Sistem Pengaman File dengan Menggunakan Metode RSA Kriptografi & Digital Signature

Achmad Zakki Falani, Muchammad Zunaidy

Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Narotama
zakki_falani@yahoo.com, edy.joen06@gmail.com

Abstrak

Semakin pesatnya perkembangan teknologi informasi dewasa ini tidak akan pernah lepas dari isu keamanan komputer (*computer security*). Keamanan komputer sebagai suatu permasalahan yang cukup kompleks selalu menuntut untuk dilakukannya update sistem secara berkala untuk memperkuat keamanan. Seperti pada kasus transfer file-file penting melalui jalur komunikasi internet, sebagaimana file-file tersebut bisa dengan aman melewati jalur tersebut ketika ditransfer ketujuan sedangkan jalur komunikasi internet itu sendiri mempunyai sifat *vulnerable* yang artinya sangat rentan terhadap serangan dari luar.

Permasalahan yang kemudian mucul adalah bagaimana agar file tersebut tidak bisa dibaca seandainya jatuh ke tangan orang lain yang tidak berkepentingan, bagaimana bisa diketahui keaslian isi dari file setelah diterima oleh yang bersangkutan.

Tujuan dari perancangan sistem ini adalah mencoba memberikan solusi dari kedua permasalahan tersebut. Sistem yang dirancang akan mampu melakukan pengamanan secara internal yang artinya pengamanan dilakukan dari dalam file itu sendiri. Proses - proses dalam sistem akan mencakup pengamanan data menggunakan teknik kriptografi yaitu enkripsi dan dekripsi yang merupakan proses untuk menyamarkan isi file menggunakan metode kriptografi kunci publik RSA, kemudian akan dilakukan pengecekan keaslian isi dari file menggunakan teknik tanda tangan digital dengan metode RSA *Digital Signature*.

Kata kunci : Enkripsi, Dekripsi, *Digital Signature*.

1.1 Latar Belakang

Semakin pesatnya perkembangan teknologi informasi (TI) dewasa ini,tidak akan pernah lepas dari permasalahan keamanan computer *(Computer Security).* Keamanan computer sebagai isu yang tidak akan pernah habis dibicarakan para pelaku bidang TI selalu menuntut adanya *update* setiap saat dan berkala. Namun hal yang tidak kalah penting dari permasalahan keamanan computer tersebut adalah elemen-elemen yang ada dalamnya.

Seperti pada kasus pengiriman file.pengiriman file via internet merupakan cara yang paling praktis di era teknologi informasi dewasa ini. Karena bisa dilakukan dengan mudah, maka aspek-aspek keamanan dalam proses pengirimannya perlu diperhatikan. Dalam *browser* internet sendiri secara *default* memang sudah ditanamkan aspek-aspek keamanan, bahkan dalam protocol internet yang umum digunakan yaitu TCP/IP,sudah ada semacam metode enkripsi data sebelum data dikirim. Namun aspek-aspek keamanan tersebut tetap saja belum bisa menghentikan para *cracker/hacker* untuk melakukan teknik *deception (man in the middle)*, yaitu suatu teknik untuk mendapatkan data dengan cara melakukan pengelabuan seakan-akan dia adalah orang yang dituju dalam pengiriman data. Bila teknik ini berhasil dilakukan, maka sudah bisa dipastikan bahwa data akan jatuh ke tangan *cracker/hacker* dan dengan mudah dapat dibaca.

Dalam dunia TI integritas suatu data yang dikirim terkadang juga menjadi pertanyaan, apakah data tersebut benar-benar dikirim oleh orang yang bersangkutan atau tidak, dan apakah isi dari data benar-benar otentik seperti sebelum dikirim. Hal ini merupakan masalah serius karena bisa saja sesorang mengirimkan data palsu. Seperti pada kasus *e-mail spam*, *inbox* akan penuh apabila *e-mail – e-mail spam* tersebut tidak dihapus. Dari hal itu maka diperlukan suatu cara untuk mengecek integritas suatu data agar data tersebut benar-benar otentik seperti waktu sebelum dikirim.

