MJI

Media Jurnal Informatika
Vol. 9, no.2,Desember 2017, hlm 78-83
http://jurnal.unsur.ac.id/mjinformatika

p-issn : 2088-2114
e-issn : 2477-2542

KEAMANAN FILE DENGAN TEKNIK ZIGZAG DAN HUFFMAN

Agus Suheri
Program Studi Teknik Informatika
Universitas Suryakancana
agussuheri@unsur.ac.id

Abstrak

Keamanan file merupakan upaya menjaga asset yang dimiliki oleh organisasi agar dapat tetap beraktivitas secara tenang. Berbagai teknik keamanan data banyak diimplementasikan dalam melakukan pengamanan terhadap data. Metode-metode klasik masih relevan untuk dapat digunakan dalam pengamanan file dimasa saat ini. Metode klasik kriptografi seperti Metode Zigzag adalah salah satu metode klasik yang masih mungkin untuk diterapkan dalam keamanan data, walaupun memiliki kelemahan karena kesederhanaannya. Untuk menutupi kesedehanaan dari metode klasik zigzag ini maka dalam pengamanan dikolaborasikan dengan pengkodeaan pemadatan data dengan metode Huffman sehingga data menjadi sulit dibaca dari pihak yang tidak berkepentingan. Dalam implementasinya digunakan tools pengembangan perangkat lunak Delphi Xe5 dalam membuat pemrograman keamanan file dengan teknik Zigzag dan Huffman.

Kata kunci: : Keamanan Data, Zigzag, Huffman, Delphi Xe5

1. Latar Belakang Masalah

Teknologi dan konsep pengelolaan data berkembang dengan pesat. File wadah untuk menyimpan sekumpulan data, yang tersimpan dalam bentuk digital. Dalam penyimpanan data dikelola dengan sedemikan rupa agar data mudah diakses dan dikelola. Dalam pengelolaan diantaranya dilakukan keamanan data. Perkembangan teori keamanan data dibagi 2 (dua) yaitu teori keamanan data klasik dan modern. Metode Zigzag merupakan teori klasik yang digunakan untuk melakukan penulisan rahasia dengan teknik transposisi dimana dilakukan permutasi atau perpindahan karakter yang biasanya dilakukan secara berulang-ulang, baik dengan kunci yang sama maupun kunci yang berbeda untuk setiap kali permutasi, dan cara untuk memasukkan huruf plaintext dalam perpindahannya dengan pola "zigzag" ().Metode Huffman adalah suatu teknik kompresi data secara statistik yang bekerja dengan mereduksi panjang kode rata-rata dan menghasilkan kode prefiks yang digunakan untuk merepresentasikan simbol-simbol dari suatu jenis huruf. Huruf itu dapat berupa huruf dalam Bahasa Inggris atau jenis huruf terkode seperti misalnya ASCII atau himpunan karakter EBCDIC.

Kumpulan data yang tersimpan dalam bentuk file merupakan sebuah aset yang sangat penting bagi organisasi. Semakin pentingnya aset tersebut maka organisasi tersebut akan melakukan pengamanan terhadap aset tersebut agar tidak dapat digunakan oleh pihak yang tidak berkepentingan. File menjadi salah satu target yang diincar oleh pihak yang tidak berhak menggunakannya untuk keperluan mereka. Terminologi keamanan jaringan atau keamanan data selalu mengingatkan segala sesuatu yang berkaitan dengan perlindungan data dan pembatasan terhadap data tersebut. Hal yang menjadi utama dan terpenting dalam menyelamatkan aset-aset yang dianggap vital bagi kelangsungan hidup sebuah organisasi dan perusahaan (Schaum's:2005:p218).

Pelindungan terhadap data maka diperlukan penerapan/aplikasi dari teori keamanan data, baik dengan keamanan data klasik ataupun modern File sebagai asset oganisasi yang perlu diamankan dapat diamanankan dengan menerapkan teori klasik keamanan data Metode Zigzag dalam penulisan rahasianya dan dikolaborasikan dengan metode kompresi/pemadatan data yang dikodekan, yaitu metode Huffman. Dengan menggunakan kolaborasi metode penulisan rahasia dan metode pemadatan data yang diimplementasikan dalam pemrograman Delphi Xe5, maka data akan tertutup dengan terkode dalam 2 (dua) metode yang berbeda, yaitu penulisan rahasia dan kompresi, sehingga data menjadi aman dari akses pihak yang tidak mendapatkan hak akses.

