PENERAPAN KRIPTOGRAFI PADA APLIKASI SECURE - MAIL BERBASIS WEB MENGGUNAKAN ALGORITMA CAESAR CIPHER DAN RC4

Safwan Reza[1)], Noni Juliasari M, Kom[2)]

[1]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1,2]Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : safwan.reza88@gmail.com[1)], dosen.pebimbing@budiluhur.ac.id[2)]

Abstrak

Seiring pesatnya perkembangan teknologi informasi serta semakin mudahnya orang untuk melakukan interaksi dalam berkomunikasi maka beberapa dampak dan permasalahan barupun muncul dalam penyampaian informasi dan komunikasi, oleh karena mudahnya pengaksesan media komunikasi oleh semua orang serta membawa dampak bagi keamanan informasi maupun pesan yang menggunakan media komunikasi tersebut. Oleh karena itu, pengiriman maupun penyimpanan data melalui media elektronik memerlukan suatu metode maupun proses yang mampu menjamin keamanan. Kriptografi adalah salah satu solusi atau metode pengamanan data yang cukup tepat untuk menjaga kerahasiaan serta keamanan informasi, dan juga dapat meningkatkan keamanan suatu data ataupun informasi. Diterapkannya metode ini agar informasi yang bersifat rahasia yang akan dikirim melalui suatu jaringan, seperti dalam jaringan Internert ataupun LAN, tidak dapat diketahui atau disalahgunakan oleh orang atau pihak yang tidak memiliki kepentingan di dalamnya. Salah satu contoh adalah penggunaan email. Kriptografi algoritma Caesar Cipher dan RC4 yang digunakan untuk proses enkripsi dan dekripsi email. Aplikasi ini menggunakan bahasa pemograman PHP. Aplikasi yang akan dibangun merupakan perangkat lunak yang berbasis web dan memiliki fungsi untuk melakukan enkripsi dan dekripsi pada surat elektronik (email). Aplikasi dapat melakukan pengiriman dan menerima email. Pengguna akan berinteraksi dengan perangkat lunak melalui web browser.

Kata kunci: Caesar Cipher, RC4, enkripsi, dekripsi, ciphertext, plaintext, AyoGan Reload, PHP

1. PENDAHULUAN

Email adalah salah satu layanan terpenting dalam penggunaan internet saat ini.*Email* juga mengubah cara dan interaksi dalam komunikasi sehingga dapat berkomunikasi jarak jauh dengan waktu yang cukup singkat.Oleh karenanya kerahasiaan dari pada email tersebut merupakaan objek penting untuk beberapa pihak.

Perkembangan teknologi dan informasi yang sangat pesat saat ini membawa pertumbuhan dunia pada ujung tombak kemajuan.Oleh karenanya informasi saat ini nilainya sangat tinggi dan penting.Teknologi informasi yang ada saat ini sangat erat hubungannya dengan media komunikasi sebagai media untuk menyampaikan informasi dari suatu tempat ke tempat lainnya.Banyaknya informasi yang ingin disampaikan berinteraksi melalui media komunikasi tersebut.

Banyaknya media komunikasi saat ini baiknya adalah media yang mudah di jangkau serta digunakan oleh semua orang.Sebagai contoh media komunikasi yang paling sering digunakan adalah jaringan internet, telepon maupun *email*.Karena media komunikasi saat ini mudah diakses oleh semua orang maka hal tersebut dapat membawa dampak terhadap keamanan informasi maupun pesan yang menggunakan media komunikasi.Pada tahapan ini informasi menjadi sangat mudah untuk diketahui bahkan dimanipulasi oleh pihak yang bahkan tidak memiliki kepentingan sekalipun.

Di perusahaan seperti CV. Alkenza Mandiri, informasi yang diterima atau dikirim melalui *email* adalah salah satu data yang paling penting. Oleh karenanya dibutuhkan suatu metode yang dapat menjaga rahasia informasi tersebut.Metode tersebut adalah kriptografi dimana merupakan seni serta bidang keilmuan dalam penyandian pesan maupun informasi untuk menjaga keamanannya.

Ilmu kriptografi merupakan salah satu teknik untuk mengamankan pesan maupun data.Pengamanan pesan maupun data ini dapat dilakukan menggunakan beberapa algoritma, pada pembahasan ini menggunakan algoritma Caesar Cipher dan RC4. Algoritma Caesar Cipher dan RC4 adalah bagian dari perkembangan ilmu kriptografi,serta digolongkan sebagai kriptografi klasik.

