IMPLEMENTASI KRIPTOGRAFI MENGGUNAKAN ALGORITMA GOST (*GOSUDARSTEVVENYI STANDARD*) UNTUK PENGIRIMAN *E-MAIL* PADA APLIKASI CIRUS-MAIL BERBASIS WEB

Sri Gustaria[1], Titin Fatimah[2]
[1,2]Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5866369
E-mail : gustaria96@gmail.com[1], titin.fatimah@budiluhur.ac.id[2]

Abstrak - Bertukar informasi adalah hal umum yang hampir setiap orang pernah melakukannya, salah satu media komunikasi yang dapat digunakan adalah e-mail. Di era modern ini, keamanan data dalam pertukaran informasi melalui e-mail merupakan hal yang sangat penting. Karena teks pesan yang dikirim terkadang adalah pesan rahasia dan pribadi, sehingga kerahasiaan pesan menjadi sangat penting. Berbagai cara dilakukan orang untuk mendapatkan data dan informasi, meskipun dengan cara mencuri dan lalu mengubah isi dari informasi tersebut. Pada penelitian ini dibuat suatu aplikasi menggunakan teknik kriptografi yang dapat digunakan untuk mengamankan informasi yang dikirim melalui e-mail. Algoritma yang digunakan adalah algoritma GOST (Gosudarstvennyi Standard). Algoritma GOST merupakan algoritma simetris blok cipher 64 bit yang dikenal cukup aman karena memiliki 32 putaran dalam proses enkripsi dan dekripsi. Bahasa pemrograman yang digunakan adalah PHP. Pada aplikasi ini kunci enkripsi dan dekripsi diatur oleh sistem, hal ini dilakukan agar informasi kunci tidak dapat diketahui dengan mudah oleh pihak yang tidak berkepentingan. Berdasarkan implementasi dan pengujian program, dapat disimpulkan bahwa aplikasi ini mudah digunakan, pesan yang dikirim dan diterima melalui aplikasi ini aman karena sudah melalui proses enkripsi terlebih dahulu.
Kata kunci - Kriptografi, *E-mail*, Algoritma GOST

1. PENDAHULUAN

Perkembangan teknologi informasi berkembang pesat saat ini, bukan hanya di negara maju, di negara berkembang pun terjadi peningkatan terhadap penggunaan komputer baik dalam aktifitas bisnis, pendidikan, dan bidang-bidang yang lain. Salah satu perkembangan komputer yang cukup pesat di Indonesia akhir-akhir ini adalah penggunaan internet. Salah satu fitur internet yang banyak dimanfaatkan adalah *e-mail* atau surat elektronik. Dengan menggunakan *e-mail,* pesan menjadi lebih mudah dan lebih cepat tersampaikan bahkan dalam hitungan detik.

Masalah keamanan dan kerahasiaan merupakan salah satu aspek penting dari suatu data, pesan dan informasi. Pengiriman suatu pesan, data dan informasi yang sangat penting membutuhkan tingkat keamanan yang tinggi. Dengan perkembangan teknologi informasi sekarang ini, dimana setiap orang akan mudah untuk mendapatkan suatu pesan, data dan informasi. Berbagai cara dilakukan orang untuk mendapatkan data dan informasi tersebut. Mulai dari tingkatan yang mudah sampai kepada cara-cara yang lebih rumit. Dan dengan berbagai cara pula orang berusaha untuk melindungi pesan, data atau informasi tersebut agar tidak dapat diketahui oleh orang yang tidak memiliki hak atas data tersebut. Pengamanan informasi dapat dilakukan dengan proses enkripsi dan proses dekripsi terhadap pesan atau informasi yang dapat disebut juga dengan kriptografi. Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara meyandikannya kedalam bentuk yang tidak dapat dimengerti lagi maknanya.

Oleh karena itu dibuat suatu aplikasi penyandian informasi atau pesan dengan memanfaatkan media *e-mail* berbasis web menggunakan algoritma GOST *(Gosudarstvennyi Standard)*. Aplikasi ini dibuat dengan tahapan perancangan sistem *SDLC (Software Development Life Cycle)* yaitu metode *waterfall.* Bahasa pemrograman yang digunakan yaitu *PHP* (*Hypertext Preprocessor)*.

