Aplikasi Enkripsi dan Dekripsi pada Soal Ujian Menggunakan Algoritma RSA Berbasis JAVA Desktop

Hariyanto[1], Fabian Rahmat Nugraha[2], Saepul Lukman[2] dan Diyah Ruri Irawati[2]

[1]Program Studi Manajemen Informatika, STMIK Jakarta STI&K
[2]Program Studi Sistem Informasi, STMIK Jakarta STI&K
Jl. BRI No.17 Radio Dalam, Kebayoran Baru, Jakarta Selatan
E-mail : ha07ri@gmail.com, fabian_rmht@gmail.com, refoel@yahoo.com, ruriz1@yahoo.com

Abstrak

Dokumen soal ujian harus terjaga kerahasiaannya dari pihak yang tidak berhak untuk mengaksesnya. Salah satu cara yang digunakan untuk menjamin keamanan data atau informasi adalah dengan menggunakan teknologi kriptografi. Kriptografi adalah teknik matematika yang diterapkan untuk mengubah data asli (plaintext) menjadi data yang diacak (ciphertext). Metode kriptografi yang digunakan pada penelitian ini menggunakan metode algoritma RSA ((Rivest Shamir Adleman). Algoritma RSA adalah metode yang digunakan untuk proses enkripsi dan dekripsi. Algoritma RSA menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya. Aplikasi kriptografi ini diterapkan untuk mengamankan soal ujian dalam bentuk file berjenis Doc dan PDF, pembuatan aplikasi menggunakan pemrograman Java.

Kata Kunci : Kriptografi, Enkripsi, Dekripsi, RSA, Java

Pendahuluan

Penerapan dari teknologi komputer sebagai salah satu media penyimpanan dan komunikasi menjadi suatu kebutuhan yang tidak dapat dipisahkan dari disetiap kegiatan yang beruhubungan dengan sistem informasi. Berbagai data atau informasi yang diperoleh akan dimanfaatkan untuk berbagai kepentingan. Pertukaran data atau informasi sudah semakin mudah dilakukan dengan atau tanpa melalui media fisik. Namun terkadang keamanan pertukaran data atau informasi tersebut kurang diperhatikan, salah satu dampak negatif dalam perkembangan teknologi adalah adanya pencurian data. Dengan adanya pencurian data maka aspek keamanan dalam pertukaran data atau informasi serta penyimpanan data dianggap penting [1].

Dokumen soal ujian harus terjaga kerahasiaannya dari pihak yang tidak berhak untuk mengaksesnya. Salah satu cara yang digunakan untuk menjamin keamanan data atau informasi adalah dengan menggunakan kriptografi. Penerapan kriptografi akan difokuskan kepada pengamanan data soal ujian yang tersimpan menjadi aman sampai dengan dokumen dapat diakses. Penelitian ini membuat suatu aplikasi keamanan data atau dokumen dengan menggunakan metode algoritma kriptograi RSA (Rivest Shamir Adleman) untuk meningkatkan keamanan dalam pengamanan data soal ujian. Pemilihan algoritma ini karena memiliki mekanisme kerja yang cukup sederhana dan mudah dimengerti namun tetap kokoh dalam tugasnya menyamarkan data [2]. Hal ini yang menjadi latar belakang penelitian untuk membuat apilkasi enkripsi dan dekripsi soal ujian menggunakan algoritma RSA berbasis java desktop.

Enkripsi Algoritma RSA

Algoritma RSA diperkenalkan oleh tiga peneliti dari MIT (Masschussets Institute of Technology), yaitu Ron Rivest, Adi Shamir, dan Len Adleman, pada tahun 1976. RSA merupakan teknik kriptografi dengan memanfaatkan 2 bilangan prima. Dari kedua bilangan prima tersebut dapat diperoleh sebuah Public Key (digunakan untuk mengenkripsi sebuah plainteks) dan sebuah Private Key (digunakan untuk mengdekripsi cipherteks)[3]. Algoritma RSA merupakan teknik kriptografi yang pal-

ing efektif sampai saat ini, hal ini disebabkan karena butuh waktu yang sangat lama untuk mendapatkan Private Key.