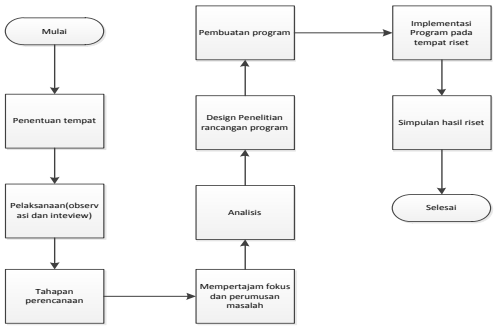
Karya ilmiah ini dibuat berbasis *web* menggunakan bahasa pemrograman *PHP*,serta menggunakan kombinasi metode kriptografi Algoritma Caesar Cipher dan RC4 untuk mengamankan pesan teks di badan *email*, agar hanya bisa di baca oleh penerima pesan email tersebut dengan menggunakan aplikasi yang dibuat.

2. METODE PENELITIAN

Merupakan penjelasan dan langkah-langkah yang dilakukan dalam melakukan penelitian. Langkah penelitian ini di jabarkan dalam bentuk diagram atupun alur langkah penelitian dan termasuk

di dalamnya algoritma, pemodelan, desain dan lain-lain yang terkait dengan aspek perancangan sistem.

2.1. Alur Pikir Penelitian



Gambar 1. Alur Pikir Penelitian

a. Penentuan Tempat

Pada tahap ini ditentukan CV. Alkenza Mandiri yang beralamat di Jl. Raya Cimareme, Cimareme Indah Blok A1/No.14 Kab. Bandung Barat sebagai tempat riset.

b. Pelaksanaan

Pada tahap pelaksanaan, dilakukan metode wawancara dan metode kepustakaan. Berikut yang dilakukan pada metode ini:

a) Pada metode wawancara penulis mewawancarai Bapak Kusnandar. selaku *GeneralManager* CV. Alkenza Mandiri.
b) Pada metode kepustakaan penulis mendatangi perpustakaan Universitas Budi Luhur.

c. Tahapan Perencanaan

Pada tahap perancanaan, Penulis merencanakan membuat aplikasi pengamanan pesan *email*, yaitu menggunakan algoritma *CaesarCipher* dan *RC4* untuk mengamankan pesan email pada CV. Alkenza Mandiri.

d. Mempertajam Fokus dan Perumusan Masalah

Pada tahap mempertajam fokus dan perumusan masalah, dilakukan pengumpulan jurnal yang mendukung Tugas Akhir yaitu jurnal tentang enkripsi data atau informasi dengan algoritma *CaesarCipher* dan *RC4*. Dilakukan pendataan tentang masalah apa saja yang bahaya dari pencurian data atau informasi pada CV. Alkenza Mandiri.

e. Analisis

Pada tahap analisis, dilakukan analisa terhadap bahaya pencurian data atau informasi serta aplikasi yang digunakan untuk mendukung perancangan sistem aplikasi pengamanan pesan *email*.

f. Desain Penelitian Rancangan Program

Pada tahap ini, membuat rancangan layar program dan alur proses kerja sistem aplikasi pengamanan pesan *email* berbasis *web* yang digunakan sebagai penelitian.

g. Pembuatan Program

Pada tahap pembuatan program, penulis menggunakan Bahasa *PHP* sebagai bahasa pemrograman dan *Apache* sebagai aplikasi *WebServer* lokal (*localhost)*.

h. Implementasi Program Pada Tempat Riset

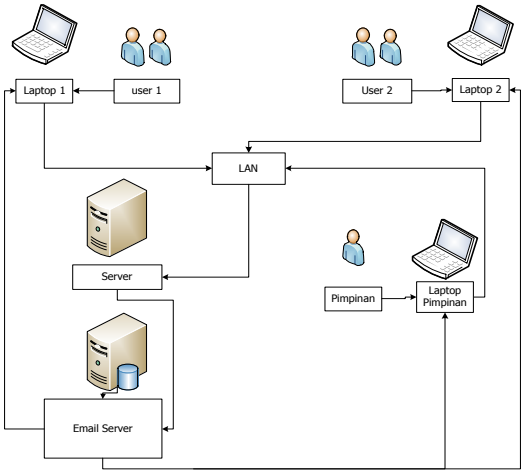
Pada tahap implementasi program pada tempat riset, dilakukan percobaan aplikasi yang dibuat pada CV. Alkenza Mandiri.

i. Simpulan Hasil Riset

Pada tahap simpulan hasil riset, Penulis menyimpulkan akan membuat aplikasi pengamanan pesan *email* berbasis *web* yang dapat mengamankan data atau infomasi yang dikirim atau diterima oleh semua divisi CV. Alkenza Mandiri.

