P-*array* dan kemudian keempat S-*box* secara berurutan, dengan hasil keluaran algoritma blowfish yang terus-menerus berubah.

Total keseluruhan, terdapat 521 iterasi untuk menghasilkan subkunci-subkunci yang dibutuhkan. Aplikasi hendaknya menyimpannya dibandingkan menghasilkan ulang subkunci-subkunci tersebut. Pada proses enkripsi, blowfish memiliki 16 iterasi, masukannya adalah 64-bit elemen data X. berikut adalah langkah-langkah untuk melakukan proses enkripsi:

1) Bagi X menjadi dua bagian yang masing-masing terdiri dari 32-bit: X_L, X_R.
2) Lakukan langkah berikut

   For i = 1 to 16
   $X_L = X_L \oplus P_i$
   X_R =F(X_L) ⊕ X_R
   Tukar X_L dan X_R
3) Setelah iterasi ke-16, tukar X_L dan X_R untuk melakukan undo pertukaran terakhir.
4) Lalu lakukan
   a$X_R = X_R \oplus P_{17}$
   A$X_L = X_L \oplus P_{18}$
5) Terakhir, gabungkan kembali X_L dan X_R untuk mendapatkan *ciphertext*.

Dilangkah kedua, telah dituliskan mengenai penggunaan fungsi F. Fungsi F adalah: bagi X_L menjadi empat bagian 8-bit: a, b, c, dan d.
F(X_L) ((S1, a + S2, b mod 2^{32}) ⊕ S3, c) + S4, d mod 2^{32}.Supaya lebih paham dengan fungsi F, tahapannya dapat dilihat pada gambar berikut ini:



Gambar 1: Skema Fungsi F

Pada algoritma blowfish, proses dekripsi dilakukan dengan urutan yang sama persis dengan proses enkripsi. Namun pada proses dekripsi P1, P2, …., P18 digunakan dalam urutan yang terbalik.

c. Dekripsi algoritma blowfish

Tahapan dekripsi sama dengan enkripsi, kecuali P1, P2,….P18 digunakan dengan urutan berbalik

(*reverse*). Dekripsi algoritma blowfish bersifat maju kedepan, yang menyebabkan dekripsi bekerja dalam arah algoritma yang sama seperti halnya dengan enkripsi akan tetapu sebagai masukannya adalah *chipertext.* Walaupun begitu, seperti yang diharapkan subkunci yang digunakan dalam urutan terbalik. Algoritmanya dapat dinyatakan sebagai berikut:

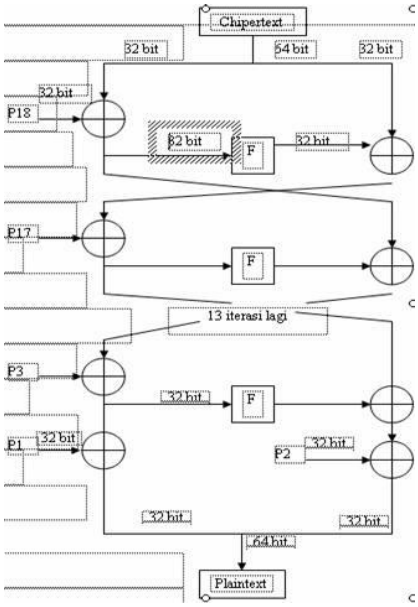
For i = 1 to 16 do

$X_{Ri} = X_{Li} \oplus P_{19}$ - i;

X_{Li} =F$[X_{Ri}] \oplus X_{Ri}$ -1;

$X_{L17} = X_{R16} \oplus P_1$;

$X_{L17} = X_{L16} \oplus P_2$;