Adapun permasalahan yang dirumuskan adalah sebagai berikut :

a. Bagaimana mengimplementasikan algoritma GOST ke dalam aplikasi pengiriman *e-mail*?
b. Bagaimana cara melakukan pengamanan terhadap informasi atau pesan yang dikirim atau diterima melalui media *e-mail* sehingga informasi tersebut bisa terjaga keamanannya?

Tujuan yang dapat diambil dari pembuatan aplikasi ini adalah sebagai berikut :

a. Mengembangkan suatu aplikasi pengamanan data menggunakan algoritma kriptografi GOST untuk mengamankan informasi atau pesan yang sifatnya rahasia.
b. Mengamankan *file* dan pesan yang dikirim atau diterima melalui *e-mail* agar tidak dapat diketahui oleh orang yang tidak bertanggung jawab.
c. Menghasilkan aplikasi *e-mail* yang menggunakan teknik kriptografi yang

diharapkan mudah dimengerti dan digunakan oleh pengguna.

Pada aplikasi ini, file yang dapat dienkripsi dan didekripsi hanya *file* dengan ekstensi: *.txt, *.doc, *.docx, *pdf, *.ppt, *.pptx, *.xls, dan *.xlsx.

2. LANDASAN TEORI

2.1. Algoritma Kriptografi

Algortima kriptografi merupakan langkah-langkah logis untuk menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut [1]. Di dalam kriptografi terdapat beberapa istilah atau terminologi, yaitu sebagai berikut:

1) Pesan, Plainteks, dan Cipherteks

Pesan (*message*) adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah plainteks (*plaintext*) atau teks-jelas (*cleartext*). Pesan dapat berupa data atau informasi yang dikirim (melalui kurir, saluran telekomunikasi, dsb) atau yang disimpan di dalam media perekaman (kertas, *storage*, dsb). Pesan yang tersimpan tidak hanya berupa teks, tetapi juga dapat berbentuk citra (*image*), suara/bunyi (audio) dan video, atau berkas biner lainnya. Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan ke bentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut cipherteks (*ciphertext*) atau kriptogram (*cryptogram*). Cipherteks harus dapat ditransformasikan kembali menjadi plainteks semula agar pesan yang diterima bisa dibaca.

2) Pengirim dan penerima

Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan. Entitas di sini dapat berupa orang, mesin (komputer), kartu kredit, dan sebagainya. Jadi, orang bisa bertukar pesan dengan orang lainnya. Pengirim tentu menginginkan pesan dapat dikirim secara aman, yaitu ia yakin bahwa pihak lain tidak dapat membaca isi pesan yang ia kirim. Solusinya adalah dengan cara menyandikan pesan menjadi cipherteks.

3) Enkripsi dan Dekripsi

Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) atau *enciphering* (standar nama menurut ISO 7498-2). Sedangkan proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi (*decryption*) atau *deciphering* (standar nama menurut ISO 7498-2). Enkripsi dan dekripsi dapat diterapkan baik pada pesan yang dikirim maupun pada pesan tersimpan. Istilah *encryption of data in motion* mengacu pada enkripsi pesan yang ditransmisikan melalui saluran komunikasi, sedangkan istilah *encrypton of data at-rest* mengacu pada enkripsi dokumen yang disimpan di dalam *storage*. Contoh *encryption of data inmotion* adalah pengiriman nomor PIN dari mesin ATM ke komputer *server* di kantor bank pusat.

Contoh *encryption of data at-rest* adalah enkripsi *file* basis data di dalam *hard disk*.

4) Cipher dan Kunci

Algoritma kriptografi disebut juga *cipher* yaitu aturan untuk *enchipering* dan *dechipering*, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk *enciphering* dan *deciphering*. Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yaitu himpunan yang berisi elemen-elemen plainteks dan himpunan yang berisi cipherteks. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara kedua himpunan tersebut. Kriptografi modern mengatasi masalah keamanan data dengan penggunaan kunci, yang dalam hal ini algoritma tidak lagi dirahasiakan, tetapi kunci harus dijaga kerahasiaannya. Kunci (*key*) adalah parameter yang digunakan untuk transformasi *enciphering* dan *dechipering.* Kunci biasanya berupa *string* atau deretan bilangan.