2.2. Arsitektur Sistem

Arsitektur sistem atau skema aplikasi usulan yang akan dibangun adalah sebagai berikut:



Gambar2. Arsitektur Sistem

2.3. Rancangan Sistem

Rancangan sistem yang digunakan dalam aplikasi pengamanan pesan *email* ini adalah rancangan *Use Case Diagram.*

Use case diagram menggambarkan mengenai interaksi antara pengguna, sistem dan eksternal sistem. Adapun langkah-langkah dalam membuat *use case diagram* yaitu sebagai berikut :

1) Identifikasi *actor*.
2) Identifikasi *Use case.*
3) *Use case* diagram.
4) Narasi *Use case.*

2.4. Rancangan Layar

Agar suatu aplikasi mudah digunakan, maka diperlukan *user interface* yang dapat dengan mudah dimengerti oleh pengguna. Untuk menghasilkan *user interface* yang mudah dimengerti dan dipahami oleh pengguna, maka diperlukan rancangan layar sebelum diimplementasikan dalam bentuk program. Berikut ini adalah rancangan layar untuk aplikasi enkripsi dan kompresi.

a. Rancangan Layar *Form Login*

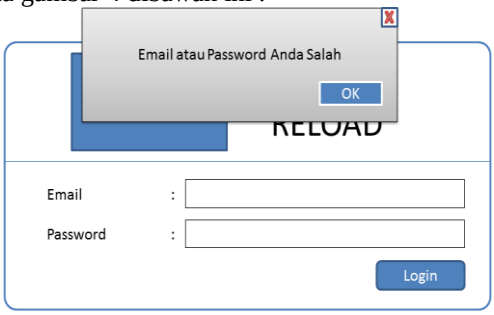
Tampilan layar *form login*, seperti terlihat pada gambar 3 dibawah ini, berfungsi sebagai akses menuju menu utama. Pada rancangan layar, disediakan menu pengisian *email* dan *password*. Tombol *login* digunakan untuk proses validasi pada

email server. Bila *email* dan *password* sesuai, maka akan tampil menu utama.

Gambar 3.Rancangan Layar Form Login

b. Rancangan Layar *Form* Salah *Login*

Proses *input login* pada gambar di atas, terdiri dari dua data yang harus dimasukkan dengan benar, yaitu *email* dan *password*. Data yang telah dimasukkan akan didefinisikan pada *email server*, bila *email* dan *password* tidak sesuai, maka menu utama tidak dapat diakses. Sehingga muncul pesan "*email* atau *pasword* anda salah". Seperti terlihat pada gambar 4 dibawah ini :



Gambar 4.Rancangan Layar Form Salah Login

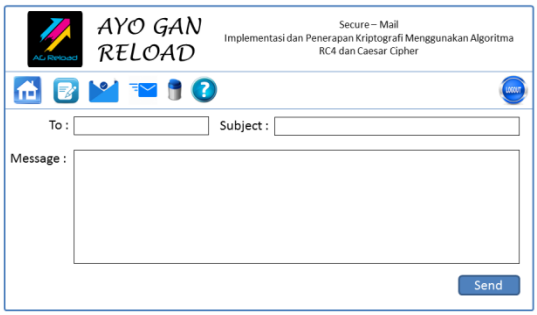
c. Rancangan Layar *Form* Home

Rancangan layar *form home* berfungsi sebagai halaman untuk menampilkan informasi singkat tentang aplikasi pengamanan email ini. Tampilan pada gambar 5 di bawah ini akan tampil setelah pengguna sukses melakukan login atau pengguna memilih menu *home*.

Gambar 5.Rancangan Layar Form Home

d. Rancangan Layar *Form* Compose

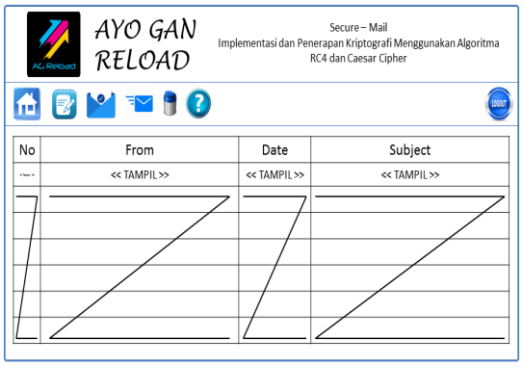
Rancangan layar *form compose* berfungsi sebagai halaman untuk membuat pesan *email* baru untuk dikirimkan ke pengguna lain. Setelah pesan *email* dibuat maka proses enkripsi akan dilakukan oleh sistem sebelum pesan *email* dikirimkan ke pengguna yang sudah ditentukan. Tampilan pada gambar 6 di bawah ini akan tampil setelah pengguna memilih menu *compose*.