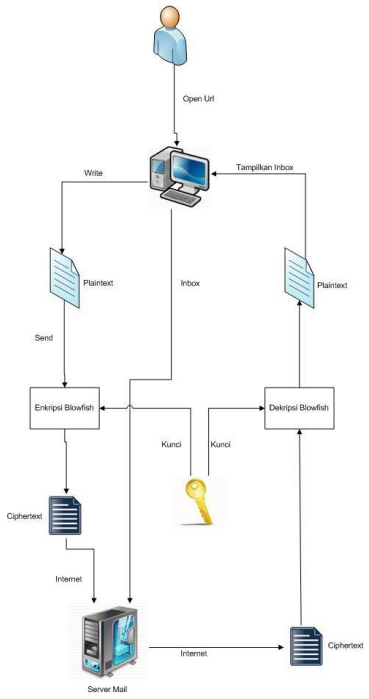


Gambar 2 : Diagram Skema Dekripsi Algoritma Blowfish

HASIL DAN PEMBAHASAN

Rancangan Sistem

Dibawah ini merupakan *Rich Picture* penerapan pengamanan surat elektronik berbasis web.



Gambar 3 : Ilustrasi Penerapan Algoritma Blowfish.

Rancangan Program

Program yang dibuat terdiri dari enam buah Form, yang terdiri dari Form Menu Login, Form Menu Utama, Form Menu Write, Form Menu Inbox, Form Menu Help dan Form Menu About serta satu *button Logout*. Untuk melakukan pengiriman email pengguna dapat memilih menu *write*, pengguna diharuskan mengisi To, Subject, dan Message dimana panjang subject harus 16 karakter karena *subject itersebut akan diambil sebagai* kunci untuk mengenkripsi data atau message dari email tersebut sebelum dikirimkan ke alamat email tujuan. Sedangkan untuk membaca email, pengguna dapat memilih menu Inbox kemudian memilih dan mengklik *subject* yang akan dibaca, sama seperti proses write, subject berperan sebagai kunci untuk mendekripsi isi dari *message* tersebut

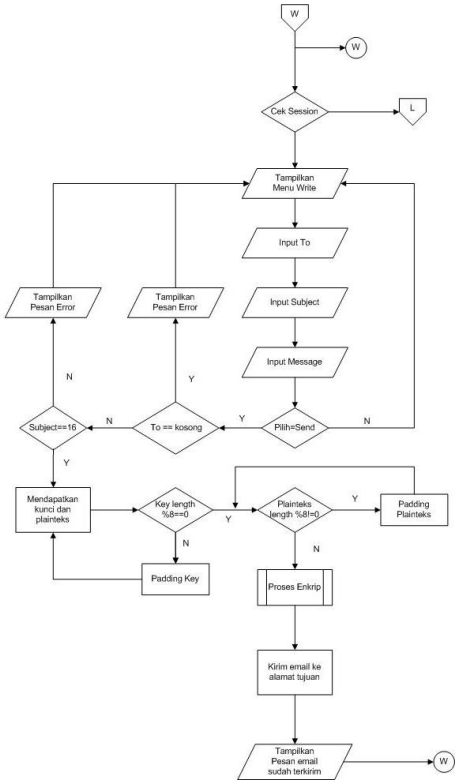
sebelum ditampilkan ke layar pengguna dalam bentuk aslinya.

Flowchart Program

Berikut ini adalah flowchart yang digunakan untuk mengetahui proses program pada aplikasi pengamanan surat elektronik yang menerapkan algoritma blowfish.

a. Flowchart Form Write

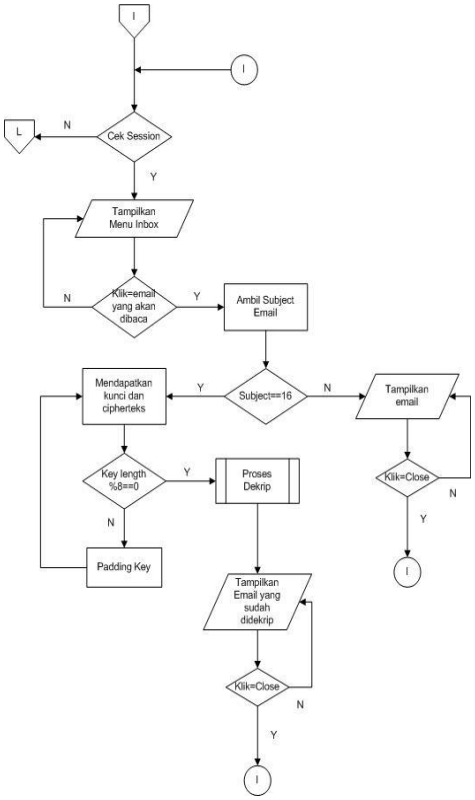
Pada *flowchart* menu *write*, pengguna harus mengisi alamat email tujuan atau *To, Subject* dan Message kemudian informasi tersebut dilakukan proses enkrip terlebih dahulu sebelum ke alamat tujuan.



