IMPLEMENTASI KRIPTOGRAFI MENGGUNAKAN METODE *BLOWFISH DAN BASE64* UNTUK MENGAMANKAN *EMAIL* BERBASIS WEB

Rizki Ripai[1)], **Safrina Amini**[2)]
[1]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1,2]Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
Telp. (021) 5853753, Fax. (021) 5866369
E-mail : rizky.ripai@gmail.com[1)], safrina.amini@budiluhur.ac.id[2)]

ABSTRAK

Seiring perkembangan teknologi saat ini yang semakin maju, keamanan data menjadi sangat penting, untuk menghindari terjadinya tindak pencurian informasi rahasia oleh pihak-pihak yang tidak bertanggung jawab. Berbagai cara dilakukan untuk menjaga keamanan data, salah satu cara yang dapat digunakan untuk mengamankan data adalah dengan memanfaatkan kriptografi.Pembuatan keamanan enkripsi *email* pada PT Anugrah Putra Mandiri merupakan hal yang ingin penulis lakukan, dikarenakan PT Anugrah Putra Mandiri merupakan perusahaan yang bergerak di bidang Jasa Legal Impor, perusahaan ini memiliki data-data penting yang perlu dijaga kerahasiaannya seperti data keuangan, penawaran, customer serta data penting perusahaan yang lainnya. Semakin berkembangnya teknologi saat ini menyebabkan proses pengiriman data dapat dilakukan dengan mudah melalui berbagai macam media, oleh karena itu dibutuhkan keamanan dalam penyimpanan data dan kerahasiaan data tersebut. Salah satu cara yang digunakan yaitu dengan cara mengenkripsi *email* tersebut melalui aplikasi kriptografi dalam menangani masalah akan bocornya informasi yang akan dikirim oleh PT Anugrah Putra Mandiri, maka diperlukan aplikasi pengamanan data yang dapat mengamankan *email* yang dimiliki. Dengan menerapkan algoritma kriptografi *Blowfish* dan algoritma *Base64*, aplikasi ini dibuat dengan bahasa pemograman *PHP* berbasis web. Aplikasi ini dapat mengamankan kerahasiaan data pada PT Anugrah Putra Mandiri dari terjadinya kebocoran informasi atau pencurian data oleh orang lain. Setelah proses enkripsi sudah dilakukan akan mengeluarkan *output* yang tentunya tidak bisa dibaca atau tidak dapat dibuka kembali oleh pihak yang tidak bertanggung jawab. Aplikasi ini dikembangkan dengan menggunakan bahasa pemrograman *PHP*. Sifatnya adalah menyandikan data untuk meningkatkan keamanan data dan informasi terhadap kerusakan, pencurian dan penyalahgunaan data penting perusahaan. Penulis mengambil kesimpulan dengan adanya sistem keamanan pada PT Anugrah Putra Mandiri ini diharapkan dapat melindungi kerahasiaan data dan informasi serta dapat memberikan manfaat bagi perusahaan dalam menjalankan usahanya.

Kata Kunci:*Blowfish, Base64*, Enkripsi, Dekripsi, *Email.*

1 PENDAHULUAN

Di PT Anugrah Putra Mandiri, untuk proses pengiriman data saat ini pada PT Anugrah Putra Mandiri bisa secara langsung berbentuk fisik dan bisa juga dikirim melalui *email*. Data yang diterima atau dikirim melalui *email* merupakan salah satu data yang penting. Oleh karena itu dibutuhkan suatu metode yang dapat menjaga rahasia data tersebut. Metode yang dimaksud adalah kriptografi yang merupakan keilmuan dalam penyandian pesan dengan tujuan menjaga keamananya. Teknik kriptografi yang dibutuhkan masa kini tetap harus menyesuaikan dirinya terhadap meluasnya penggunaan *computer digital* pada masa kini.

Perlunya ada suatu system pengamanan data baik saat pengiriman maupun penerimaan *email*. Penyandian data adalah salah satu cara untuk mengamankan data. Dalam penelitian ini akan mencoba mengimplementasikan suatu cabang ilmu kriptografi, semua data diubah menjadi sandi-sandi yang tidak di ketahui orang lain bahkan tidak bisa mengembalikannya kebentuk semula, ini disebut enkripsi dan dekripsi. Dalam pembuatan system ini, akan menggunakan metode *Blowfish* dan *Base64* untuk proses enkripsi dan dekripsi pada *email*.