Gambar 6.Rancangan Layar Form Compose

e. Rancangan Layar *Form* Inbox

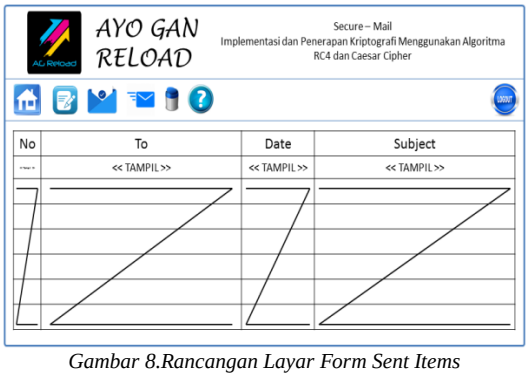
Rancangan layar *form inbox*, berfungsi sebagai halaman untuk menampilkan daftar pesan *email* yang ada didalam kotak masuk, tampilan pada gambar 7 di bawah ini akan tampil setelah pengguna memilih menu *inbox*. Apabila pengguna memilih pesan yang terenkripsi oleh sistem untuk dibuka dan dibaca, maka sistem akan melakukan dekripsi terlebih dahulu pada pesan yang dipilih agar pesan dapat tampil sesuai dengan pesan asli.



Gambar 7.Rancangan Layar Form Inbox

f. Rancangan Layar *Form* Sent Items

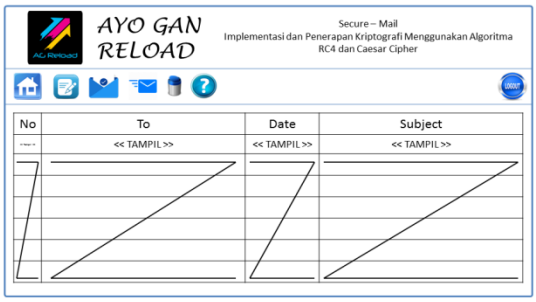
Rancangan layar *form sent items*, berfungsi sebagai halaman untuk menampilkan daftar pesan *email* yang ada didalam kotak pesan terkirim, tampilan pada gambar 8 di bawah ini akan tampil setelah pengguna memilih menu *sent items*. Apabila pengguna memilih pesan yang terenkripsi oleh sistem untuk dibuka dan dibaca, maka sistem akan melakukan dekripsi terlebih dahulu pada pesan yang dipilih agar pesan dapat tampil sesuai dengan pesan asli.



Gambar 8.Rancangan Layar Form Sent Items

g. Rancangan Layar *Form* Trash

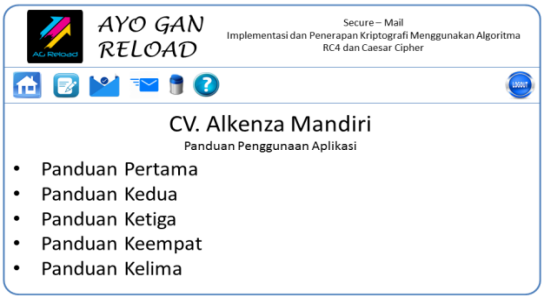
Rancangan layar *form trash*, berfungsi sebagai halaman untuk menampilkan daftar pesan *email* yang ada didalam kotak sampah, tampilan pada gambar 9 di bawah ini akan tampil setelah pengguna memilih menu *trash*. Apabila pengguna memilih pesan yang terenkripsi oleh sistem untuk dibuka dan dibaca, maka sistem akan melakukan dekripsi terlebih dahulu pada pesan yang dipilih agar pesan dapat tampil sesuai dengan pesan asli.



Gambar 9.Rancangan Layar Form Trash

h. Rancangan Layar *Form* Help

Rancangan layar *form help* berfungsi sebagai halaman untuk menampilkan informasi tentang bagaimana cara menggunakan aplikasi pengamanan *email* CV. Alkenza Mandiri. Tampilan pada gambar 10 di bawah ini akan tampil setelah pengguna memilih menu *help*.



Gambar 10.Rancangan Layar Form Help

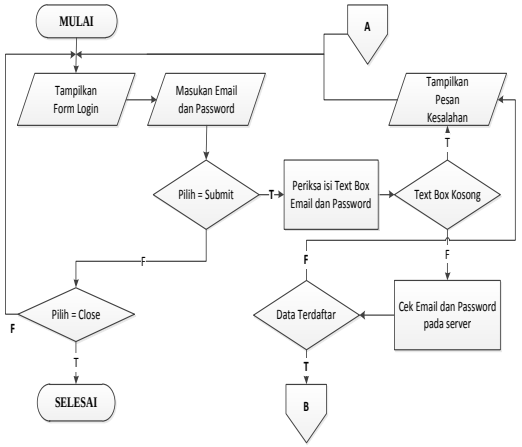
2.5. Flowchart dan Algoritma Aplikasi

Beberapa urutan-urutan proses yang harus di lalui digambarkan dalam bentuk *flowchart* dan diikuti dengan algoritma. *Flowchart* dan algoritma dari setiap proses pada sebuah halaman akan dibahas pada penjelasan berikut ini :