Permasalahan-permasalahan diatas merupakan masalah utama dalam hal kemanan data. Dan hal ini tersebut melatarbelakangi Penulis untuk mencoba merancang suatu program yang dapat menjadi solusi permasalahan tersebut. Penulis merancang sebuah perangkat lunak pengaman file yang menggabungkan beberapa teknik, antara lain kriptografi menggunakan algoritma RSA (Rivest, Shammer, Adelman) untuk masalah penyamaran isi file, tanda tangan digital *(Digital Signature)* untuk masalah integritas data.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan tersebut dapat diketahui bahwa sebetulnya ada permasalahan yang perlu dicari solusinya. Dan diperlukan suatu analisa yang matang agar program yang akan dirancang bisa berjalan dengan akurat dan sesuai dengan apa yang diharapkan. Dengan demikian perumusan masalahnya dapat diusai sebagai berikut:

1. Bagaimana menyamarkan isi file melalui metode kriptografi agar file tersebut tidak bisa dibaca oleh orang lain yang tidak berkepentingan/tidak mempunyai kuncinya?
2. Bagaimana integritas file dikirim, apakah file tersebut benar-benar dari orang yang mengirim atau bukan, serta apakah isi dari file benar-benar otentik?

1.3 Batasan Masalah

Agar pembahasan terhadap permasalahan lebih terarah, maka Penulis membatasi permasalahan dalam ruang lingkup pengamanan file dan cara kerja program secara dasar. Ruang lingkup permasalahan tidak akan membahas proses pengiriman file ke tujuan, karena cara kerja program secara dasar adalah pengolahan file.

1. Enkripsi menggunakan algoritma kriptografi RSA.
2. Tanda tangan digital mnggunakan algoritma *digital signature* RSA.
3. Verifikasi tanda tangan digital menggunakan algoritma *digital signature* RSA.
4. Deskripsi menggunakan algoritma kriptografi RSA.
5. Pengamanan e-dokumen hanya meliputi : kerahasiaan yaitu dekripsi terhadap tandatangan *digital* ,autentikasi *signer* dan integritas file sebagai hasil verifikasi pada pihak *verifier.*
6. Tidak membahas aspek keamanan pada jalur komunikasi yaitu pada proses transmisi file lewat internet via *email*, keamanan yang bersifat fisik dan keamanan yang berhubungan dengan *personal* .
7. Tidak ada proses validasi dari pihak penjamin,yaitu infrastruktur kunci publik manapun.

1.4 Tujuan

Dengan berdasarkan pada alas an pemilihan judul/topic tersebut, maka tujuan dari pembuatan program ini adalah untuk melakukan pengamanan terhadap file yang akan dikirim/didistribusikan melalui jalur komunikasi internet/jaringan computer local agar file tersebut tidak bisa dibaca oleh orang lain yang tidak berkepentingan/tidak mempunyai kuncinya, dapat diketahui juga integritas dari file tersebut apakah benar-benar dikirim oleh orang yang benar serta pemeriksaan otentik atau tidaknya file tersebut.

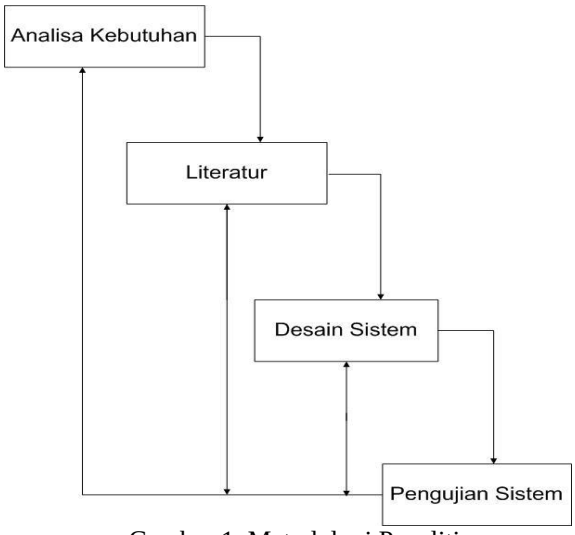
1.5 Manfaat Penelitian

Manfaat yang bisa di dapat dari adanya penelitian ini adalah :