5) Sistem Kriptografi

Kriptografi membentuk sebuah sistem yang dinamakan sistem kriptografi. Sistem kriptografi (*cryptosystem*) adalah kumpulan yang terdiri dari algoritma kriptografi, semua plainteks dan cipherteks yang mungkin, dan kunci. Di dalam sistem kriptografi, *cipher* hanyalah salah satu komponen saja.

6) Penyadap

Penyadap (*eavesdropper*) adalah orang yang mencoba menangkap pesan selama ditransmisikan. Tujuan penyadap adalah untuk mendapatkan informasi sebanyak-banyaknya mengenai sistem kriptografi yang digunakan untuk berkomunikasi dengan maksud untuk memecahkan cipherteks. Nama lain penyadap: *enemy*, *adversary*, *intruder*, *interceptor*, *bad guy.* Ron Rivest, seorang pakar kriptografi, menyatakan bahwa *cryptography is about communication in the presence of adversaries* (Kriptografi adalah perihal berkomunikasi dengan keberadaan pihak musuh).

7) Kriptanalis dan Kriptologi

Kriptografi berkembang sedemikian rupa sehingga melahirkan bidang yang berlawanan yaitu kriptanalisis. Kriptanalisis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan chiperteks menjadi plainteks tanpa mengetahui *kunci* yang digunakan. Pelakunya disebut kriptanalis. Jika seorang kriptografer (*cryptographer*) mentransformasikan plainteks menjadi cipherteks

dengan suatu algoritma dan kunci maka sebaliknya seorang kriptanalis berusaha untuk memecahkan cipherteks tersebut untuk menemukan plainteks atau kunci. Kriptologi (*cryptology*) adalah studi mengenai kriptografi dan kriptanalisis. Baik kriptografi maupun kriptanalisis keduanya saling berkaitan [3].

2.2. Algoritma GOST

Algoritma GOST merupakan blok kode dari bekat Uni Soviet, yang merupakan singkatan dari *Gosudarstvennyi Standard* atau standar pemerintah. GOST merupakan blok kode 64 bit dengan panjang kunci 256 bit. Algoritma ini mengiterasi algoritma enkripsi sederhana sebanyak 32 putaran.

Untuk mengenkripsi pertama-tama plainteks 64 bit dipecah menjadi 32 bit bagian kiri, L dan 32 bit bagian kanan, R. Subkunci (subkey) untuk putaran i adalah Ki. Pada satu putaran ke-i operasinya adalah sebagai berikut [1]:

Li = R i-1

Ri = L i-1 xor f(R i-1, Ki)

1. Proses Pembangkitan Kunci

Kunci internal pada algoritma GOST dibangkitkan dari kunci eksternal yang diberikan oleh pengguna [3]. Pembangkitan kunci internal dilakukan dengan membagi kunci eksternal 256 bit (k1, k2, k3, k4, …, k256) ke dalam delapan bagian yang masing-masing panjangnya 32 bit. Pembagiannya adalah sebagai 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)

2. Proses Enkripsi

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

1) 64 bit plainteks dibagi menjadi 2 buah bagian 32 bit, yaitu L_i dan R_i.
   Caranya :
   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) $(R_i + K_i) \bmod 2^{32}$. Hasil dari penjumlahan modulo 2^{32} berupa 32 bit.
3) Hasil dari penjumlahan modulo 2^{32} dibagi menjadi 8 bagian, dimana masing-masing 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.

Tabel 1 : 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 didapat dari substitusi ke S-Box kemudian digabungkan kembali menjadi 32 bit dan kemudian 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 dilakukan proses

Langkah nomor 2 sampai 6 dilakukan sebanyak 32 kali (putaran). Pada langkah nomor 2 penggunaan kunci dijadwalkan penggunaannya sesuai dengan putarannya.

Tabel 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 sedikit berbeda. Langkah 5 dan 6 untuk putaran 31 adalah sebagai berikut :

R32 = R31 sebelum dilakukan proses
L32 = L31 XOR R31
Sehingga cipherteks yang dihasilkan adalah,
L32 : b(32), b(31), ….. ,b(1)
R32 : a(32), a(31), ….. ,a(1)
Chiperteks = a(1), ….. ,a(32); b(1), ….. ,b(32).