1. *Flowchart* dan Algoritma *Form Login*

a. *Flowchart Form Login*

Berikut ini adalah *flowchart form login* yang digunakan oleh pengguna untuk masuk kedalam aplikasi enkripsi dan kompresi



Gambar 11.Flowchart Form Login

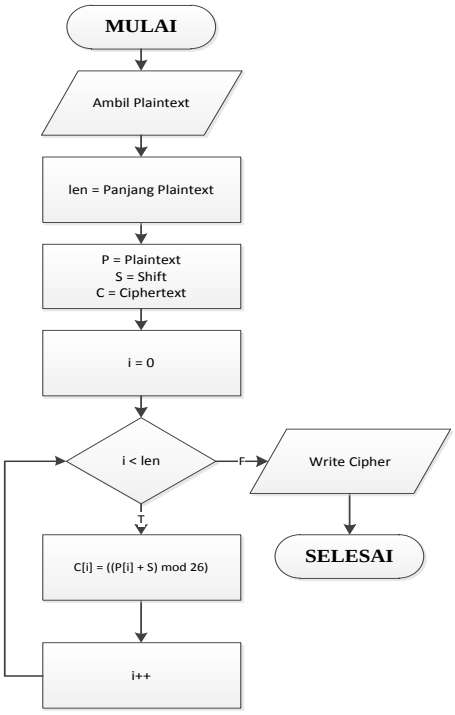
b. *Algoritma Form Login*

Berikut ini adalah algoritma pada *form login* yang menjelaskan proses pada saat pengguna akan masuk kedalam aplikasi kriptografi dan kompresi, dimana pengguna harus memasukkan *username* dan *password* pada kolom yang tersedia, lalu sistem akan mencocokkan data yang dimasukkan pengguna dengan data yang terdapat di *database*. Apabila data *login* yang dimasukkan ada dalam database, maka sistem akan membawa pengguna ke halaman *home*. Jika pengguna salah memasukkan *username* dan *password*, maka sistem akan menampilkan pesan kesalahan dan membawa pengguna kembali ke halaman *login.*

1. Tampilkan *FormLogin*
2. *InputEmail* dan *Password*
3. *Input* Pilihan
4. *If* Pilih = "*Submit*" *Then*
5. Periksa Isi *TextBoxEmail* dan *Password*
6. *IfTextBox* = "Empty" *Then*
7. Tampilkan Pesan Kesalahan
8. Kembali ke baris 1
9. *Else*
10. Cek *Email* dan *Password* di *server email*
11. *If* data sesuai *Then*
12. Tampilkan halaman *home*
13. *Else*
14. Tampilkan Pesan Kesalahan
15. Kembali ke baris 1
16. *EndIf*
17. *EndIf*
18. *Else*
19. Selesai
20. *End if*