2 METODE PENELITIAN

Dalam proses perancangan aplikasi ini penulis menggunakan dua metode, antara lain:

2.1 Metode Pustaka

Mencari, informasi dari buku dan berbagai macam artikel berkaitan dengan *Enkrip, dekrip,*

algoritma *Blowfish*, *Base64* serta semua yang berhubungan dengan topik ini. Serta terus menggali sumber-sumber pustaka lainnya yang juga mendukung, seperti jurnal, forum diskusi, pendapat ahli, dan sebagainya baik media cetak maupun elektronik.

2.2 MetodeProrotyping

Berikut tahapan dari Metode *Prototyping*menurut [1]:

1) Menentukan Kebutuhan

Bertemu dengan pengguna yang akan menggunakan aplikasi dengan mengidentifikasi kebutuhan, dan kemudian secara bersama-sama menentukan secara garis besar konsep sistem yang akan dikembangkan.

2) Membuat *Protoype*

Bila garis besar konsep sudah disepakati, selanjutnya lakukan *prototype* dengan membuat rancangan aplikasi sementara sesuai pada garis besar konsep.

3) Evaluasi *Prototype*

apakah sudah sesuai dengan keinginan pengguna. Jika *prototype* sudah selesai maka langkah keempat akan diambil, namun jika tidak maka *prototype* akan diperbaiki dengan mengulang langkah pertama, kedua, dan ketiga.

4) Mengkodekan Aplikasi

prototyping diterjemahkan kedalam bahasa pemograman yang sesuai.

5) Menguji Aplikasi

jika aplikasi sudah selesai dibuat menjadi produk yang dapat digunakan, aplikasi akan dites terlebih dahulu sebelum digunakan.

6) Evaluasi Aplikasi

Pengguna melakuakan pengecekan pada aplikasi,. Jika sudah maka langkah ketujuh dapat dijalankan, namun jika aplikasi belum sesuai maka akan dibuat kembali dengan mengulang langkah keempat dan kelima.

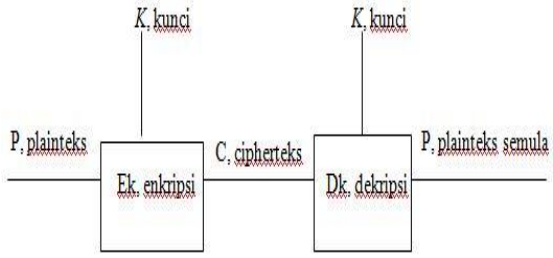
7) Menggunakan *Prototype*

Dilakukannya pengembangan aplikasi sesuai dengan hasil evaluasi, penambahan dan perbaikan-perbaikan sesuai dengan permintaan *user* ditahap sebelumnya (evaluasi *prototype)*, jika aplikasi sesuai dengan keinginan *user*, maka aplikasi tersebut dapat digunakan.

3 LANDASAN TEORI

3.1 Enkripsi dan Dekripsi

Enkripsi dan dekripsi bias digunakan pada data yang dikirim. Teks asli dirubah menjadi cipherteks disebut enkripsi. Sedangkan cipherteks dirubah menjadi teks asli disebut dekripsi. Istilah *encryption of data inmotion* mengarah pada enkripsi pesan yang pindahkan melewati saluran komunikasi, dan istilah *encryption of data at-rest* mengarah pada enkripsi dokumen yang disimpan di dalam storage [2].



Gambar 1: Skema enkripsi dan dekripsi

3.2. Algoritma Blowfish

Blowfish dibuat untuk memenuhi kriteria desain yang cepat dalam penerapannya dimana pada keadaan maksimal akan mencapai 26 *clock cycle* per Byte, bahkan bisa berjalan pada memori yang kurang dari 5 KB, sangat sederhana dalam algoritmanya dan mudah diketahui kesalahannya, bahkan keamnannya pun dimana jumlah kunci bervariasi (min 32 bit, maks 448 bit, multiple 8 bit, default 128 bit) [2].

Algoritma Blowfish terdiri atas dua bagian :

3.2.1 Key-Expansion

kunci (min 32-bit, maks 448-bit) menjadi beberapa array subkunci (*subkey*) dengan jumlah (18x32-bit untuk P-array dan 4x256x32-bit untuk S-box sehingga totalnya 4168 Byte atau 33344 bit). Kunci disimpan dalam Q-array:

$$Q1, Q2, \ldots Qj \; 1 \leq j \leq 14$$

Sebelum enkripsi dan dekripsi data Kunci-kunci ini diambil dengan menggunakan subkunci yang harus dihitung terlebih dahulu. Sub-sub kunci yang digunakan terdiri dari: R-array yang terdiri dari 18 buah 32-bit subkunci,

$$R1, R2, \ldots, R18$$

P-box yang terdiri dari 4 buah 32-bit, masing-masing memiliki 256 entri:

P1,0, P1,1, P1,2, P1,3, ... P1,255
P2,0, P2,1, P2,2, P2,3, ... P2,255
P3,0, P3,1, P3,2, P3,3, ... P3,255
P4,0, P4,1, P4,2, P4,3, ... P4,255

perhitungan atau pembangkitan subkunci tersebut adalah langkah-langkahnya sebagai berikut:

a) Inisialisasi R-array yang pertama dan juga empat P-box, berurutan, dengan *string* yang telah pasti. *String* bernilai digit-digit heksadesimal dari phi, bukan termasuk angka 3 di awal.
b) XOR-kan R1 dengan 32-bit awal kunci, XOR-kan R2 dengan (32-bit) selanjutnya dari kunci,

dan seterusnya untuk semua bit kunci. Ulangi siklus seluruh bit kunci secara berurutan sampai seluruh R-array terXOR-kan dengan bit-bit kunci. Atau jika disimbolkan : R1 = R1 XOR Q1, R2 = R2 XOR Q2, R3 = R3 XOR Q3, . . . R14 = R14 XOR Q14, R15 = R15 XOR Q1, . . . R18 = R18 XOR Q4.

c) Enkripsikan *string* yang seluruhnya 0 dengan algoritma *Blowfish*, menggunakan subkunci yang sudah gambarkan pada langkah 1 dan 2.
d) Gantikan R1 dan R2 dengan *output* dari langkah 3.
e) Enkripsikan *output* langkah 3 menggunakan algoritma Blowfish dengan subkunci yang sufah dirubah.
f) Tukar R3 dan R4 dengan *output* dari langkah 5.

3.2 Enkripsi Data

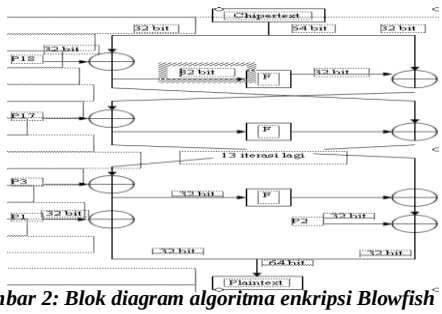
Dilakukan sebanyak 16 kali putaran, jumlah data yang diambil yaitu 64 bit dari nilai x. Setiap putaran terdiri dari permutasi kunci dependen,. Langkahnya adalah seperti berikut:

a) kelompokan X menjadi 2 bagian yang masing-masing terdiri dari 32-bit: L, R.
b) Lakukan langkah berikut
   For i = 1 to 16:
   L = L XOR Ri
   R = F(L) XOR R
   Tukar L dan R
c) Setelah iterasi ke-16, tukar L dan R lagi untuk melakukan kembali pertukaran terakhir.
d) Lalu lakukan
   R = R XOR R17
   L = L XOR R18
e) Terakhir, gabungkan kembali L dan R untuk mendapatkan cipherteks.

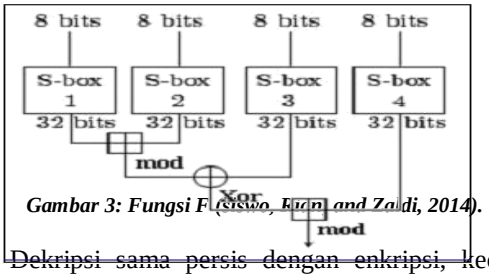
Pada gambar 2 telah dijelalskan tentang blok diagram enkripsi blowfish.

Berikut adalah fungsi F: bagi L dalam empat kuarter 8-bit yaitu a, b, c dan d seperti Gambar 3 maka:

$F(L) = ((P_{1,a} + P_{2,b} \bmod 2^{32}) (+) P_{3,c}) + P_{4,d} \bmod 2^{32}$



Gambar 2: Blok diagram algoritma enkripsi Blowfish

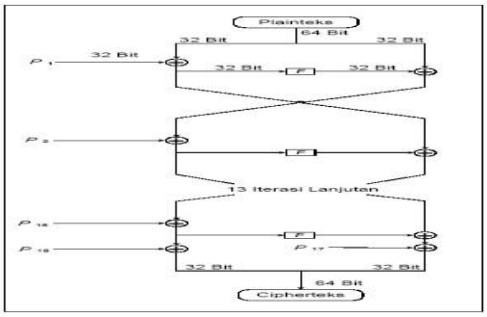


Gambar 3: Fungsi F (Siswo, Erna and Zaidi, 2014).