Algoritma Pembangkitan Pasangan Kunci

Untuk pembangkitan pasangan kunci RSA, digunakan algoritma sebagai berikut [4]:

1. Pilih dua buah bilangan prima sembarang yang besar p dan q. Nilai p dan q harus dirahasiakan
2. Hitung n = p x q. Besaran n tidak perlu dirahasiakan.
3. Hitung m = (p-1)(q-1)
4. Pilih e (kunci publik) yang relative prima terhadap m.
5. Relatif prima terhadap m artinya faktor pembagi terbesar keduanya adalah 1, secara matematis disebut gcd (e,m) = 1. Untuk mencarinya dapat digunakan algoritma Euclid.
6. Hitung d (kunci pribadi), untuk mencari nilai d secara matematis (d x e) mod n = 1. Dapat juga digunakan algoritma Extended Euclid.

Maka hasil dari algoritma tersebut diperoleh:

1. Kunci publik adalah pasangan (e,n).
2. Kunci pribadi adalah pasangan (d,n).

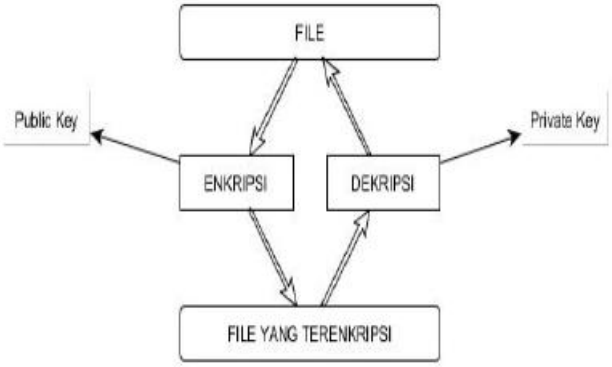
Enkripsi Pesan

1. Menggunakan kunci publik (e,n)
2. Plaintext M dinyatakan menjadi blok-blok m1, m2, m3, mi
3. Setiap blok mi di enkripsikan menjadi blok ci, dengan rumus ci = mie mod n.

Dekripsi Pesan

1. Menggunakan kunci pribadi (d,n).
2. Pilih ciphertext C
3. Setiap blok ci didekripsikan menjadi blok mi, dengan rumus mi = cid mod n.

Metode kriptografi secara umum dapat dilihat pada gambar 1 [5].



Gambar 1: Cara Kerja Aplikasi.

Contoh penggunaan algoritma RSA

Misalkan p=47 dan q= 71 (keduanya prima), kemudian menghitung nilai n = p . q = 47 . 71 = 3337, m = (p - 1)(q - 1) = (47 – 1)(71 -1) = 3220 pilih kunci e = 79, karena 79 relatif prima dengan 3220, e dan n dapat di publikasikan ke umum. Selanjutnya akan dihitung kunci dekripsi d menggunakan: d=e-1 mod m atau e.d (mod m)=1 sehingga dapat diperoleh (e=79 dan m=3220) : 79. d mod 3220 = 1 dengan mencoba nilai-nilai d = 1,2,3, n, diperoleh nilai kunci pribadi yang bulat dengan 1019. Ini adalah kunci dekripsi yang harus di rahasiakan.

Pesan yang akan dikirim adalah M = TEGUH atau dalam decimal (kode ASCII) adalah: 8469718572, nilai tersebut dipecah menjadi blok-blok m. Maka blok yang akan terbentuk adalah: m1 = 84; m2 = 69; m3 = 71; m4 = 85; m5 = 72;

Sebelumnya telah diketahui kunci publik adalah e=79 dan n=3337. Maka pesan M dapat dienkripsikan, yakni: c1 = 8479 mod 3337 = 1995; C2 = 6979 mod 3337 = 1689; c3 = 7179 mod 3337 = 1988; c4 = 8579 mod 3337 = 3048; C5 = 7279 mod 3337 = 285; sehingga ciphertext yang dihasilkan adalah : 1995 1689 1988 3048 285

Selanjutnya pesan yang terenkripsi tersebut dikirim kepada penerima pesan, yang mana telah memiliki kunci pribadi (d,m) = (1019, 3337) sehingga: m1 = 19951019 mod 3337 = 84; m2 = 16891019 mod 3337 = 69; m3 = 19881019 mod 3337 = 71; m4 = 30481019 mod 3337 = 85; m5 = 2851019 mod 3337 = 72; Maka akan dihasilkan kembali M = 8469718572, yang dalam pengkodean ASCII dapat dibaca sebagai berikut: M = TEGUH.

Perancangan Aplikasi

Aplikasi yang dibuat terdiri dari Form Menu Utama, Generate key, Encryption, Decryption, Help, dan Profile.