Gambar 4 : Flowchart Menu Write

b. Flowchart Form Inbox

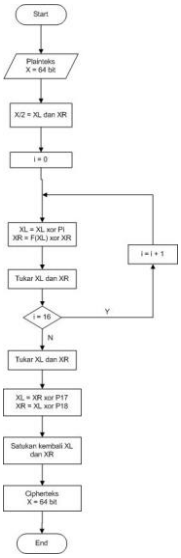
Dalam *flowchart* menu *inbox,* pengguna dapat membaca *inbox* atau kotak masuk dengan cara memilih *email* yang akan dibaca lalu kemudian mengklik *subject email* yang akan dibaca.



Gambar 5 : Flowchart Menu Inbox

c. Flowchart Proses Enkripsi

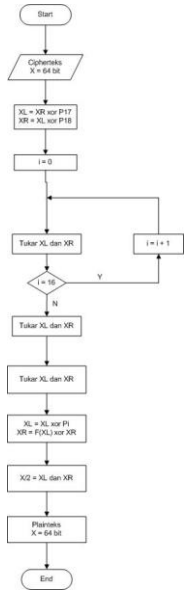
Pada *flowchart* proses enkripsi merupakan alur jalannya enkripsi email sebelum dikirim ke alamat tujuan.



Gambar 6 : Flowchart Proses Enkripsi

d. Flowchart Proses Dekripsi

Pada *flowchart* proses dekripsi menjelaskan proses pengembalian data dari *chipertext* ke *plaintext*.

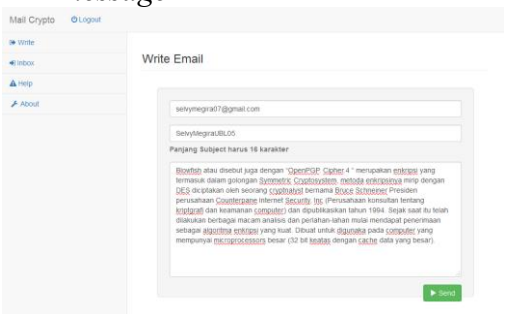


Gambar 7 : Flowchart proses dekripsi

Tampilan Layar Program

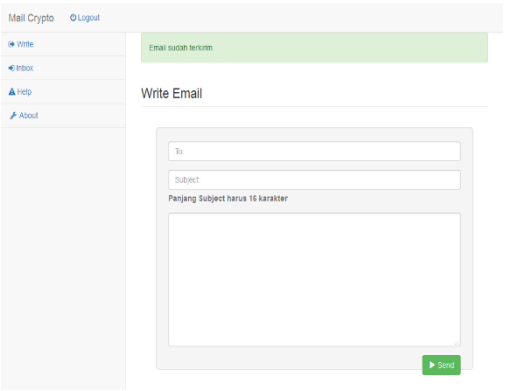
a. Tampilan layar Menu Write

Fungsi dari menu *write* adalah untuk mengirimkan email ke alamat tujuan. Jika pengguna tidak mengiisi alamat email maka email tidak akan terkirim dan apabila pengguna tidak mengisi panjang *subject* sebanyak 16 karakter maka email tidak akan terkirim. Pada layar menu *write* isi dari *message* tersebut akan dirubah terlebih dahulu dari *plaintext* ke *ciphertext* dengan menggunakan kunci yang didapat dari subject setelah *message* tersebut dienkripsi atau telah menjadi *chipertext* maka pesan tersebut baru akan dikirim ke alamat tujuan. *Subject* berperan sebagai kunci untuk megenkripsi isi dari *message*.