Dekripsi sama persis dengan enkripsi, kecuali bahwa R1, R2,… ,R18 digunakan pada urutan kebalkannya (reverse). Algoritmanya dapat dinyatakan sebagai berikut:

for i = 1 to 16 do
Ri = Li-1 XOR R19-i;
Li = F[Ri] XOR Ri-1;
L17 = R16 XOR R1;
R17 = L16 XOR R2;

Dibawah ini adalah gambar blok diagram decode algoritma blowfish, lihat pada gambar 4.



Gambar 4: Blok diagram dekripsi Blowfish

3.3 Algoritma Base64

Karakter yang dihasilkan pada transformasi *base64* ini terdiri dari 0..9, a..z dan A..Z, serta ditambah simbol "+" dan "/" serta satu buah karakter sama dengan (=) di dua karakter terakhir yang dipakai untuk pengisian pad atau dengan kata lain penyesuaian dan menggenapkan data *binary*. Karakter simbol yang akan dihasilkan akan sesuai dari proses algoritma yang berjalan [4].

base64 sering digunakan dalam dunia Internet sebagai media data format untuk mengirimkan data, penggunaan tersebut dikarnakan hasil dari *encode base64* berupa *plaintext*, maka data ini akan jauh lebih mudah dikirim, dibandingkan dengan format data yang berupa *binary*. Algoritma *base64* menggunakan kode *ASCII* dan kode index *base64* dalam melakukan proses enkripsi ataupun dekripsinya. Dalam melakukan enkripsi pada URL website, kode index *base64* perlu dimodifikasi. Simbol "+" dimodifikasi menjadi "-" dan simbol

simbol "/" menjadi "_". Tabel index *base64* dapat dilihat pada tabel

Tabel 1: Tabel Base64 lengkap

Data 6 bit	Karakter encoding 64	Data 6 bit	Karakter encoding 64	Data 6 bit	Karakter encoding 64	Data 6 bit	Karakter encoding 64
0	A	16	Q	33	h	50	y
1	B	17	R	34	i	51	z
2	C	18	S	35	j	52	0
3	D	19	T	36	k	53	1
4	E	20	U	37	l	54	2
5	F	21	V	38	m	55	3
6	G	22	W	39	n	56	4
7	H	23	X	40	o	57	5
8	I	24	Y	41	p	58	6
9	J	25	Z	42	q	59	7
10	K	26	A	43	r	60	8
11	L	27	B	44	s	61	9
12	M	28	C	45	t	62	+
13	N	29	D	46	u	63	/
14	O	30	E	47	v	pad	=
15	P	31	F	48	w		
16	Q	32	G	49	x		

Menurut [5] yang dikutip oleh [3], teknik enkripsi *base64* sangat sederhana, jika terdapat sebuah (*string*) *bytes* yang akan disandikan ke algoritma *base64* maka tahapannya yaitu:

a. kelompokan *string* bytes tersebut ke per-3 bytes.
b. Buat 3 byte tersebut menjadi 8 bit masing-masing bytenya sehingga berjumlah 24 bit.
c. kemudian 24 bit yang disimpan di kelompokan menjadi 6 bit lagi sehingga menjadi 4 kelompok.
d. jadikan nilai-nilai desimal dari bit diatas menjadi index untuk memilih maksimal index ke 64 atau karakter ke 63 dari penyusun *base64*.

Dan seterusnya hingga akhir *stringbytes* yang akan mengalami konversi. Apabila dalam proses *encoding* terdapat sisa pembagi, maka tambahkan karakter pad (=) sebagai penggenap sisa tersebut. Oleh karena itu, biasanya pada *base64* selalu tampil satu atau dua karakter (=).

Transformasi *Base64* adalah salah satu algoritma untuk *encoding* dan *decoding* suatu data kedalam format *ASCII* yang di dasarkan oleh bilangan 64, karakter yang didapat dari *Base64* terdiri dari 0-9, a-z, dan A-Z, serta ditambah dengan 2 karakter terakhir yaitu / dan +. Langkah enkripsi menggunakan algoritma Base64.

a. Ubah huruf-huruf yang akan di enkripsi menjadi kode-kode *ASCII.*
b. Kode-kode *ASCII* tersebut diubah lagi menjadi kode BINER.
c. Bagi kode biner tersebut menjadi hanya 6 angka per blok dan berjumlah kelipatan 4 blok.
d. Jika angka biner tidak berjumlah 6 angka dan 4 blok maka akan ditambah kode biner 0 sehingga mencukupi menjadi 4 blok.
e. Blok-blok tersebut diubah kembali menjadi kode desimal (data di baca sebagai *Index*).
g. Untuk *decode* teks akan diproses dan akan diurutkan secara terbalik yaitu :
   1) *Ciphertext* dikonverssi ke *Decimal*
   2) *Decimal* dikonversi ke bit *pattern* 6 bit
   3) Bit *pattern* 6 bit dikelompokkan menjadi bit *pattern* 8 bit / Biner
   4) Biner di konversi lagi ke *ASCII*
   Dan terakhir akan kembali lagi ke *plaintext*