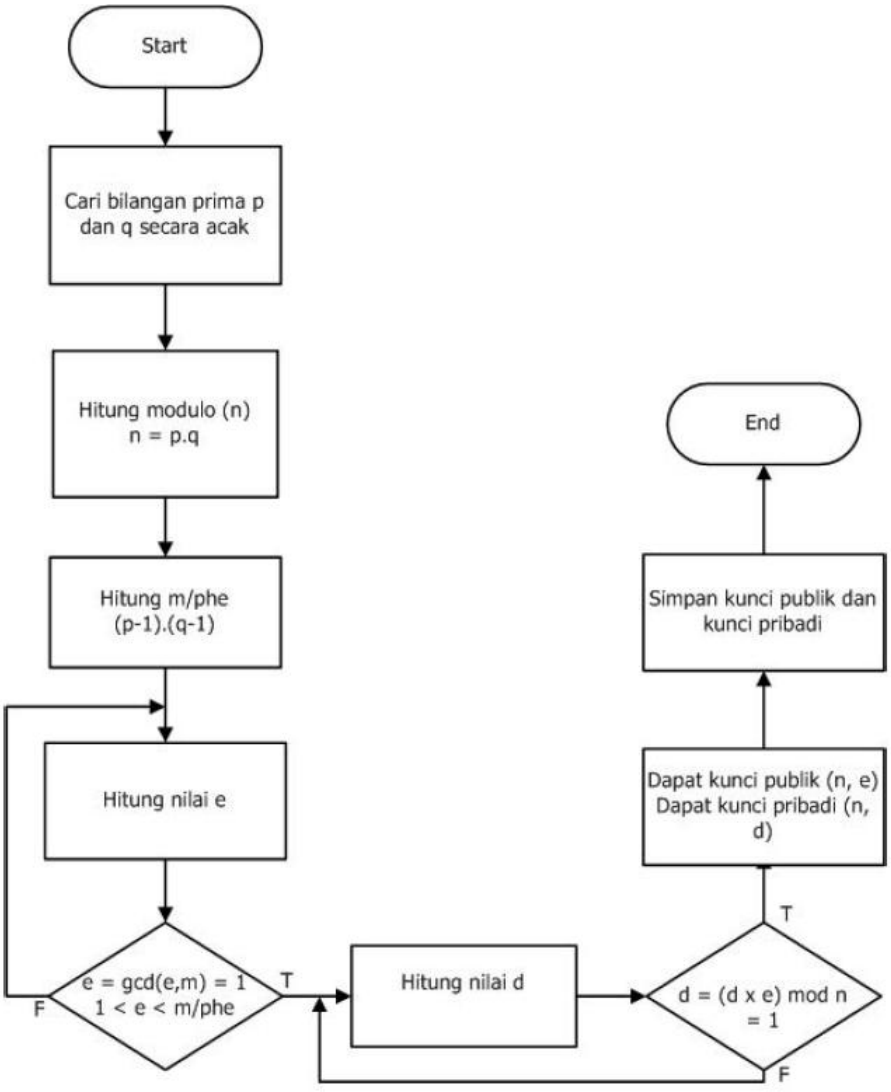
Untuk mendapatkan Public dan Private Key dapat menggunakan Menu Generate Key.

Untuk dapat melakukan enkripsi file dapat menggunakan Menu Encryption dengan memilih file yang akan dienkripsi dan menggunakan Public Key yang telah didapat untuk mengenkripsi file. Namun file dokumen soal ujian tidak boleh lebih besar dari ukuran file yang telah ditentukan..

Sedangkan untuk mengembalikan file yang sudah dienkripsi menjadi file asli, dapat memilih menu dekripsi, dengan memilih file yang terenkripsi dan juga menggunakan Private Key yang sepasang dengan Public Key yang digunakan untuk mengenkripsi file tersebut. Serta ada menu help untuk membantu user dalam menggunakan program tersebut.