3. Proses Dekripsi

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

Tabel 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	K7	K6	K5	K4	K3	K2	K1	K0
Putaran	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
Kunci Internal	K7	K6	K5	K4	K3	K2	K1	K0
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 sebagai berikut :

R32 = R31 sebelum dilakukan proses
L32 = L31 XOR R31

Sehingga, plainteks yang dihasilkan pada proses dekripsi adalah [4],
L32 : b(32), b(31), ….. ,b(1)
R32 : a(32), a(31), ….. ,a(1)
Plainteks = a(1), ….. ,a(32); b(1), ….. ,b(32).

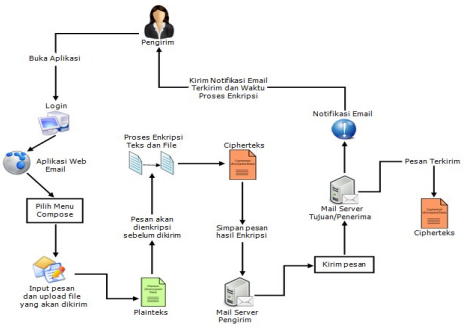
2.3. *E-mail*

E-mail merupakan singkatan dari *Electronic Mail* (surat elektronik) adalah fasilitas dalam internet untuk surat-menyurat. Dengan adanya *e-mail* berkirim surat dan berbagi informasi kepada keluarga dan sahabat menjadi lebih mudah, cepat dan hemat karena tidak memerlukan biaya yang terlalu mahal jika dibandingkan dengan berkirim surat melalui Pos atau jasa pengiriman lainnya. *E-mail* yang dikirimkan pun bukan hanya berupa teks saja, tetapi dapat berupa file gambar, musik, film, dan *file* dokumen lainnya [5].

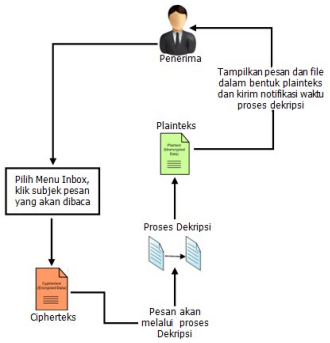
3. RANCANGAN SISTEM DAN APLIKASI

3.1. Alur Aplikasi

Berikut ini adalah gambaran alur kerja aplikasi untuk proses pengiriman dan penerimaan *email* :



Gambar 1 : Alur Proses Pengiriman Email



Gambar 2 : Alur Proses Penerimaan Email

3.2. Algoritma Program

a. Algoritma Proses Enkripsi GOST

Algoritma ini menjelaskan tentang proses enkripsi dan merupakan kelanjutan dari halaman *compose*.

```
1.  Start
2.      Ambil Subject
3.      Proses pembangkitan kunci
4.      Bagi kunci menjadi k1 sampai k7
    masing-masing 32 bit
5.      Simpan ke dalam table Penjadwalan
    Kunci Enkripsi
6.      Ambil Plainteks
7.      Hitung panjang Plainteks
8.      Simpan Plainteks menjadi blok-blok 64
    bit ke dalam array
9.      i = 0
10.     If i < 32
11.             Pi = (Ri+Ki) mod 2^32
12.             Substitusikan dengan table Sbox
13.             Bagi menjadi 8 bagian
14.             RLS (Rotate Left Shift) 11 bit ke
    kiri
15.             Ri+1 = RLS XOR Li
                Li + 1 = Ri sebelum proses
16.             i ++
17.             Kembali ke baris 6
        Else
18.             lakukan pembalikan string Li dan
    Ri
19.             RL [i] = Ri dan Li
20.         Cipherteks = RL [i]
21.     End If
22. End
```

b. Algoritma Proses Dekripsi GOST

Algoritma ini menjelaskan tentang proses dekripsi dan merupakan kelanjutan dari halaman *inbox*.

```
1.  Start
2.      Ambil Subject
3.      Proses pembangkitan kunci
4.      Bagi kunci menjadi k1 sampai k7
    masing-masing 32 bit
5.      Simpan ke dalam table Penjadwalan
    Kunci Enkripsi
6.      Ambil Cipherteks
7.      Hitung panjang Cipherteks
8.      Simpan Cipherteks menjadi blok-blok
    64 bit ke dalam array
9.      i = 0
10.     If i < 32
11.             Pi = (Ri+Ki) mod 2^32
12.             Substitusikan dengan table Sbox
13.             Bagi menjadi 8 bagian
14.             RLS (Rotate Left Shift) 11 bit ke
    kiri
15.             Ri+1 = RLS XOR Li
                Li + 1 = Ri sebelum proses
16.             i ++
17.             Kembali ke baris 6
        Else
18.             lakukan pembalikan string Li dan
    Ri
19.             RL [i] = Ri dan Li
20.         Plainteks   = RL [i]
21.     End If
22. End
```