1. Memberikan manfaat yang besar terhadap para professional di bidang IT khususnya dan diberbagai bidang umumnya dalam pengiriman berkas data-data penting yang bersifat rahasia via internet.
2. Memberikan pengamanan terhadap berkas-berkas data perusahaan yang saling dikirimkan antara kantor pusat dan kantor cabang.
3. Memberikan verifikasi keabsahan data yang dikirim dari pihak penerima maupun pihak pengirim.
4. Memberikan kemudahan kepada pihak penerima untuk mengetahui tentang siapa yang mengirim data tersebut.

1.6 Metodologi Penelitian

Dalam pengerjaan penelitian ini meliputi langkah-langkah sebagai berikut:



Gambar 1. Metodologi Penelitian

1. Analisa kebutuhan
   Yaitu teknik pengumpulan data dengan melakukan tanya jawab kepada orang-orang yang dianggap mengerti/paham terhadap permasalahan ini dan yang dianggap dapat memberikan informasi yang diperlukan perancangan system.
2. Literatur
   Yaitu teknik perancangan sistem dan penyusunan laporan dengan berpedoman pada literature-

literatur yang dianggap mendukung. Literatur tersebut berupa buku,e-book,dan data-data dari internet.

3. Desain sistem
   Yaitu teknik perancangan system yang mengacu pada desain baik alur program maupun desain tampilan dari program. Teknik desain system ini melakukan perbandingan dengan system-sistem yang lain mengenai alur program serta desain tampilan yang tepat.
4. Pengujian sistem
   Yaitu teknik perancangan system dengan terlebih dahulu melakukan percobaan implementasi terhadap metode-metode yang akan dipakai dalam program dengan tujuan agar metode tersebut benar-benar bisa diimplementasikan dalam system nantinya.

2.1 Kriptografi

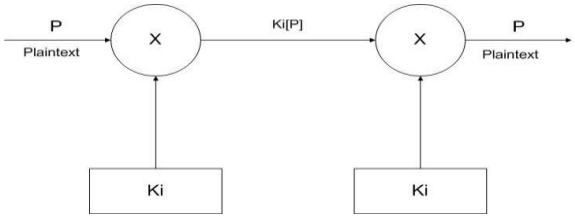
Kriptografi merupakan suatu bidang ilmu tentang cara menuliskan pesan dalam kode rahasia, atau bisa dikatakan bahwa kriptografi merupakan ilmu untuk menjaga keamanan pesan. Pertama kali digunakan kira-kira 1900 tahun SM di Mesir,dimana ketika itu orang Mesir menulis menggunakan huruf *hieroglyph* yang tidak standart dalam sebuah prasasti.

Kriptografi merupakan suatu cara untuk melakukan pengamanan data pada proses pengirimannya. Sebelum dikirim, data merupakan informasi yang dapat dibaca dan disebut *plaintext.* Sebelum dikirimkan, data akan melewati suatu proses dimana proses ini bertujuan untuk membuat data tidak dapat dibaca oleh orang lain,yang disebut Enkripsi. Dan data yang tidak bisa dibaca ini disebut *Ciphertetxt*. Setelah data diterima, maka akan dilakukan proses yang merupakan kebalikan dari proses enkrpsi, yaitu Deskripsi.

Orang yang ahli dibidang kriptografi disebut Kriptografer, dan orang yang berusaha untuk memecahkan *ciphertext* menjadi *plaintext* tanpa melalui cara yang seharusnya disebut Kriptanalis. Sedangkan seni untuk memecahkan kode rahasia diisebut Kriptanalis.