2. *Flowchart* dan Algoritma Enkripsi *Caesar Cipher*

a. *Flowchart* Algoritma Enkripsi *Caesar Cipher*

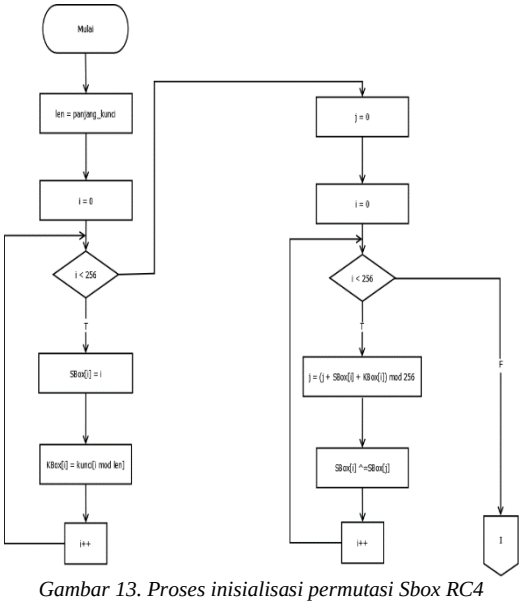


Gambar 12.Flowchart Enkripsi Caesar Cipher

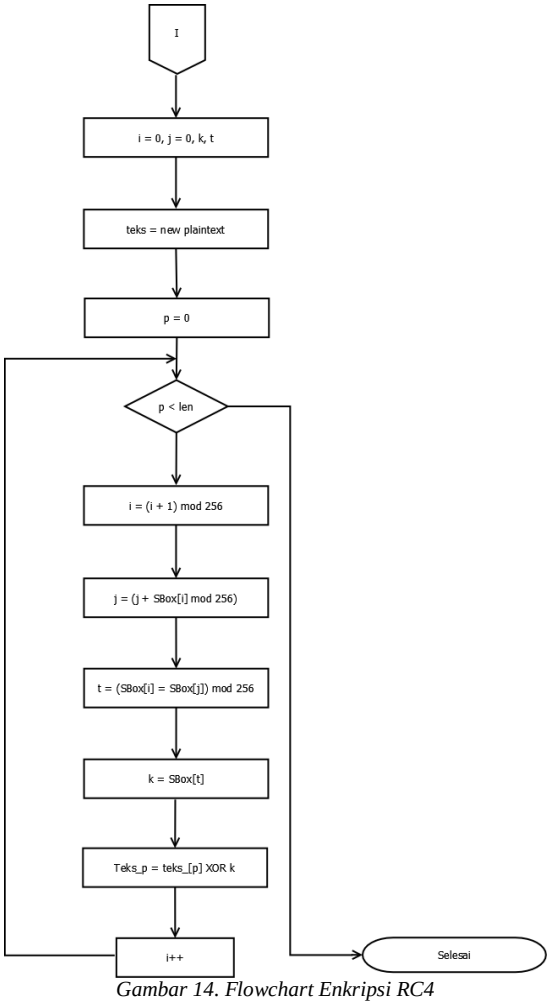
b. *Algoritma Enkripsi Caesar Cipher*

a. Start
b. Ambil Plaintext
c. len = Panjang plaintext
d. P = Plaintext ; S = Shift ; C = Ciphertext
e. i = 0
f. for (i < len)do
g. ((P[i]+S)mod26)
h. i++
i. End for
j. write (ciphertext)

3. *Flowchart* dan Algoritma Enkripsi*RC4*
   a. *Flowchart* Algoritma Enkripsi*RC4*



Gambar 13. Proses inisialisasi permutasi Sbox RC4



Gambar 14. Flowchart Enkripsi RC4

b. *Algoritma Enkripsi RC4*

Algoritma pada proses enkripsi ini menjelaskan tentang bagaimana *plaintext* diubah menjadi *ciphertext*.

Inisialisasi SBox (*Array S*) & KBox (*Array K*)yaitu *array* sepanjang 256 yang berisi permutasi dari bilangan 0 sampai 255, dan S-Box kedua, yang berisi permutasi merupakan fungsi dari kunci dengan panjang variabel

1. len = panjang_kunci
2. i = 0
3. For i < 256 Do
4. SBox[i] = i
5. KBox[i] = kunci[I mod len]
6. I++
7. End

Permutasi pada Sbox adalah Penyusunan kembali suatu kumpulan objek dalam urutan yang berbeda dari urutan yang semula.

1. j = 0
2. i = 0
3. For i < 256 Do
4. j = (j + SBox[i] + KBox[i]) mod 256

```
5. SBox[i] ^=SBox[j]
6.    I++
7. End
```

Pembentukan *pseudo random byte* dan proses enkripsi

Pseudo Random Generation Algorithm (PRGA) ini digunakan untuk mendapatkan byte acak untuk enkripsi.

```
9.  j = 0
10. i = 0
11. p = 0
12. t = ""
13. teks_ = new int[plaintext length]
14. len = plaintext length
15. For p<len Do
16.    i = (i + 1) mod256
17. j = (j + SBox[i]) mod 256
18. SBox[i] ^=SBox[j]
19. t = (SBox[i] + SBox[j]) mod 256
20. k = SBox[t]
21. teks_[p] = teks_[p] XOR k
22. I++
```

3. HASIL DAN PEMBAHASAN

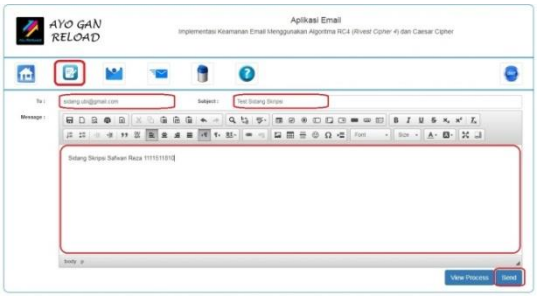
3.1. Tampilan dan Halaman Login

Halaman login pertama kali akan tampil pada saat aplikasi dijalankan. Pengguna harus mengisi *email* dan *password* yang sudah didaftarkan kemudian klik tombol*login*, sehingga dapat masuk ke dalam aplikasi. Berikut merupakan tampilan halaman login.