c. Algoritma Proses Enkripsi *File*

Algoritma ini menjelaskan tentang proses enkripsi *file* dan merupakan kelanjutan dari halaman *compose*.

```
1.  Start
2.      Buka isi file
3.      Ambil isi file
4.      Hitung panjang isi file
5.      If Isi panjang isi file <= 34
6.          Jalankan proses Enkripsi base64
7.          Jalankan proses Enkripsi Gost
8.          Cipherteks = Cipher_Gost
9.      Else
10.         Jalankan proses Enkripsi base64
11.         Jalankan proses Enkripsi Gost
12.         Cipherteks = Cipher_Gost + Cipher_Base64
13.     End If
14. End
```

d. Algoritma Proses Dekripsi *File*

Algoritma ini menjelaskan tentang proses dekripsi *file* dan merupakan kelanjutan dari halaman *inbox*.

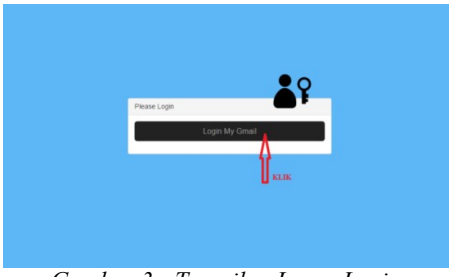
```
1.  Start
2.      Ambil isi Cipherteks
3.      Hitung panjang Cipherteks
4.      If Isi panjang Cipherteks <= 34
5.          Jalankan proses Dekripsi Gost
6.          Jalankan proses Dekripsi Base64
7.          Plainteks = Plain_Gost
8.      Else
9.          Jalankan proses Dekripsi Gost
10.         Jalankan proses Dekripsi Base64
11.         Plainteks = Plain_Gost + Plain_Base64
12.     End If
13. End
```

4. HASIL DAN PEMBAHASAN

4.1. Tampilan Layar Program

a. Tampilan Layar *Login*

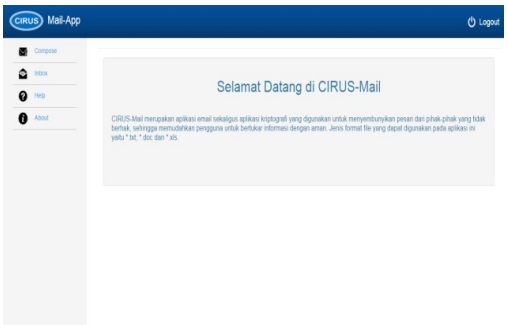
Halaman *login* ini akan tampil pertama kali ketika aplikasi dijalankan yang menjadi penghubung ke halaman utama.



Gambar 3 : Tampilan Layar Login

b. Tampilan Layar Halaman Utama

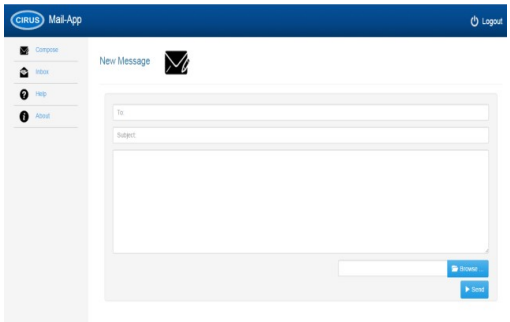
Halaman utama ini akan tampil ketika *user* sudah berhasil melakukan *login*. Pada halaman ini terdapat beberapa menu yang dapat dipilih *user*, yaitu menu *compose*, menu *inbox*, menu *help*, dan menu *about*.