2.2 Algoritma Kriptografi Kunci Simetris

Algoritma kriptografi kunci simestris disebut juga sebagai algoritma konvensional. Merupakan algoritma yang menggunakan kunci enkripsi yang sama dengan kunci dekripsinya. Disebut konvensional karena algoritma ini telah biasa digunakan sejak berabad-abad yang lalu. Algoritma ini mengharuskan pengirim dan penerima menyetujui suatu kunci tertentu sebelum mereka dapat berkomunikasi dengan aman. Jadi keamanannya tergantung pada kunci. Membocorkan kunci berarti mempersilahkan orang lain untuk meng-enkripsi dan men-dekripsi pesan. Yang termasuk dalam algoritma kunci simetris adalah OTP, DES, RC2, RC4, RC5, RC6, IDzEA, Twofish, Magenta, LOKI, Rijndael, Blowfish, dan lain-lain.



Gambar 2. Sistem Kunci Simetris

2.3 RSA Digital Signature

RSA *Digital Signature* merupakan bagian dari Algoritma Kriptografi Kunci *Public* RSA. Meskipun algoritma ini tidak menjadi standart *digital signature*, skema dari algoritma ini telah membuktikan bahwa algoritma ini mempunyai keandalan yang sama dengan standart *digital signature* DSA. Algoritma RSA *Digital Signature* ini bersifat fleksibel dengan beberapa metode *hash* (*hash* : pencarian nilai dari hasil pengolahan data secara matematis, dengan panjang hasil yang selalu sama), artinya RSA *Digital Signature* bisa menggunakan salah satu metode hash dalam prosesnya, antara lain MD4, MD5, RIPEMD, dll. Namun yang akan diimplementasikan dalam program ini adalah metode *hash* RIPEMD.

2.3.1 RIPEMD-160

RIPEMD-160 (*RACE Integrity Primitives Evaluation Message Digest*) merupakan algoritma perpendekan pesan (*Message Digest*) 160 bit dan juga merupakan fungsi *hash*. Dikembangkan di Eropa oleh Hans Dobbertin, Antoon Bosselaers dan Bart Preneel. Pertama kali dipublikasikan pada tahun 1996. RIPEMD-160 merupakan pengembangan dari versi awalnya yaitu RIPEMD. Dimana fuungsi *hash* ini dikembangkan berdasarkan prinsip yang digunakan dalam MD4.

Keseluruhan proses dari RIPEMD-160 adalah mengolah input 21-word (5-word *variable chaning* ditambah 16-word blok pesan, dimana 1 word=32 bit) menjadi output 5 word. Setiap blok input diproses secara parallel dengan versi fungsi kompresi yang berbeda (*left line* dan *right line*). Output 160 bit dari line yang berbeda dikombinasikan untuk menghasilkan output 160 bit yang baru.