Gambar 15.Tampilan halaman Login

3.2. Tampilan Halaman Compose

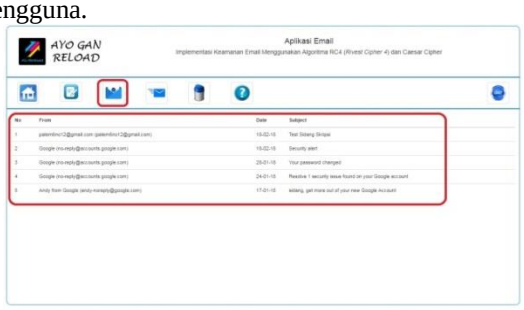
Halaman ini merupakan fungsi utama dari aplikasi.Dimana proses enkripsiakan dilakukan pada saat pengguna menekan tombol kirim. Proses enkripsi akan berjalan dan menghaslkan *ciphertext.* Lalu sistem akan mengirimkan *email* dengan isi yang telah dienkripsi tersebut ke *email* tujuan. Dan pengguna dapat membuka *email* tersebut dengan aplikasi. Apabila *email* dibuka tanpa aplikasi, maka isi *email* yang akan ditampilkan adalah *ciphertext*yang tidak didekripsi. Sehingga isi *email* tidak dapat terbaca.Berikut merupakan tampilan halaman *compose*.



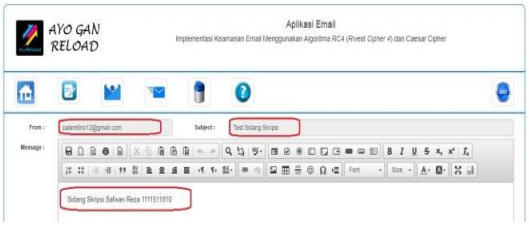
Gambar 16.Tampilan halaman Compose

3.3. Tampilan Halaman Inbox dan Read Mail

Untuk dapat melihat isi *email* yang telah didekripsi, pengguna diharuskan membuka *email* yang dimaksud dengan aplikasi, karena jika dibuka tanpa aplikasi, maka *email* yang telah dienkripsi tidak akan melalui proses dekripsi dan yang akan ditampilkan adalah isi *email* yang berupa *ciphertext* sehingga penerima tidak akan dapat membaca isi *email* tersebut. Berikut adalah tampilan halaman *inbox* yang berisikan daftar *email* yang diterima pengguna.



Gambar 17.Tampilan halaman Inbox



Gambar 18.Tampilan halaman Read Mail

Dan berikut adalah tampilan halaman *reademail* yang menampilkan isi *email* yang dibuka tanpa aplikasi yang tidak melalui proses dekripsi.

Gambar 19.Tampilan halaman Read Mail tanpa Aplikasi

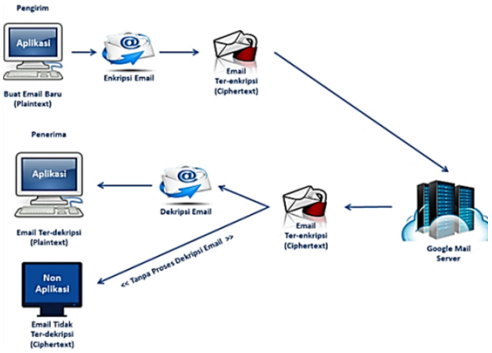
3.4. Perancangan Aplikasi

Aplikasi yang dibuat dari 5 Halaman, yaitu Halaman *Inbox*, Halaman *Sent Mail*, Halaman *Trash*,

Halaman *Compose* dan Halaman *Help*. Untuk proses enkripsi akan dilakukan sistem pada saat pengguna melakukan pengiriman *email*, dan akan diterima oleh pengguna yang dituju dalam keadaan ter-enkripsi. Pengguna yang menerima *email*, harus membuka *email* tersebut dengan aplikasi, agar isi *email* bisa terbaca.Apabila pengguna yang menerima *email* membuka *email* tersebut tanpa aplikasi, maka isi *email* yang dibuka adalah hasil enkripsi

3.5. Skema Proses Sistem Aplikasi

Secara umum, rancangan program yang akan dibuat dapat dilihat pada gambar berikut ini:



Gambar 20. Skema ProsesSistem Aplikasi

Enkripsi merupakan proses mengubah plainteks menjadi kode-kode yang sulit dimengerti. Adapun langkah-langkah pada proses enkripsi dan dekripsi yang dapat dilihat sebagai berikut :

1) Pengguna melakukan input informasi atau data yang ingin diamankan dengan cara membuat *email* baru.
2) Pengguna mengirimkan *email* ke penerima yang sudah ditentukan.
3) Proses enkripsiakan dilakukan pada saat pengguna menekan tombol kirim. Dilanjutkan dengan mengirimkan *email* yang telah ter-enkripsi ke alamat *email* penerima yang sudah ditentukan.
4) Penerima dapat membaca isi email yang diterima dengan membuka email yang diterima dengan aplikasi. Karena proses dekripsi akan dilakukan pada saat penerima email membuka email dari pengirim. Apabila penerima email membuka email tanpa aplikasi, yang akan ditampilkan adalah email yang ter-enkripsi.