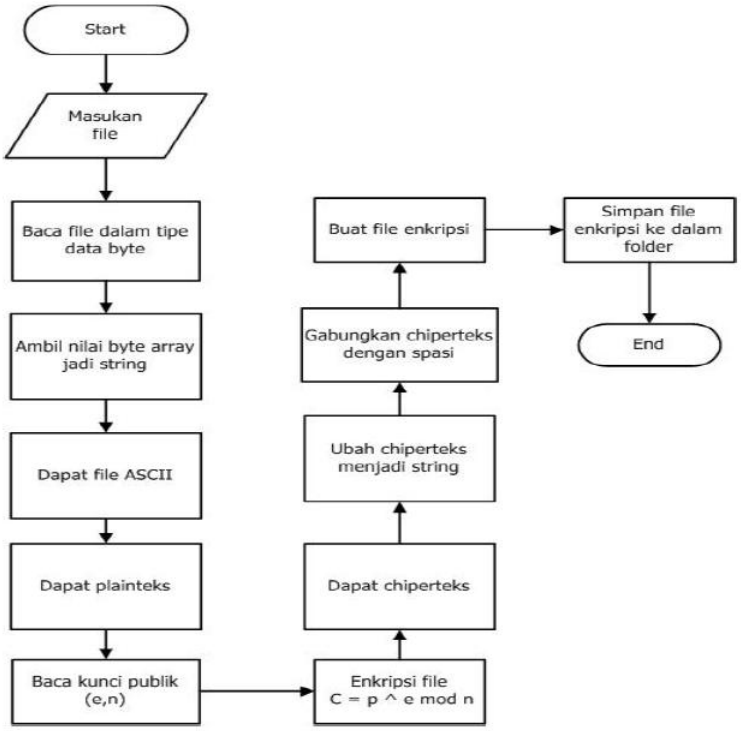
Flowchart Proses Pembuatan Kunci

Flowchart proses pembuatan kunci merupakan gambaran pembuatan kunci n, e, dan d yang akan menghasilkan private key dan public key. Flowchart proses pembuatan kunci dapat dilihat pada gambar 2.



Gambar 2: Flowchart Pembuatan Kunci Flowchart Proses Enkripsi RSA

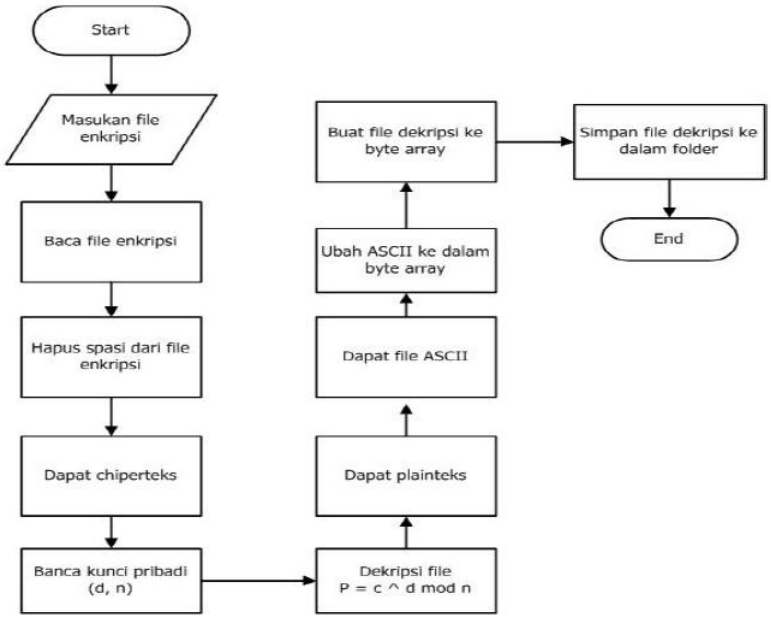
Flowchart proses enkripsi RSA merupakan gambaran alur sebuah file yang akan mengalami proses enkripsi (pengacakan isi file). Flowchart proses enkripsi RSA dapat dilihat pada gambar 3



Gambar 3: Flowchart Proses Enkripsi RSA.

Flowchart Proses Dekripsi RSA

Flowchart proses dekripsi RSA merupakan gambaran alur sebuah file yang akan mengalami proses dekripsi. Flowchart proses dekripsi RSA dapat dilihat pada gambar 4.



Gambar 4: Flowchart Proses Dekripsi RSA

Implementasi Aplikasi

Bagian ini menjelaskan pengimplementasian proses sebuah aplikasi kriptografi dari pertama kali dijalankan sampai selesai.

Tampilan Form Menu Utama

Tampilan layar dari form Menu Utama pada gambar 5 ini muncul pada pertama kali aplikasi dijalankan dan terlihat ada bermacam-macam menu yang bisa digunakan seperti Generate key, Encryption, Decryption, Profile, Help, dan Exit untuk keluar dari program.