Tabel 1 Akses order dan perhitungan rotasi RIPEMD – 160

Variable	Nilai
$z_L[0..15]$	[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
$z_L[16..31]$	[7, 4, 13, 1, 10, 6, 15, 3, 12, 0, 9, 5, 2, 14, 11, 8]
$z_L[32..47]$	[3, 10, 14, 4, 9, 15, 8, 1, 2, 7, 0, 6, 13, 11, 5, 12]
$z_L[48..63]$	[1, 9, 11, 10, 0, 8, 12, 4, 13, 3, 7, 15, 14, 5, 6, 2]
$z_L[64..79]$	[4, 0, 5, 9, 7, 12, 2, 10, 14, 1, 3, 8, 11, 6, 15, 13]
$z_R[0..15]$	[5, 14, 7, 0, 9, 2, 11, 4, 13, 6, 15, 8, 1, 10, 3, 12]
$z_R[16..31]$	[6, 11, 3, 7, 0, 13, 5, 10, 14, 15, 8, 12, 4, 9, 1, 2]
$z_R[32..47]$	[15, 5, 1, 3, 7, 14, 6, 9, 11, 8, 12, 2, 10, 0, 4, 13]
$z_R[48..63]$	[8, 6, 4, 1, 3, 11, 15, 0, 5, 12, 2, 13, 9, 7, 10, 14]
$z_R[64..79]$	[12, 15, 10, 4, 1, 5, 8, 7, 6, 2, 13, 14, 0, 3, 9, 11]
$s_L[0..15]$	[11, 14, 15, 12, 5, 8, 7, 9, 11, 13, 14, 15, 6, 7, 9, 8]
$s_L[16..31]$	[7, 6, 8, 13, 11, 9, 7, 15, 7, 12, 15, 9, 11, 7, 13, 12]
$s_L[32..47]$	[11, 13, 6, 7, 14, 9, 13, 15, 14, 8, 13, 6, 5, 12, 7, 5]
$s_L[48..63]$	[11, 12, 14, 15, 14, 15, 9, 8, 9, 14, 5, 6, 8, 6, 5, 12]
$s_L[64..79]$	[9, 15, 5, 11, 6, 8, 13, 12, 5, 12, 13, 14, 11, 8, 5, 6]
$s_R[0..15]$	[8, 9, 9, 11, 13, 15, 15, 5, 7, 7, 8, 11, 14, 14, 12, 6]
$s_R[16..31]$	[9, 13, 15, 7, 12, 8, 9, 11, 7, 7, 12, 7, 6, 15, 13, 11]
$s_R[32..47]$	[9, 7, 15, 11, 8, 6, 6, 14, 12, 13, 5, 14, 13, 13, 7, 5]
$s_R[48..63]$	[15, 5, 8, 11, 14, 14, 6, 14, 6, 9, 12, 9, 12, 5, 15, 8]
$s_R[64..79]$	[8, 5, 12, 9, 12, 5, 14, 6, 8, 13, 6, 5, 15, 13, 11, 11]