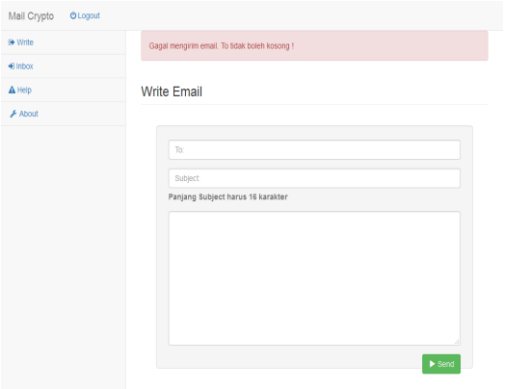
Gambar 8 : Tampilan Layar Menu Write

Saat email berhasil dikirim akan ada pesan pemberitahuan “email berhasil dikirim”.



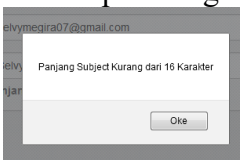
Gambar 9 : Tampilan Layar Email sudah terkirim

Saat pengguna tidak mengisi alamat email tujuan atau To, dan sudah mengklik *button send* maka akan muncul pemberitahuan “gagal mengirim email, To tidak boleh kosong”.

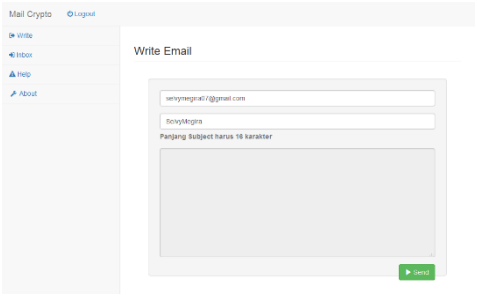


Gambar 10 : Tampilan Layar Error To

Apabila pengguna belum mengisi subject sampai 16 karakter maka pengguna tidak dapat mengisi *message*.



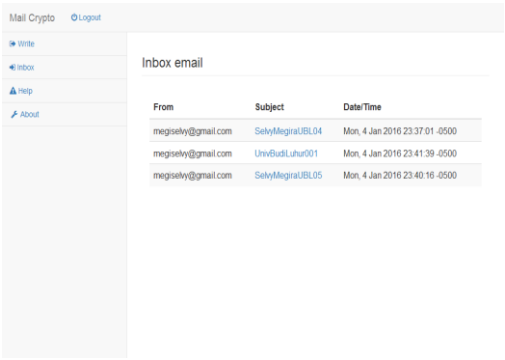
Gambar 11 :Tampilan Layar Error Panjang Subject



Gambar 12 : Tampilan layar message tidak bisa diinput

d. Tampilan Layar Menu Inbox

Pada tampilan layar ini, pengguna dapat membaca *email* yang akan dibaca dengan cara memilih dan mengklik subject *email* namun sebelum dibaca *email* tersebut akan didekripsikan terlebih dahulu.



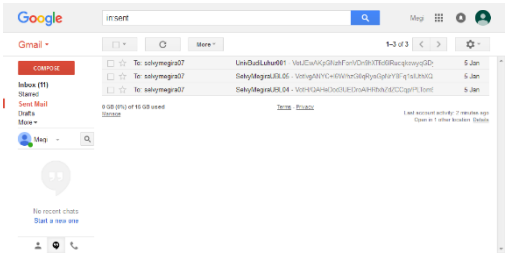
Gambar 12 : Tampilan layar menu Inbox

Berikut adalah tampilan layar read inbox, setelah pengguna mengklik subject yang akan dibaca :

Gambar 13 : Tampilan layar baca inbox

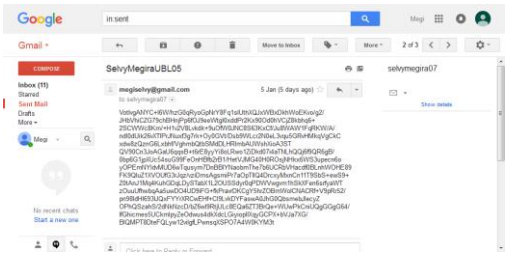
e. Tampilan Layar bukti pesan telah dikirim melalui aplikasi mail crypto

Pada tampilan layar ini, merupakan tampilan layar yang membuktikan bahwa isi dari email tersebut telah dienkripsi ke email penerima.