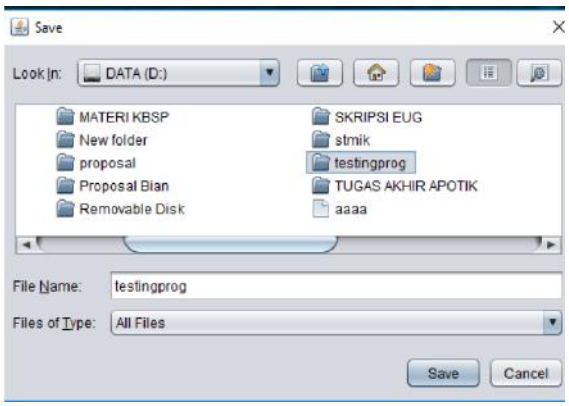
Gambar 5: Tampilan Form Menu Utama..

Tampilan Form Generate Key

Tampilan layar pada Form Generate Key pada gambar 6 ini muncul pada saat memilih menu Generate key pada menu utama, form generate key digunakan untuk untuk membuat Public dan Private Key.

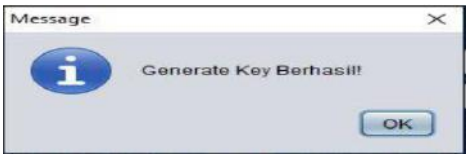
Gambar 6: Tampilan Form Generate key..

Tampilan gambar 7 menggambarkan memilih folder yang ingin digunakan untuk menyimpan hasil pembuatan kunci.



Gambar 7: Tampilan Penyimpanan Folder Generate Key..

Tampilan Gambar 8 menggambarkan bahwa proses membuat kunci Public dan Private sudah berhasil di buat.



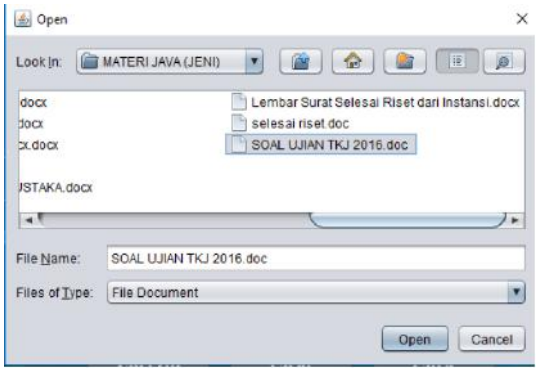
Gambar 8: Tampilan Layar Pesan Generate Key berhasil.

Tampilan Layar Form Encryption File

Tampilan layar form Encryption pada gambar 9 ini muncul pada saat menu Encryption dijalankan.

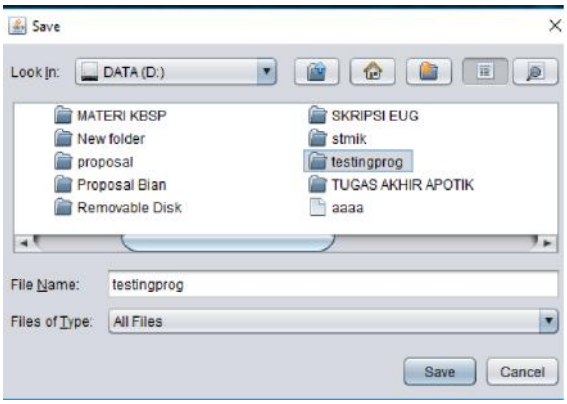
Gambar 9: Tampilan Layar Form ENCRYPTION .

Tampilan gambar 10 menggambarkan memilih file yang akan dienkripsi, adapun jenis file yang bisa digunakan yaitu *txt, *doc, *docx, *xls dan *xlsx. Tetapi batas maksimum file yang bisa dienkripsi adalah 2MB.



Gambar 10: Tampilan layar memilih file asli.

Tampilan gambar 11 menggambarkan memilih folder untuk menyimpan hasil file yang telah dienkripsi.



Gambar 11: Tampilan layar menyimpan file.

Tampilan gambar 12 menggambarkan memilih public key yang di gunakan pada proses enkripsi.

Gambar 12: Tampilan memilih public key

Tampilan gambar 13 menggambarkan pesan ketika enkripsi berhasil di lakukan.