Gambar 4 : Tampilan Layar Halaman Utama

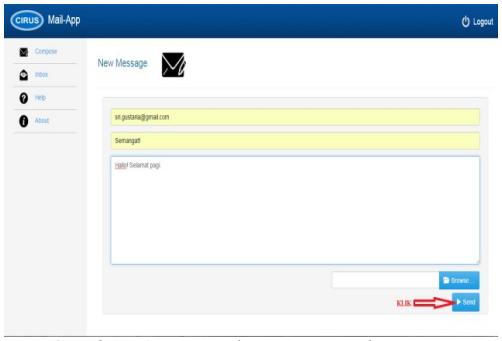
c. Tampilan Layar *Compose*

Halaman *compose* ini adalah halaman yang digunakan *user* untuk mengirimkan *email* menggunakan aplikasi ini.



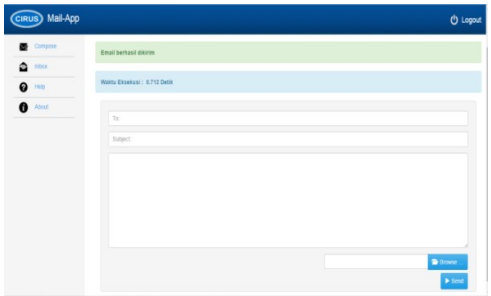
Gambar 5 : Tampilan Layar Compose

User harus mengisi alamat e-mail tujuan, *subject* pesan dan isi pesan yang akan dikirimkan, serta melampirkan *file* jika diperlukan.



Gambar 6 : Tampilan Layar Tulis Pesan

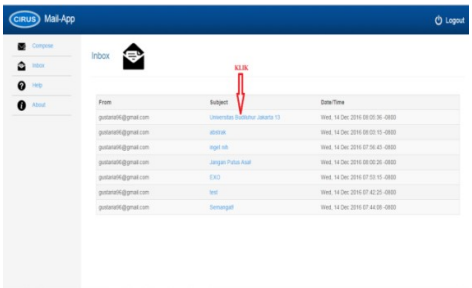
Saat *e-mail* berhasil dikirim akan ada pesan pemberitahuan, *e-mail* berhasil dikirim dan waktu eksekusi untuk kirim *e-mail*.



Gambar 7 : Tampilan Layar Pesan Terkirim

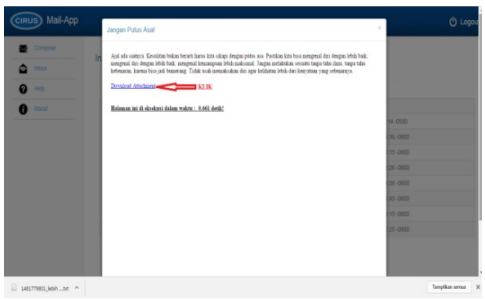
d. Tampilan Layar *Inbox*

Pada halaman ini, *user* dapat melihat daftar pesan yang masuk dan membaca pesan tersebut.



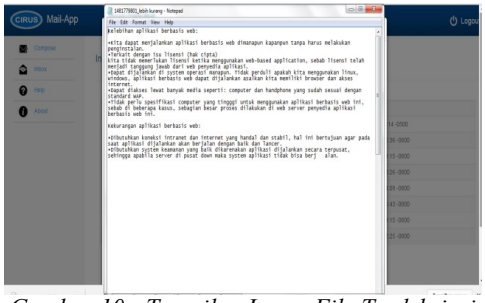
Gambar 8 : Tampilan Layar Inbox

Sebelumnya isi *e-mail* tersebut telah dienkripsi, dan saat *user* mengklik *subject,* isi *e-mail* tersebut langsung terdekripsi. Dan jika pada *e-mail* tersebut terdapat lampiran *file*, maka untuk mengunduhnya *user* harus mengklik link *download attachment*.



Gambar 9 : Tampilan Layar Baca Pesan

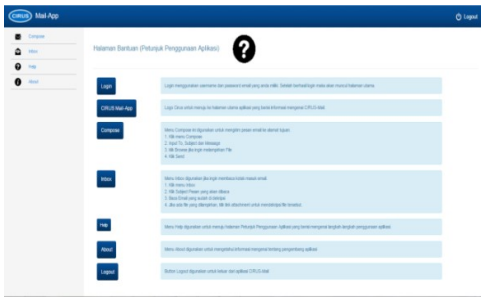
Isi dari *file* secara otomatis akan terdekripsi setelah *user* berhasil mengunduh *file* tersebut.