Gambar 14 : Tampilan Layar bukti pesan telah dikirim melalui aplikasi mail crypto

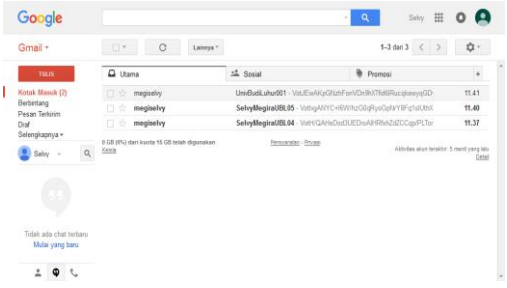
Berikut adalah tampilan jika pengguna sudah membaca *outbox email,* dimana pesan tersebut sudah dienkrip:



Gambar 15 : Tampilan layar berhasil dienkripsi

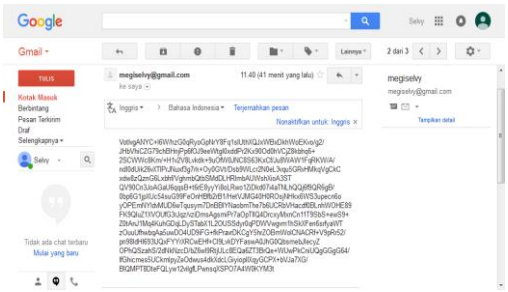
f. Tampilan layar bukti telah diterima di gmail

Pada tampilan layar ini, adalah bukti/hasil bahwa pengguna telah menerima pesan dari aplikasi Mail Crypto yang telah dienkripsi.



Gambar 16 : Tampilan layar email berhasil diterima

Berikut adalah tampilan layar ketika pengguna ingin membaca pesan digmail yang sudah terenkripsi.



Gambar 17 : Tampilan layar inbox berhasil dienkripsi

4.2 Pengujian Program

Pengujian aplikasi dilakukan menggunakan *Notebook Acer Aspire One*. Berikut rincian dan gambaran pengujian aplikasi:

a. Pengujian Proses Enkrips

Tabel 1 : Pengujian proses enkripsi

Plainteks	Kunci	Cipherteks	Pengiriman Email
Skripsi 2015 Universitas Budi Luhur	Selvy MegiraUBL01	VotOrgAK1VSz30cnvodc5dnoO0fT7kM1H+/31QW77QNzMRjJdMnWHXqNmhPJ2Zxw	Berhasil Terkirim
Blowfish atau disebut juga dengan "OpenPGP. Cipher.4 "	UnivBudiLuhur003	VotZDgANMn9ZBS6QbyQ6n7m+kvOsEaJmEvo9H9yD7b8iaWH5jcMFFiSCkAD+IvYFIWW1vnWlkjj1nXU0ilevNinI3HYWWOgd	Berhasil Dikirim
32 bit keatas dengan cache data yang besar.	UnivBudiLuhur004	VotZNwAK3ig6YT9iccFUDlSKtR+r2wzVy9OmF2TiQp3Sy2nrsVA32gypYTDxVXo01aXsxLZM/AI=	Berhasil Dikirim
18x 32-bit untuk P-array dan 4x256x32-bit untuk S-box	UnivBudiLuhur005	VotZ6AAH2EWdC9AMBjxY8WBfrUi0s76zTJPL0LHfOQu5oRFxPtpv7HBBKktZF4csZQF8llx4TdsNBbneInH4QQ==	Berhasil Dikirim
S-box terdiri dari 4 buah 32-bit, masing-masing memiliki 256 entri : S1,0, S1,1,….,S1,255 S2,0, S2,1,….,S2,255 S3,0, S3,1,….,S3,255 S4,0, S4,1,….,S4,255	UnivBudiLuhur006	VotaEgABlEdEOFPEFsl1Ogir1U7GYSIYQDArb7pPmw+6HgQNEUq9HtHf+XegFKwjtgFNk7SnjrNoYzMyAxiAmveb+2FQwj2No/M73p8bOl9sfL3m2/HfGWdt0o2YhcwgrZX5Y6oIYkv8u9IH8SNZBMdr6zmCObc4hRR+/75hED+jYEh/APWKDtyPoratTc3ACG2iDLkjLkP46VK4TkvmBoR6Iv/i8Zesw6uqsUD2CSk=	Berhasil Dikirim