2. Rumusan Masalah

Berdasarkan hal itu maka rumusan masalah penlitian ini adalah: Bagaimana membuat Keamanan file dengan metode Zigzag dan Huffman sehingga data dalam file tidak dapat dibaca oleh pihak yang tidak memiliki hak akses.

3. Maksud dan Tujuan

Maksud dari penelitian ini adalah membuat aplikasi Keamanan file dengan metode Zigzag dan Huffman sehingga data dalam file tidak dapat dibaca oleh pihak yang tidak memiliki hak akses.

Sedangkan tujuan dari penelitian ini, adalah:

a. Menghasilkan aplikasi yang akan mengamanankan file dari akses pihak yang tidak memilik hak akses
b. Menerapkan teori klasik kedalam terapan pemrograman untuk keamanan data.

4. Landasan Teori

4.1 File

File adalah arsip aktif yang masih terdapat di unit kerja dan masih diperlukan dalam proses administrasi secara aktif, masih secara langsung digunakan. File atau arsip merupakan wadah yang digunakan untuk menyimpan atau menampung atau mengumpulkan data. Penyimpanan data dalam file terbagi dalam 2 jenis yaitu penyimpanan terstruktur dan tidak tersturktur. Penyimpanan terstruktur artinya penyimpanan data dikelola dengan memiliki bentuk atau keteraturan atau format tertentu, contoh file table, gambar, worksheet, dan lain-sebagainya. Ciri dari file terstruktur adalah memiliki header yang menjadi data tentang struktur file tersebut dilanjutkan dengan data yang disimpannya. Sedangkan file tidak terstruktur adalah penyimpanan data dilakukan tanpa memiliki format data, data tersimpan sesuai dimana posisi disimpan.

4.2 Keamanan Data

Menurut Ian Sommervile mengatakan keamanan sistem merupakan penilaiaan sampai sejauh mana sistem dapat melindungi dirinya sendiri dari serangan eksternal yang mungkin sengaja ataupun tidak sengaja, seperti Virus. Penggunaan yang tidak sah atas pelayanan system, modifikasi yang tidak berijin. Keamanan penting untuk semua sistem, tanpa ada tingkatan pengamanan yang memadai, ketersediaannya, keandalan, dan keselamatan, sistem akan mudah rusak (Ian Sommervile, Software Engineering, edisi 6 200: p16).

Keamanan informasi didapat dengan cara mengimplementasi seperangkat alat kontrol yang layak yaitu berupa kebijakan-kebijakan, praktek-praktek, prosedur-prosedur, struktur-struktur organisasi dan piranti lunak. Keamanan informasi mengunci informasi dari ancaman yang luas untuk memastikan kelanjutan usaha, memperkecil rugi perusahaan dan memaksimalkan laba atas investasi dan kesempatan usaha.

Keamanan informasi terdiri dari perlindungan terhadap aspek-aspek berikut:

a. Confidentiality (kerahasiaan) aspek yang menjamin kerahasiaan data atau informasi, memastikan bahwa informasi hanya dapat diakses oleh orang yang berwenang dan menjamin kerahasiaan data yang dikirim, diterima dan disimpan.
b. Integrity (integritas) aspek yang menjamin bahwa data tidak dirubah tanpa ada ijin pihak yang berwenang (authorized), menjaga keakuratan dan keutuhan informasi serta metode prosesnya untuk menjamin aspek integrity ini.
c. Availability (ketersediaan) aspek yang menjamin bahwa data akan tersedia saat dibutuhkan, memastikan pengguna yang berhak dapat menggunakan informasi dan perangkat terkait (aset yang berhubungan bilamana diperlukan).

Perkembangan Teknologi Informasi yang begitu pesat tidak selamanya memberikan dampak positif. Berbagai dampak negatif bermunculan akibat perkembangan Teknologi Informasi sehingga meresahkan berbagai pihak. Untuk mengurangi keresahan tersebut maka diperlukan suatu sistem pengaman yang dapat membuat berbagai pihak merasakan ketenangan, kenyamanan dan keaman dalam menikmati Teknologi Informasi.