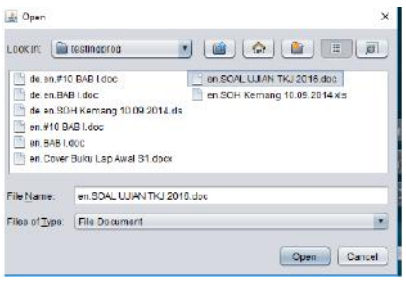
Gambar 13: Tampilan layar pesan proses encrypt berhasil.

Tampilan Layar Form Decryption File

Tampilan layar dari Form Decryption file pada gambar 14 ini muncul pada saat menu Decryption dijalankan.

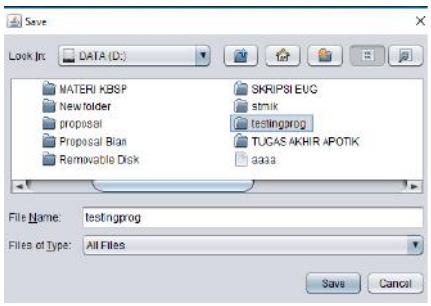
Gambar 14: Tampilan layar form Decryption

Tampilan gambar 15 menggambarkan ketika user memilih file yang akan didekripsi.



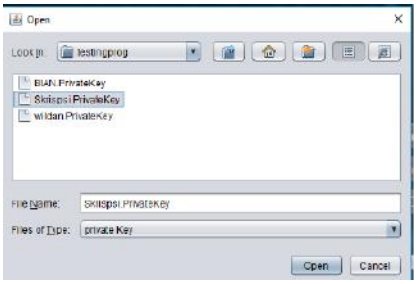
Gambar 15: Tampilan layar memilih file enkripsi

Tampilan gambar 16 menggambarkan memilih lokasi untuk menyimpan hasil file yang telah berhasil didekripsi.



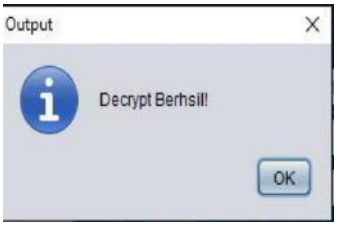
Gambar 16: Tampilan layar memilih folder penyimpanan Decryption

Tampilan gambar 17 menggambarkan memilih private key atau public key yang akan digunakan untuk melakukan proses dekripsi.



Gambar 17: Tampilan layar memilih private key.

Tampilan gambar 18 menggambarkan pesan ketika dekripsi berhasil dilakukan.



Gambar 18: Tampilan layar pesan proses decrypt berhasil. .

Tampilan layar form Help

Tampilan layar form help berguna untuk membantu pengguna dalam melakukan proses proses yang ada pada aplikasi ini. Tampilan layar bisa dilihat pada gambar 19.

Gambar 19: Tampilan layar form help

Tampilan Layar Form Help Generate Key

Tampilan form help generate key berfungsi untuk menjalankan proses pembuatan public key atau private key pada aplikasi ini , gambar bisa di lihat pada tampilan layar gambar 20.

Gambar 20: Tampilan layar form help generate key.

Tampilan Layar Form Help Encryption

Diaplikasi ini ada form help encryption , berfungsi untuk memberitahu cara kerja dari proses enkripsi pada aplikasi ini . gambar bisa di lihat pada tampilan layar gambar 21.

Gambar 21: Tampilan layar form help encryption.

Tampilan Layar Form Help Decryption

Tampilan layar form help decryption , berfungsi untuk memberi tahu pengguna cara menjalankan proses dekripsi pada aplikasi ini. Gambar bisa di lihat pada tampilan layar 22.

Gambar 22: Tampilan layar form help decryption..

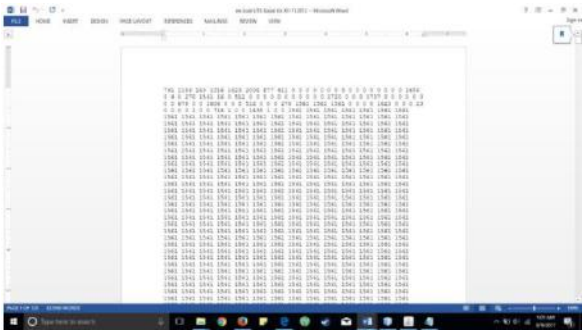
Uji Coba Aplikasi

Pada peneltian ini uji coba dilakukan untuk meng- enkripsi sebuah file soal ujian, dengan memilih Menu Enkripsi pada aplikasi. Kemudian memilih file soal ujian yang akan diuji coba. Pada gambar 23 dan gambar 24, memperlihatkan tampilan awal file soal ujian yang akan di enkripsi.