a. Melakukan inisialisasi terhadap H. H[i] = h[i], untuk $1 \le I \le 5$.
   $H_1 = h_1$
   $H_2 = h_2$
   $H_3 = h_3$
   $H_4 = h_4$
   $H_5 = h_5$
b. Proses untuk tiap I dari 0 → m-1, salin blok ke-I dari 16-word input ke penyimpanan sementara: X[j] = x_{16+j}, $0 \le j \le 15$ sehingga : Mengeksekusi 5 kali 16 langkah secara parallel left line dan right line. Bila untuk life line fungsi round nya secara urut: f, g, h, k, l. Sedangkan untuk right line menggunakan fungsi round kebalikan dari left line yaitu: l, k. h, g, f. Berikut proses left line:
   (A_L, B_L, C_L, D_L, E_L) ← (H_1, H_2, H_3, H_4, H_5)
   (left line round 1) untuk j from 0 to 15:
   t ← (A_L + f (B_L, C_L, D_L) + X[$z_L[j]$] + $y_L[j]$),
   (A_L, B_L, C_L, D_L, E_L) ← (E_L, E_L + (t ← s_L [j], B_L, C_L, ← 10, D_L).
   (left line round 2) untuk j from 16 to 31:
   t ← (A_L + g (B_L, C_L, D_L) + X[$z_L[j]$] + $y_L[j]$),
   (A_L, B_L, C_L, D_L, E_L) ← (E_L, E_L + (t ← s_L [j], B_L, C_L, ← 10, D_L).
   (left line round 3) untuk j from 32 to 47:
   t ← (A_L + h (B_L, C_L, D_L) + X[$z_L[j]$] + $y_L[j]$),
   (A_L, B_L, C_L, D_L, E_L) ← (E_L, E_L + (t ← s_L [j], B_L, C_L, ← 10, D_L).
   (left line round 4) untuk j from 48 to 64:
   t ← (A_L + k (B_L, C_L, D_L) + X[$z_L[j]$] + $y_L[j]$),
   (A_L, B_L, C_L, D_L, E_L) ← (E_L, E_L + (t ← s_L [j], B_L, C_L, ← 10, D_L).
   (left line round 5) untuk j from 65 to 79:
   t ← (A_L + l (B_L, C_L, D_L) + X[$z_L[j]$] + $y_L[j]$),
   (A_L, B_L, C_L, D_L, E_L) ← (E_L, E_L + (t ← s_L [j], B_L, C_L, ← 10, D_L).
3. Proses untuk right line :
   (A_R, B_R, C_R, D_R, E_R) ← (H_1, H_2, H_3, H_4, H_5)
   (right line round 1) untuk j from 0 to 15:
   t ← (A_R + l (B_R, C_R, D_R) + X[$z_R[j]$] + $y_R[j]$),
   (A_R, B_R, C_R, D_R, E_R) ← (E_R, E_R + (t ← s_R [j], B_R, C_R, ← 10, D_R).
   (right line round 1) untuk j from 0 to 15:
   t ← (A_R + l (B_R, C_R, D_R) + X[$z_R[j]$] + $y_R[j]$),
   (A_R, B_R, C_R, D_R, E_R) ← (E_R, E_R + (t ← s_R [j], B_R, C_R, ← 10, D_R).
   (right line round 2) untuk j from 16 to 31:
   t ← (A_R + k (B_R, C_R, D_R) + X[$z_R[j]$] + $y_R[j]$),
   (A_R, B_R, C_R, D_R, E_R) ← (E_R, E_R + (t ← s_R [j], B_R, C_R, ← 10, D_R).
   (right line round 3) untuk j from 32 to 47:
   t ← (A_R + h (B_R, C_R, D_R) + X[$z_R[j]$] + $y_R[j]$),
   (A_R, B_R, C_R, D_R, E_R) ← (E_R, E_R + (t ← s_R [j], B_R, C_R, ← 10, D_R).
   (right line round 4) untuk j from 48 to 64:
   t ← (A_R + g (B_R, C_R, D_R) + X[$z_R[j]$] + $y_R[j]$),
   (A_R, B_R, C_R, D_R, E_R) ← (E_R, E_R + (t ← s_R [j], B_R, C_R, ← 10, D_R).
   (right line round 5) untuk j from 65 to 79:
   t ← (A_R + f (B_R, C_R, D_R) + X[$z_R[j]$] + $y_R[j]$),
   (A_R, B_R, C_R, D_R, E_R) ← (E_R, E_R + (t ← s_R [j], B_R, C_R, ← 10, D_R).
3. Setelah melakukan eksekusi left line dan right line, berikutnya adalah melakukan update nilai berantai (*chaining values*). Yaitu :
   t ← H_1
   H_1 ← H_2 + C_L + D_R
   H_2 ← H_3 + C_L + E_R
   H_3 ← H_4 + E_L + A_R
   H_4 ← H_5 + A_L + B_R
   H_5 ← t + B_L + C_R
e. Nilai hash adalah rangkaian dari: $H_1 \| H_2 \| H_3 \| H_4 \| H_5$

3.1 Analisa Sistem

Dari hasil analisa cara kerja algoritma kriptografi RSA dan digital signature terdapat beberapa permasalahan yang perlu diperhatikan antara lain:

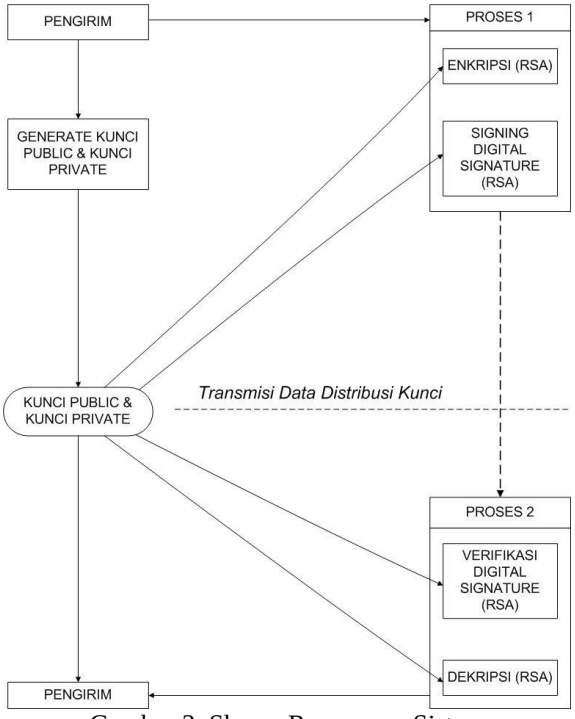
1. Pada RSA kriptografi kunci public ada satu proses untuk membuat kunci 4ublic dan kunci private. Begitu juga pada RSA digital signature juga terdapat proses untuk membuat kunci 4ublic dan kunci private. Dari permasalahan diatas dapat ditarik kesimpulan bahwa untuk masing-masing metode memerlukan pembuatan kunci masing-masing. Hal ini dirasa kurang effisien dan rumit jika harus ada 2 proses untuk membuat kunci masing-masing,yaitu 2 kunci 4ublic dan 2 kunci private mengingat kedua algoritma tersebut adalah sama-sama algoritma kunci

5ublic RSA dan dengan proses generate kunci yang sama. Dari hasil analisa tersebut penulis mencoba merancang agar pembuatan kunci cukup dilakukan dalam satu proses saja dengan generate kunci yang tidak terbatas namun pembuatan kunci ini sudah mewakili kunci 5ublic dan kunci private baik pada RSA kripografi kunci piblik dan RSA digital signature.

2. Jika isi dari data / file yang dikirimkan ternyata terkorupsi sewaktu proses pengirimannya, maka besar kemungkinan proses dekompresi akan menghasilkan isi file yang tidak valid dan kemungkinan besar file gagal diverifikasi dan tidak bias di dekripsi.

3.2 Skema Rancangan Sistem

Alur dari sistem dimualai dari pengrim yang men-genarate kunci, kemudian setelah kunci berhasil dibuat maka pengirim akan menggunakan kunci tersebut untuk melakukan Enkripsi dan signing. Jika sudah dilakukan maka data beserta kunci dikirimkan ke penerima. Jika sudah diterima maka perima akan melakukan proses Verifikasi dan Dekripsi pesan menggunakan kunci private.



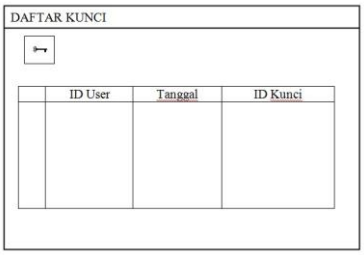
Gambar 3. Skema Rancangan Sistem

4.1 Generate dan Menampilkan Daftar Kunci

Dalam Proses Generate Kunci akan ditampilkan window yang berisi antara lain kotak-kotak edit yang harus diisi agar proses Generate Kunci dapat dilakukan. Kotak-kotak edit tersebut antara lain kotak edit ID user yang harus diisi id user yang akan melakukan generate kunci, kotak edit password yang harus diisi untuk melengkapi keamanan ID user, serta kotak edit Tulis Ulang Password untuk memastikan bahwa password tidak salah ketik.

Gambar 4. Form Generate Kunci

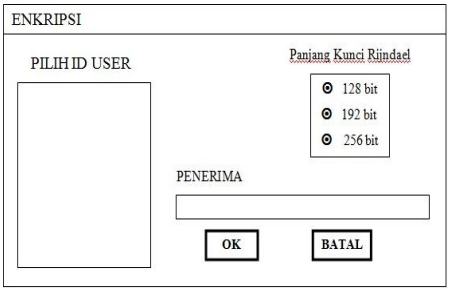
Pada proses daftar kunci ini akan menampilkan window yang berisi semua kunci yang telah dibuat dalam satu file kunci terdiri dari ID user, Tanggal pembuatan kunci dan ID kunci itu sendiri, serta dalam dalam window ini juga ditambahkan fungsi Generate Kunci baru yang merupakan refrensi dari proses generate kunci.