4.3 Metode Zigzag

Metode Zigzag adalah metode kriptografi klasik dimana plaintext yang akan dienskrip diketahui dahulu panjangnya, selanjutnya ditentukan lebar zigzagnya yang telah ditentukan dan tuliskan setiap huruf membentuk zigzag.

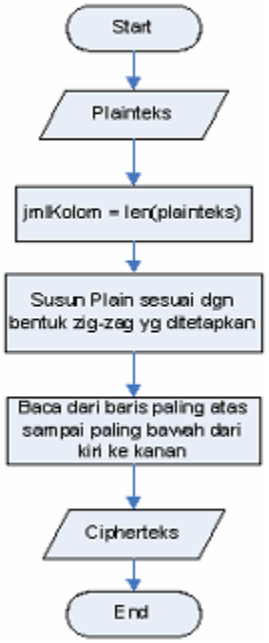
Contoh:

Keamanan File Ok

K						a					e			
	e				n		n				l			
		a		A						i				O	
			m						f						k

Chipertext:

Kaeennl aa iOmfk



Gambar 1 Flowchart Enskripsi Metode Zigzag

Gambar 2 Flowchart Deskripsi Zigzag

4.4 Metode Huffman

Prosedur dari metode kompresi ini dibangun pertama kali oleh D.A. Huffman pada tahun 1950. Metode ini merupakan pengembangan dari metode sebelumnya yang dikemukakan oleh Shannon-Fano. Dari hasil perhitungan dan analisa yang telah dilakukan oleh Gilbert Held (1991) metode ini relatif menghasilkan unjuk kerja yang sedikit lebih efektif dibanding metode Shannon Fano.

Metode Huffman adalah suatu teknik kompresi data secara statistik yang bekerja dengan mereduksi panjang kode rata-rata dan menghasilkan kode prefiks yang digunakan untuk merepresentasikan simbol-simbol dari suatu jenis huruf. Huruf itu dapat berupa huruf dalam Bahasa Inggris atau jenis huruf terkode seperti misalnya ASCII atau himpunan karakter EBCDIC.

Kode Huffman digunakan secara luas dan sangat efektif untuk kompresi data. Algoritma huffman menggunakan tabel yang menyimpan frekuensi kemunculan dari masing-masing simbol yang digunakan dalam file tersebut dan kemudian mengkodekannya dalam bentuk biner.

Pertama-tama buat tabel frekuensi dari semua simbol atau karakter yang muncul dalam suatu file teks. Kemudian diurutkan mulai dari simbol dengan frekuensi paling sedikit sampai simbol dengan frekuensi paling banyak. Pembentukan pohon huffman dimulai dari dua simbol paling depan untuk dijadikan anak kiri dan anak kanan dari pohon yang terbentuk dan frekuensinya dijumlahkan. Setelah itu diurutkan kembali berdasarkan frekuensi yang baru. Demikian dilakukan terus menerus sampai semua simbol terbentuk menjadi pohon hufman.

Dengan algoritma di atas maka terbentuklah suatu pohon yang menyimpan simbol - simbol sedemikian rupa sehingga simbol dengan frekuensi kemunculan paling banyak akan ditempatkan di kedalaman yang paling dangkal, sedangkan simbol-simbol dengan frekuensi kemunculan paling sedikit akan ditempatkan di kedalaman yang paling dalam. Setelah proses pembuatan pohon selesai, maka pohon siap digunakan untuk membentuk kode. Dari cara kerja metode ini, maka dibutuhkan pohon untuk membaca kembali (encode). Oleh karena itu maka pohon hasil proses huffman harus disimpan dalam file kompresi. Jika file yang dikompresi ukurannya kecil, maka tidak banyak bit yang dihemat. Semakin besar ukuran file maka semakin banyak bit yang berhasil dihemat. Tetapi karena harus menyimpan pohon huffman yang dihasilkan maka jika ukuran file yang dikompresi kecil maka file hasil kompresi bisa jadi bukannya semakin kecil tetapi malah semakin besar. Selain itu jika variasi huruf dalam file yang akan dikompresi sangat besar, maka ada kemungkinan pohon yang terbentuk sangat dalam dan kode yang dihasilkan ukurannya sama dengan ukuran satu karakter. Jika hal ini terjadi maka file hasil kompresi bukannya lebih kecil malah menjadi lebih besar. Ide awal pembentukan pohon huffman adalah untuk membuat prefik code, yaitu kode berupa string biner dimana kode yang satu bukan merupakan awalan dari kode yang lain. Secara garis besar proses kompresi dimulai dengan membuat pohon huffman, setelah itu mengubah setiap karakter menjadi kode biner dan terakhir menyimpan kembali menjadi karakter yang lain karena ada pengerutan jumlah bit. Untuk dekompresinya maka pertama-tama harus membentuk kembali pohon yang disimpan diawal file hasil kompresi. Setelah pohon terbentuk kembali maka semua karakter diubah dalam bentuk biner kemudian diterjemahkan satu persatu dengan menggunakan pohon, sampai menemukan karakter di setiap leaf(daun).