4 RANCANGAN APLIKASI

Aplikasi yang dibuat terdiri dari sembilan buah tampilan menu, yang terdiri dari *Login*, Menu Utama atau Beranda, menu Tulis Pesan, menu Kotak Masuk, menu Kotak Keluar, menu Trash, menu *Management* Pengguna, menu Panduan dan menu *My Profile.*

Di bawah ini adalah rancangan database dari aplikasi ini yang digunakan untuk menyimpan semua data yang dibutuhkan untuk kelangsungan proses aplikasi. Tabel 2 menunjukan gambar *Class Diagram* yang digunakan dalam pembuatan basis data aplikasi.

Tabel 2: Rancangan Database Email

Nama Field	Type	Ukuran	Keterangan
email_id	Int	11	ID Email
email_keys	Varchar	255	Keys Enkripsi Email
email_dari_id	Int	11	Pengirim Email
email_kepada_id	Int	11	Penerima Email
email_cc_id	Int	11	Penerima Email Pasif (Carbon Copy)
email_bcc_id	Int	11	Penerima Email yg tidak diketahui penerima lain (Blank Carbon Copy)
Subject	Varchar	255	Subject Email
message_textasli	Text		Pesan Asli (*Plaintext*)
message_textenkripsi	Text		Pesan Enkripsi (*Chipertext*)
document_filename	varchar	150	Nama file yang terenkripsi
document_size	varchar	150	Ukuran file
document_fileasli	varchar	150	Nama file asli
time_encrypt	varchar	50	Waktu proses enkrip
time_decrypt	varchar	50	Waktu prosess dekrip
created_at	Datetime		Tangga Kirim Email
Status	Tinyint	4	Status Email
status_decrypt	Tinyint	4	

Tabel 3: Rancangan Database Pengguna

Nama Field	Type	Ukuran	Keterangan
pengguna_id	Int	11	ID Pengguna
pengguna_fullname	Char	20	Nama Lengkap Pengguna
pengguna_photo	Varchar	100	Foto Pengguna
pengguna_email	Varchar	100	Email Pengguna
Password	Varchar	100	Password
pengguna_phone	Varchar	30	No. Handphone
pengguna_address	Varchar	50	Alamat Pengguna
Status	Varchar	100	Status
create_at	Datetime		Tanggal Daftar
update_at	Datetime		Ubah Data

5 HASIL DAN PEMBAHASAN

5.1 Spesifikasi Hardware dan Software

Pengimplementasian aplikasi ini memerlukan kelengkapan.Diperlukan 1 set

komputer / *notebook* dan software pendukung. Pembahasannya meliputi, antara lain:

a. Spesifikasi Perangkat keras

Perangkat keras (*Hardware*) yang mendukung sistem apliasi ini secara maksimal adalah sebagai berikut :

1) *Processor Intel(R) Core(TM)* i3-2328M @ 2,20GHz
2) *RAM / Memory* 4 GB
3) *Hardisk* 500 GB

b. Spesifikasi Perangkat Lunak

Perangkat lunak (*Software*) yang mendukung istem apliasi ini secara maksimal adalah sebagai berikut :

1) Sistem Operasi Microsoft Windows 10 Enterprise 64-bit
2) *Browser* (Google Chrome)
3) *SMTP Elastic Email*
4) XAMPP, PHP 5.6 dan Mysql
5) *Sublime Text3*
6) *SMTP Elastic Email*

5.2 Tampilan Layar Program

a. Tampilan Layar Halaman *Login*

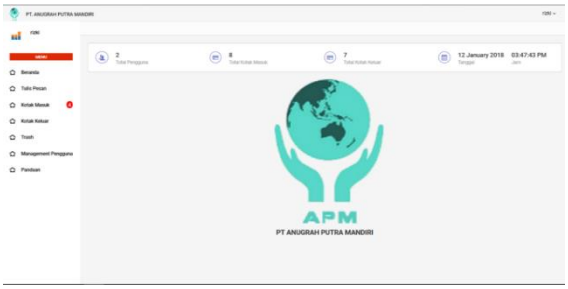
Tampilan layar dari halaman *login* pada gambar 5 ini muncul ketika pada saat aplikasi ini pertama kali dijalankan atau diakses. Pada halaman *login* terdapat *email* dan *password*. *User* harus memasukan *email* dan *password* yang terdaftar agar