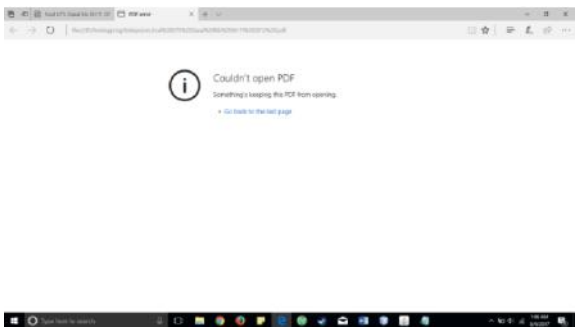
Gambar 23: Tampilan isi file soal ujian .doc

Gambar 24: Tampilan isi file soal ujian .pdf

Setelah file soal ujian berhasil dienkripsi, maka file Doc atau Pdf yang sudah dienkripsi tidak akan bisa dibaca atau dibuka lagi seperti terlihat pada gambar 25 dan ganbar 26.



Gambar 25: Tampilan hasil enkripsi soal ujian .doc



Gambar 26: Tampilan hasil enkripsi soal ujian .pdf.

Untuk melakukan proses dekripsi file, dapat memilih menu dekripsi. Lalu memilih file soal ujian yang sudah dienkripsi dan melakukan proses dekripsi hingga menjadi file soal ujian yang asli. Tahapan untuk melakukan dekripsi file sama seperti mengenkripsi file. Pada gambar 27 dan gambar 28 memperlihatkan tampilan file soal ujian yang telah berhasil didekripsi.

Gambar 27: Tampilan hasil dekripsi soal ujian .doc.

Gambar 28: Tampilan hasil dekripsi soal ujian .doc.

Dari hasil iji coba aplikasi ini terlihat bahwa file yang akan dilakukan enkripsi baik yang berjenis dokumen doc atau dokumen pdf berhasil melalui tahap-tahap proses enkripsi dan deskripsi pada aplikasi tersebut.

Penutup

Berdasarkan perancangan, pembuatan, serangkaian uji coba dan analisa program dari aplikasi enkripsi ini, maka dapat diambil suatu kesimpulan :

1. Aplikasi telah berjalan sesuai dengan rancangannya dan ujicoba file data berjenis Doc dan PDF.
2. Dengan adanya aplikasi enkripsi, proses penyimpanan soal ujian dan pertukaran informasi menjadi lebih aman.
3. Satu kunci pasangan public key dan private key bisa digunakan berkali-kali untuk mengenkripsi soal ujian dengan jenis file yang sama ataupun berbeda.
4. Proses dekripsi dengan kunci yang sesuai akan mengembalikan file menjadi file semula tanpa mengalami perubahan sedikitpun.
5. Waktu yang digunakan untuk melakukan proses enkripsi dan dekripsi berbanding lurus dengan ukuran file yang diproses (semakin kecil ukuran file yang diproses, semakin cepat proses enkripsi dan dekripsi dilakukan, semakin besar ukuran file yang diproses, semakin lama proses enkripsi dan dekripsi dilakukan).

Daftar Pustaka

[1] Sumarkidjo dan Pinuji Prasetyaningtyas, “Jelajah Kriptologi”, Lembaga Sandi Negara, Jakarta, 2007

[2] Candra Putra Devha, ”Pengamanan Pesan Rahasia Menggunakan Algoritma Kriptografi Rivest Shamir Adleman”, Universitas Pendididikan Indonesia, 2007

[3] William Stallings, “Cryptography and Network Security”, 7th ed., Pearson Education, Inc., New Jersey, 2017

[4] Wico Chandra, ” Kriptografi dan Algoritma RSA”, Bandung:Institut Teknologi Bandung, 2010.

[5] Prasetya Andy Wicaksono, ”Enkripsi Menggunakan Algoritma RSA”, Bandung: Institut Teknologi Bandung, 2012.

[6] Meidina, “Visualisasi Algoritma RSA dengan Menggunakan Bahasa Pemrograman Java”, Unpublished PI Universitas Gunadarma, 2008.

[7] Suarga, “Dasar Pemrograman Komputer dalam Bahasa Java”, Cv. Andi Offset, Yogyakarta, 2009.