Gambar 5. Form Daftar Kunci

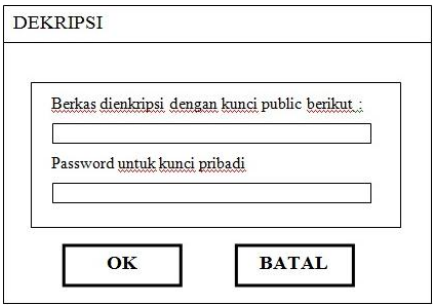
4.2 Enkripsi dan Dekripsi

Proses Enkripsi merupakan proses untuk menyamarkan isi dari suatu file dengan menggunakan teknik kriptografi. Yang nantinya hasil dari proses ini adalah terciptanya file baru yang merupakan hasil enkripsi dengan menampilkan Pilihan ID user beserta Penerima.



Gambar 6. Form Enskripsi

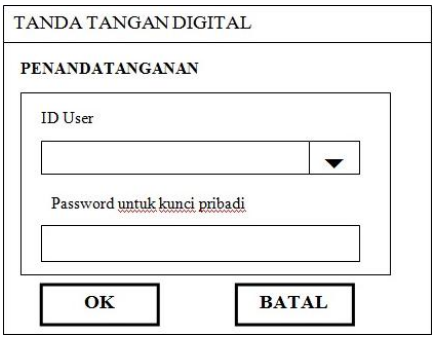
Proses Dekripsi ini adalah proses untuk membuka kunci dari file yang ter-enkripsi, sehingga file dapat dikembalikan seperti saat sebelum di-enkripsi pertama. Apabila file sudah dipilih, maka akan ditampilkan window dekripsi. Secara otomatis program akan mendeteksi kunci mana yang digunakan utk enkripsi, dan user hanya tinggal masukkan password dari kunci tersebut.



Gambar 7. Form Deskripsi

4.3 Signing dan Verifikasi Digital Signature

Proses ini merupakan suatu proses untuk membauat tanda tangan digital dari satu pesan dan proses ini muncul pada saat kita setelah memilih berkas yang akan kita tanda tangani. Kemudian pada proses Signing, User diminta untuk memilih kunci mana yang akan digunakan untuk menandatangai pesan dan user juga harus mengisi password karena kunci yang dipakai untuk signing adalah kunci private.



Gambar 8. Form Signing / Penandatanganan

Proses verifikasi ini menampilkan berkas yang sudah kita tanda tangani dengn melihat ID user dan memasukkan password , jika proses pemulihan berkas asli tidak mengalami perubahan isi ditengah jalan maka proses verifikasi tersebut akan menampilkan emotion sukses pengiriman atau mengalami percobaan perubahan berkas.

Gambar 9. Form Verifikasi

Daftar Pustaka

A.Menezes, P. Van Oorschot,S. Vanstone. 1996. *Handbook of Applied Cryptography*, CRC Press Inc.

http://budi.insan.co.id/courses/ec5010/handbook.pdf

http://library.binus.ac.id/eColls/eThesis/Bab2/2012-1-00520-mtif%202.pdf

http://www.en.wikipedia.com/wiki/euclidean.html

http://www.en.wikipedia.com/wiki/extended_euclidean.htm

http://www.en.wikipedia.com/wiki/Rijndael_S-Box

Kurniawan, Yusuf. 2004. *Kriptografi Keamanan Internet dan Jaringan Telekomunikasi*, Informatika, Bandung.

Munir, R. 2005. Penggunaan Tanda Tangan Digital untuk Menjaga Integritas Berkas Perangkat Lunak, Seminar Nasional Aplikasi Teknologi Informasi, Departemen Teknik Informatika Institut Teknologi Bandung.

Rahardjo, Budi. 2005. Insan Infonesia – Bandung & PT. INDOCISC – Jakarta.

Sapty Rahayu, Flourensia. 2005. *Cryptography*, Magister Teknologi Informasi FIKOM UI, Jakarta.