dapat

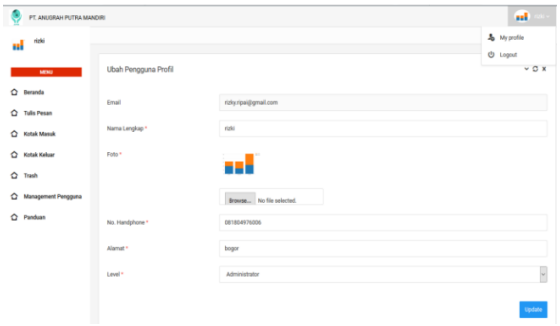
Gambar 5: Tampilan Layar Halaman Login

b. Tampilan Layar Halaman Beranda

Tampilan layar dari halaman beranda atau menu utama bisa dilihat dari gambar 6.halaman ini akan pertama kali muncul pada saat user telah berhasil melakukan *login*. Pada halaman ini terdapat menu-menu yang dapat dipilih oleh user, diantaranya menu *Beranda*, Tulis Pesan, Kotak Masuk, Kotak Keluar, Trash, *Management* Pengguna dan Panduan. Selain itu tiap-tiap menumemiliki submenu, seperti menu beranda memilikisubmenu *myprofile* dan *logout*dapat dilihat dari gambar 4.3. Sedangkan *logout* yang memiliki submenu untuk kembali ke menu *login*. Pada menu *my profile* yang terdapat nama*user* memiliki submenu, seperti ubah pengguna ~~profil dan ubah *password* profil.~~



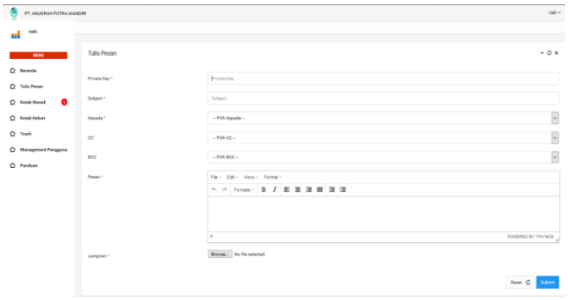
Gambar 6: Tampilan Layar Menu Beranda



Gambar 7: Tampilan Layar Submenu Beranda MyProfile Dan Log out

c. Tampilan Layar *Form* Tulis Pesan

Pada *form* ini Untuk melakukan proses pengiriman pesan email dan proses enkripsi *email*, *user* harus "Tulis Pesan", setelah itu isi "kata kunci", isi "subject", dan isi "Kepada". Sedangkan jika *user* pengirim ingin menambahkan tujuan alamat pengiriman ke 2 *user* penerima, maka *user* diharuskan untuk memasukan alamat *email* pada "CC" dan "BCC", setelah itu *user* bisa menuliskan isi *email* pada Pesan. Setelah itu *user* pengirim bisa langsung mengirimkan *email* yang terenkripsi pada *user* penerima.seperti gambar 8.

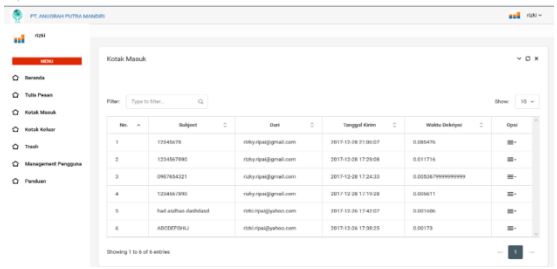


Gambar 8: Tampilan Layar Halaman Tulis Pesan

d. Tampilan Layar HalamanKotak Masuk

Halaman ini difungsikan sebagai tempat penyimpanan masuknya *emailuser* penerima yang sudah terenkripsi dari *user* lain atau pengirim. Selain berfungsi sebagai tempat penyimpanan *email* masuk yang sudah terenkripsi, halaman ini juga berfungsi sebagai tempat untuk mendekripsi *email* yang masuk ~~pada *user penerima*. Untuk mendekripsi *email* yang~~

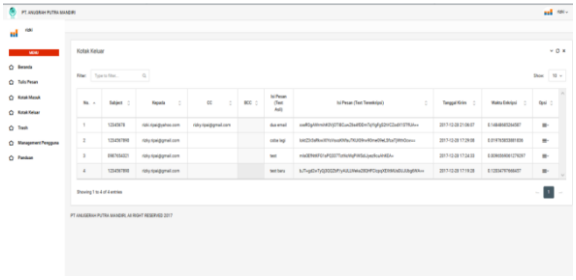
masuk *user* penerima dapat memilih perintah ”*Decrypt*” pada tabel “Opsi” maka *email* yang terenkripsi tadi sudah berubah menjadi terdekripsi, dan untuk melihat hasil isi dari *email* yang sudah terdekripsi *user* penerima dapat melihat langsung pada halaman Decrypt pesan. Selainitu kotakmasuk juga diberikan fungsi untuk menghapus *email*masuk dan melihat *email* yang sudah terdekrip. Seperti gambar 9.