b. Pengujian Proses Dekripsi

Tabel 2 : Pengujian Proses Dekripsi

Cipherteks	Kunci	Plainteks	Pengiriman Email
VotOrgAK1VSz30cnvodc5dnoO0fT7kM1H+/31QW77QNzMRjJdMnWHXqNmhPJ2Zxw	Selvy MegiraUBL01	Skripsi 2015 Universitas Budi Luhur	Berhasil Diterima
VotZDgANMn9ZBS6QbyQ6n7m+kvOsEaJmEvo9H9yD7b8iaWH5jcMFFiSCkAD+IvYFIWW1vnWlkjj1nXU0ilevNinI3HYWWOgd	UnivBudiLuhur003	Blowfish atau disebut juga dengan "OpenPGP. Cipher.4 "	Berhasil Diterima
VotZNwAK3ig6YT9iccFUDlSKtR+r2wzVy9OmF2TiQp3Sy2nrsVA32gypYTDxVXo01aXsxLZM/AI=	UnivBudiLuhur004	32 bit keatas dengan cache data yang besar.	Berhasil Diterima
VotZ6AAH2EWdC9AMBjxY8WBfrUi0s76zTJPL0LHfOQu5oRFxPtpv7HBBKktZF4csZQF8llx4TdsNBbneInH4QQ==	UnivBudiLuhur005	18x 32-bit untuk P-array dan 4x256x32-bit untuk S-box	Berhasil Diterima
VotaEgABlEdEOFPEFsl1Ogir1U7GYSIYQDArb7pPmw+6HgQNEUq9HtHf+XegFKwjtgFNk7SnjrNoYzMyAxiAmveb+2FQwj2No/M73p8bOl9sfL3m2/Hf	UnivBudiLuhur006	S-box terdiri dari 4 buah 32-bit, masing-masing memiliki 256 entri :	Berhasil Diterima

GWdt0o2YhcwgrZX5 Y6oIYkv8u9IH8SNZ BMdr6zmCObc4hRR +/75hED+jYEh/APW KDtyPoratTc3ACG2i DLkjLkP46VK4Tkvm BoR6Iv/i8Zesw6uqsU D2CSk=		S1,0, S1,1,….,S1 ,255 S2,0, S2,1,….,S2 ,255 S3,0, S3,1,….,S3 ,255 S4,0, S4,1,….,S4 ,255	

SIMPULAN DAN SARAN

Kesimpulan

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

a. Meminimalisir kemungkinan kebocoran pesan yang terdapat di akun *email* apabila menjadi korban dari *hacker*, karena pesan tersebut sudah dienkripsi.
b. Diharapkan aplikasi ini dapat menjadi sebuah solusi dari kekhawatiran masyarakat terhadap keamanan data yang terdapat di akun *email*.
c. Aplikasi ini telah diatur oleh sistem sehingga isi pesan atau data yang terkandung dalam akun akun *email* tersebut otomatis telah dienkripsi.

Saran

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

a. Aplikasi ini dapat dikembangkan lagi dengan melakukan penambahan metode lain sehingga menjadi pengamanan ganda.
b. Aplikasi diharapkan dapat dikembangkan lagi sehingga dapat mengenkrip format data digital seperti gambar(.*jpg, *.jpeg, *.png, dll) , file (doc, pdf, xls, ppt dll) dan suara (*.mp3, *.mp4, dll) sehingga dapat mengirim dalam bentuk *attachment file*.
c. Interface masih sangat sederhana diharapkan ditambah beberapa fitur seperti *outbox*, draft dan lain sebagainya.

DAFTAR PUSTAKA

[1] Pardosi, Mico. 2006. *E-mail Gratis Yahoo! Indonesia*. Surabaya : Dua Selaras
[2] Ariyus, Donny. 2008. *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi*. Yogyakarta : C.V Andi OFFSET
[3] Schneier, Bruce. 1996. *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C*. John Wiley and Sons.