Gambar 10 : Tampilan Layar File Terdekripsi

e. Tampilan Layar *Help*

Pada halaman ini, *user* dapat mengetahui petunjuk penggunaan aplikasi ini.



Gambar 11 : Tampilan Layar Help

4.2. Pengujian Program

Pengujian aplikasi dilakukan menggunakan Laptop Acer Aspire 4750. Berikut rincian dan gambaran pengujian aplikasi :

a. Pengujian Proses Enkripsi

Tabel 4 : Tabel Pengujian Proses Enkripsi Teks

No	Plainteks	Kunci	Cipherteks	Waktu
1.	Hallo! Selamat pagi.	Semangat!	56d9e02fb9f6af5ce5bb2734da23c86c72c08cc2804c4293	0.984 Detik
2.	131150289, Sri Gustaria, Fakultas Teknologi Informasi, Teknik Informatika.	Universitas Budiluhur Jakarta 13	61111d86a05fdac249c555b039f27e686df2507b93c10c9e75620174faad5dd4bb9f8b7c975ea3a10d6ed9a720528b56bbe526cb0bc4637ade0b00886d8fded8b1b7335b9c5e6730f173130478b0b750	1.624 Detik
3.	"Jika ragu dalam melakukan sesuatu, sebaiknya tanya kepada diri sendiri, apa yang kita inginkan esok hari dari apa yang telah kita lakukan sebelumnya ". Oleh Jonh Lubbock - pakar boilogi Inggris.	John Lubbock	ba9103bdc151ae9bf9fc71cf5ec5f246b9c46e06b49561204ab9ecc95a48343e2bd1f5048d07aa5d4297d89e3c40e3d4b94e3d5a6de9d0d10a51e496e749d3995aa6133cfe0b26aae3c6a34f60998cf98cc76695e1797b56faac823f24ddc531e8442c943b5baa2a8dde52e489ec7c801c144dd7e51e1744188b330319baf71e40d6199	2.418 Detik

			c724b03b9f18c4c3093fcd0eb1d28acd879fe634f00bfe713945fad8d7d5c3014e9fc50b8ef7c2a7194d5e25c1555ec4c145fc0c5daddd5868f256bd446f3b07bcdced3bbb	

Tabel 5 : Tabel Pengujian Proses Enkripsi File

Nama *File*	**Ukuran *File***	**Nama *File* Hasil Enkripsi**	**Ukuran *File* Hasil Enkrips**	**Waktu**
Tutorial API.txt	2.28 Kb	Tutorial API.txt	3.1 Kb	1.974 Detik
LPJ Pelatihan Eksternal.doc	205 Kb	LPJ Pelatihan Eksternal.doc	273 Kb	3.832 Detik
PERHITUNGAN KRIPTOGRAFI METODE GOST.docx	105 Kb	PERHITUNGAN KRIPTOGRAFI METODE GOST.docx	141 Kb	10.440 Detik
Pengembangan Algoritma Kriptografi Gost.pdf	651 Kb	Pengembangan Algoritma Kriptografi Gost.pdf	869 Kb	8.354 Detik
Algoritma GOST.ppt	1.87 Mb	Algoritma GOST.ppt	2.50 Mb	20.018 Detik
LAPORAN SENSUS KAMPAR.pptx	1.44 Mb	LAPORAN SENSUS KAMPAR.pptx	1.91 Mb	17.820 Detik
List Abaya.xls	33 Kb	List Abaya.xls	44 Kb	2.851 Detik
Ganchart Pelatihan.xlsx	15.7 Kb	Ganchart Pelatihan.xlsx	21 Kb	2.661 Detik