Gambar 9: Tampilan Layar Halaman Kotak Masuk

e. Tampilan Layar Halaman Kotak Keluar

Halaman ini difungsikan sebagai tempat penyimpanan riwayat keluarnya *emailuser* pengirim yang sudah terenkripsi oleh aplikasi ini. Selain berfungsi sebagai tempat penyidfccccccmpanan riwayat *email* keluar yang sudah terenkripsi, pada halamanhanya ada perintah untuk membuang riwayat *email* yang telah tersimpan pada kotak keluar *user* pengirim. Untuk menjalankan fungsi membuang riwayat *email* yang telah terkirim, untuk menghapus *email* keluar yang telah dibuang kedalam*trash.* Pengirim dapat menjalankan perintah *trash* pada tabel opsi untuk membuang data dari kotak keluar ke *trash*.Seperti gambar 10.



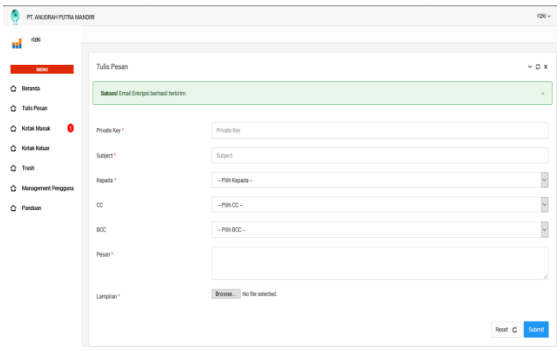
Gambar 10: Tampilan Layar Halaman Kotak Keluar

6 PENGUJIAN APLIKASI

Setelah semua kebutuhan terpenuhi baik software dan maupun hardware, maka tahapan selanjutnya adalah uji coba aplikasi. Pada tahap ini akan diuraikan mengenai pengujian *encrypt* dan *decryptemail.* Pada tahap ini akan mendapatkan hasil perbandingan *email* asli dengan *email* hasil *encrypt.*

a. Proses Encrypt dan Decrypt Email

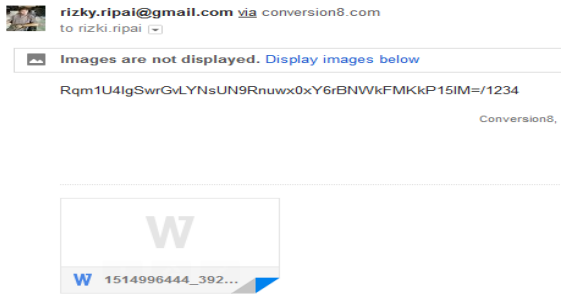
Untuk melakukan proses *encrypt email,user* pengirim terlebih dahulu memilih menu “Tulis Pesan”, kemudian mengisi beberapa *form* wajib, seperti *form* “Kata Kunci” Yang wajib untuk tidak dikosongkan dengan minimal jumlah kata kunci 4 karakter dan maksimal jumlah kata kunci sebanyak 32 karakter didalam form ini hanya digunakan *input* kata kunci berbentuk angka saja untuk dalam berbentuk huruf alphabet belum bisa digunakan didalam aplikasi ini, selanjutnya *user* diwajibkan mengisi *form “Subject”* dengan minimal jumlah 8 karakter dengan bentuk *input* bebas (angka,huruf,Simbol), selanjutnya *user* diwajibkan mengisi *form* “Kepada” dengan minimal jumlah karakter sesuai alamat *email* yang didaftarkan. Kemudian jika user ingin menambahkan 1 atau bahkan 2 email berbarengan, user diharuskan mengisi *form* “CC” dan *form* “BCC” yang difungsikan sebagai *form* tambahan tujuan alamat *email*. Setelah *form-form* diatas sudah terisi sesuai dengan fungsinya, maka *user* diarahkan kedalam *form* ”Pesan” yang difungsikan untuk mengirim pesan *email* pada seluruh *user* dari aplikasi ini yang batas karakter pengiriman *email* sebanyak 1 karakter dan jumlah maximal karakter yang tidak dibataskan pada aplikasi ini. Kemudian didalam aplikasi ini disediakan *form upload* lampiran jika *user* ingin menambahkan file lampiran, kemudian 2 papan tombol yaitu papan tombol “*Reset*” dan papan tombol “*Submit*”, papan tombol reset berfungsi untuk mengosongkan seluruh isi dari *form* tulis pesan, dan papan tombol submit berfungsi untuk melanjutkan proses enkripsi *email* dengan mengirimkan seluruh isi pesan yang telah terenkripsi kepada user penerima. Setelah penulisan pesan *email* yang telah berhasil terenkripsi maka tampil halaman,