4. KESIMPULAN

Berdasarkan pengkajian aplikasi terhadap masalah dan penyelesaian yang telah dilakukan, maka ditarik kesimpulan dan saran yang akan diperlukan untuk pengembangan sistem ini ke tahap lebih lanjut. Hal ini untuk menjadikan aplikasi yang dibuat lebih sempurna.

4.1 Kesimpulan

Berdasarkan dari uraian bab sebelumnya terhadap permasalahan dan aplikasi yang telah dikembangkan, maka dapat ditarik kesimpulan mengenai proses enkripsi dan dekripsi terhadap masalah keamanan data perusahaan, antara lain :

a. Dengan adanya aplikasi yang dibangun ini, informasi internal perusahaan yang dikirim atau diterima melalui email menjadi lebih aman, sehingga tidak timbul kekhawatiran informasi internal perusahaan akan dicuri atau disalahgunakan oleh pihak lain yang tidak bertanggung jawab.
b. Aplikasi yang dibangun berbasis web dengan metode pengembang waterfall, bahasa pemrograman PHP.
c. Aplikasi mampu memberikan keamanan data yang dimiliki oleh perusahaan.

4.2 Saran

Aplikasi enkripsi dan dekripsi *email* pada CV. Alkenza Mandiri ini masih memiliki beberapa keterbatasan dan kekurangan, sehingga untuk itu penulis menyarankan untuk pengembangan aplikasi selanjutnya agar :

a. Aplikasi dapat mengirimkan *file* yang di *attach* melalui aplikasi ini.
b. Aplikasi bisa mendukung banyak jenis akun layanan *email.*
c. Pemilihan algoritma enkripsi untuk digunakan sebagai algoritma keamanan dengan mempertimbangkan tingkat keamanan yang lebih tinggi lagi
d. Memilih personil yang bertanggung jawab dan kompeten sesuai dengan kebutuhan aplikasi demi menjaga keamanan aplikasi.

5. DAFTAR PUSTAKA

[1] Amin, M Miftakul 2016, Implementasi Kriptografi Klasik Pada Komunikasi Berbasis Teks, Jurnal Pseudocode, Volume III Nomor 2, September 2016, Palembang, hal. 129-136.
[2] Ariyus, Dony 2008, Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi, Yogyakarta, Andi Offset.
[3] Burton, David M, 2002, Elementary Number Theory Fifth Edition, New York, McGraw-Hill.
[4] Cormen, Thomas H, 1989, Introduction to Algorithms, Cambridge: The MIT Press.
[5] David, Salomon 2007, Data Compression, The Complete Reference, Forth Edition, Northridge, Springer Publishers.
[6] Department of Computer & Information System Engineering, 2012, Practical Workbook, Information Theory, 4th edition, Karachi, Pakistan, NED University of Engineering & Technology
[7] Diffie, Whitfield, Martin E Hellman, 1976, New Directions in Cryptography, IEEE Trans, Info, Theory IT-22.
[8] Menezes, Oorschot, Vanstone, 1996, Handbook Of Applied Cryptography, Canada, CRC Press.
[9] Munir, Rinaldi, 2006, Kriptografi, Bandung, Penerbit Informatika.
[10] Rifai, Yunan Rizal, Christyono, Yuli & Santoso, Imam, 2016, Implementasi Algoritma

Kriptografi Rivest Code 4, Rivest Shamir Adleman Dan Metode Steganografi Untuk Pengamanan Pesan Rahasia Pada Berkas Teks Digital, Transient, Vol. 5, No. 1, Maret 2016, Semarang.

[11] Sadikin, Rifki, 2012, Kriptografi Untuk Keamanan Jaringan, Yogyakarta, Andi Offset.

[12] Setiawan, Okie, Fiati, Rina & Listyorini, Tri, 2014, Algoritma Enkripsi RC4 Sebagai Metode Obfuscation Source Code PHP, Prosiding SNATIF Ke-1, Tahun 2014, Kudus, hal. 113-120.

[13] Simamora, Dedi Putra Oloan, 2017, Implementasi Algoritma RC4 Dan Playfair Cipher Untuk Mengamankan Data Teks, Jurnal Pelita Informatika, Volume 16, Nomor 3, Juli 2017, Medan, hal. 328-334.

[14] Sommerville, Ian 2011, Software Engineering, Boston, Pearson Education.

[15] Triyuswoyo, Y & Ferdianti, F, 2014, Implementasi Algoritma Caesar, Cipher Disk dan Scytale pada Aplikasi Enkripsi dan Dekripsi Pesan Singkat, Prosiding Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2014), Depok, hal. 467-472.

[16] Zuli, Faisal & Irawan, Ari, 2014, Penerapan Kombinasi Sandi Caesar Dan Vigenere Untuk Pengamanan Data Pesan Pada Surat Elektronik, Jurnal Sistem Informasi, 7(2), 2014, Jakarta, hal. 1-11.