b. Pengujian Proses Dekripsi

Tabel 6 : Tabel Pengujian Proses Dekripsi Teks

No	**Cipherteks**	**Kunci**	**Plainteks**	**Waktu**
1.	56d9e02fb9f6af5ce5bb2734da23c86c72c08cc2804c4293	Semangat!	Hallo! Selamat pagi.	0.013 Detik
2.	61111d86a05fdac249c555b039f27e686df2507b93c10c9e75620174faad5dd4bb9f8b7c975ea3a10d6ed9a720528b56bbe526cb0bc4637ade0b00886d8fded8b1b7335b9c5e6730f173130478b0b750	Universitas Budiluhur Jakarta 13	1311502890, Sri Gustaria, Fakultas Teknologi Informasi, Teknik Informatika.	0.072 Detik
3.	ba9103bdc151ae9bf9fc71cf5ec5f246b9c46e06b49561204ab9ecc95a48343e2bd1f5048d07aa5d4297d89e3c40e3d4b94e3d5a6de9d0d10a51e496e749d3995aa6133cfe0b26aae3c6a34f60998cf98cc76695e1797b56faac823f24ddc531e8442c943b5baa2a8dde52e489ec7c801c144dd7e51e1744188b330319baf71e40d6199c724b03b9f18c4c3093fcd0eb1d28acd879fe634f00bfe713945fad8d7d5c3014e9fc50b8ef7c2a7194d5e25c1555ec4c145fc0c5dadd5868f256bd446f3b07bcdced3bbb	John Lubbock	"Jika ragu dalam melakukan sesuatu, sebaiknya tanya kepada diri sendiri, apa yang kita inginkan esok hari dari apa yang telah kita lakukan sebelumnya". Oleh Jonh Lubbock - pakar boilogi Inggris.	0.209 Detik

Tabel 7 : Tabel Pengujian Proses Dekripsi File

Nama *File*	**Ukuran *File***	**Nama *File* Hasil Enkripsi**	**Ukuran *File* Hasil Enkrips**
Tutorial API.txt	3.1 Kb	Tutorial API.txt	2.28 Kb
LPJ Pelatihan Eksternal.doc	273 Kb	LPJ Pelatihan Eksternal.doc	205 Kb
PERHITUNGAN KRIPTOGRAFI METODE GOST.docx	141 Kb	PERHITUNGAN KRIPTOGRAFI METODE GOST.docx	105 Kb
Pengembangan Algoritma Kriptografi Gost.pdf	869 Kb	Pengembangan Algoritma Kriptografi Gost.pdf	651 Kb
Algoritma GOST.ppt	2.50 Mb	Algoritma GOST.ppt	1.87 Mb
LAPORAN SENSUS KAMPAR.pptx	1.91 Mb	LAPORAN SENSUS KAMPAR.pptx	1.44 Mb

List Abaya.xls	44 Kb	List Abaya.xls	33 Kb
Ganchart Pelatihan.xlsx	21 Kb	Ganchart Pelatihan.xlsx	15.7 Kb

5. PENUTUP

5.1. Kesimpulan

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

a. Aplikasi ini telah diatur oleh sistem sehingga isi pesan atau data yang terkandung dalam akun *e-mail* tersebut otomatis telah dienkripsi.
b. Meminimalisir kemungkinan kebocoran pesan yang terdapat di akun *e-mail* apabila menjadi korban dari *hacker*, karena pesan tersebut sudah dienkripsi.
c. Waktu untuk mengirim pesan berbanding lurus dengan ukuran teks dan ukuran file yang akan dienkripsi, semakin kecil ukuran teks dan *file*nya maka semakin cepat 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 perlu dikembangkan untuk menambahkan *file* dengan *extension* lainnya.
b. *Interface* masih sederhana, diharapkan dapat ditambahkan beberapa fitur seperti *forward message, delete message*, halaman *draft*, halaman *sent*, dan lainnya.
c. Dalam pengembangannya, aplikasi ini dapat menggunakan metode kompresi sehingga ukuran file yang dienkrip atau didekrip dapat lebih diminimalisir.

DAFTAR PUSTAKA

[1] Ariyus, Doni. 2008. *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi.* Yogyakarta : C.V Andi OFFSET.

[2] Churchhouse, Robert. 2004. *Codes and Ciphers, Julius Caesar, the Enigma, and the Internet.* UK : Cambridge University Press.

[3] Munir, R. 2006. *Kriptografi*. Bandung : Informatika.

[4] Oppliger, Rolf. 2005. *Contemporary Cryptography.* USA : Artech House, Inc.

[5] Pardosi, Mico. 2006. *E-mail gratis Yahoo! Indonesia.* Surabaya : Dua Selaras.

[6] Presmann, S. Roger. 2001. *Software Engineering : A Practitioner is Approach.* McGraw- Hill Companies. New York.

[7] Sadikin, Rifki. 2012. *Kriptografi untuk Keamanan Jaringan.* Yogyakarta.