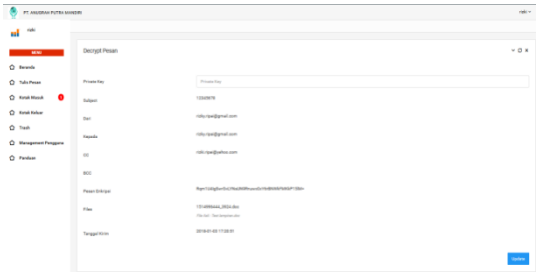
seperti gambar 11.

Gambar 11: Tampilan Layar Form Tulis Pesan Berhasil Di Enkripsi

Setelah muncul notifikasi email berhasil terenkripsi oleh aplikasi ini, maka *email* yangberisi pesan dalam bentuk chiperteks akan masuk ke email penerima dan penerima dapat melihat email yang terenkripsi tadi pada akun *emailgoogle* (gmail) atau yahoo *user* penerima yang terdaftar pada aplikasi ini. Dapat dilihat pada gambar 12.



Gambar 12: Tampilan Layar Email Masuk Terenkripsi di Gmail



Gambar 13: Tampilan Layar Proses Dekripsi Email Dari Kotak Masuk

6.1 Evaluasi Program

Dalam aplikasi ini ditemukan beberapa kelebihan dan kekurangan antara lain : keunggulan dan kelemahan, diantaranya :

a. Kelebihan Aplikasi
   1) *Email* yang dikirim melalui aplikasi langsung dapat terenkripsi dan tidak dapat dibaca apabila tidak menggunakan aplikasi ini.
   2) Aplikasi ini berbasis web yang dapat digunakan untuk kedepannya secara *online*, sehingga dapat diakses dengan mudah dari mana saja.
   3) Adanya pemberitahuan pesan masuk via akun*email* yang terdaftar pada aplikasi.
   4) Pengguna aplikasi ini bisa mendaftar dengan menggunakan akun penyedia layanan *email*, seperti : akun *Gmail*, akun *Yahoo!*, lainnya.

b. Kekurangan Aplikasi
   1) Belum adanya fitur *Reply*, *Forward*, *Draft*, dan *Restore* pada menu *trash*.
   2) Belum bisa meng-upload gambar *profile* dengan ukuran besar.
   3) Belum bisa mengirimkan email lebih dari tiga penerima secara langsung.

7 KESIMPULAN

Aplikasi pengamanan *email* berbasis Web menggunakan metode algoritma *Blowfish* dan *Base64* sangat diperlukan karena:

a. Sebuah aplikasi yang mengimplementasikan algoritma kriptografi *Blowfish* dan *Base64* untuk enkripsi *email* telah berhasil diciptakan.
b. Kerahasiaan Email dapat terjaga dengan adanya aplikasi ini.
c. Semoga kekhawatiran semua orang terhadap pentingnya keamanan data dapat ditangani oleh aplikasi ini .
d. Aplikasi ini berjalan dengan sistem sehingga isi pesan atau data yang terkandung didalam akun-akun *email* yang terdaftar tersebut otomatis telah dienkripsi dengan baik.

DAFTAR PUSTAKA

[1] Roger, S. Pressman, Ph.D., 2012,Rekayasa Perangkat Lunak (Pendekatan Praktisi) Edisi 7 Buku 1“, Yogyakarta: Andi.

[2] Siswo, W., Rian, F., and Zaldi, I. (2014) 'Aplikasi Teknik Enkripsi Dan Dekripsi File Dengan Algoritma Blowfish Pada Perangkat Mobile Berbasis Android', SETRUM, Vol. 3, No. 1.

[3] STTG. (2009) Modul Praktikum Algoritma dan Pemograman dalam Bahasa Pascal dan C. Garut: Sekolah Tinggi Teknologi Garut

[4] Wahyu, F., Rahangiar, A. P., & Fretes, F. d. (2012). Penerapan Algoritma Gabungan RC4 dan Base64 Pada Sistem Keamanan E-Commerce.Retrieved Mei 25, 2016, from Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana web site on World Wide Web: http://journal.uii.ac.id/index.php/Snati/article/viewFile/2873/2628.

[5] Ariyus, Dony. 2008. *Pengantar Ilmu Kriptografi*.Yogyakarta: Andi offset