Hal yang perlu diperhatikan adalah penyimpanan pohon huffman. Jika terjadi kesalahan, maka file kompresi tidak akan terbaca.

Algoritma kompresi file:

a. Baca file yang akan dikompresi (file asal).
b. Hitung frekuensi dari setiap karakter yang ada. Jenis karakter dan frekuensinya disimpan sebagai tree dalam sebuah list.
c. Selama list masih mempunyai node lebih dari satu maka
    1. Urutkan list berdasarkan frekuensinya, secara ascending.
    2. Buat pohon baru, kaki kiri diisi dengan node pertama list dan kaki kanannya diisi dengan node kedua list. Karakter yang disimpan berupa “æ“dan frekuensinya merupakan jumlahan dari frekuensi kedua kakinya.
d. Sebelum hasil kompresi disimpan dalam file hasil, maka simpan terlebih dahulu pohon huffmannya. Pertama simpan jumlah karakter keseluruhan. Kemudian telusuri pohon secara preorder, kaki kiri diberi kode “0“dan kaki kanan diberi kode “1“. Setelah sampai pada leaf maka simpan karakter pada leaf tersebut baru kemudian simpan deretan kode yang dihasilkan. Setiap kode disimpan dengan dibatasi “æ. Setelah pohon selesai disimpan, untuk membatasi penyimpanan pohon dengan hasil kompresi, disisipkan “Æ“.
e. Telusuri pohon secara preorder, kaki kiri diberi kode “0“dan kaki kanan diberi kode

"1". Setelah sampai pada leaf maka deretan kode yang didapat digunakan untuk mengkodekan karakter pada leaf tersebut. Setelah penelusuran selesai maka akan didapat daftar karakter dan kode binernya.

f. Baca setiap karakter pada file asal. Ubah menjadi kode biner. Setelah semua dikodekan, setiap 8 kode dicari desimalnya (ASCII) kemudian simpan kembali sebagai karakter baru.

Algoritma dekompresi file:

a. pertama baca dahulu pohon dan bentuk kembali pohonnya. Antara "æ"ada karakter dan deretan biner. Karakter nantinya akan disimpan di leaf. Deretan biner digunakan untuk menelusuri pohon, dengan aturan kode "0"ke kiri dan kode "1"ke kanan. Jika node yang ditelusuri belum ada, maka buat node baru. Karakter disimpan di node terakhir yang dibentuk dari setiap penelusuran kode biner.
b. setelah pohon terbentuk, baca setiap karakter kemudian ubah ASCII dari karakter tersebut menjadi biner. Setelah itu telusuri pohon dengan aturan, jika kode biner "0"maka telusuri kaki kiri. Jika kode biner "1"maka telusuri kaki kanan. Telusuri terus sampai leaf. Karakter pada leaf tersebut adalah hasil dekompresinya. Setelah ketemu leaf maka penelusuran dilakukan dari root lagi.

5. Analisa Keamanan File

Pengelolaan file biasanya dilakukan karena pembuatan dan penerimaan dokumen. File yang dikelola akan disimpan pada setiap media penyimpan di komputer kerja pengelola, dan atau dimedia penyimpan pada komputer pusat data, dan sebagian disimpan dalam media lain seperti CD dan FlashDisk. Pengaksesan file dapat dilakukan dengan teknik stand-alone dan atau jaringan. Pengelolaan file menggunakan sistem operasi berbasis Microsoft Windows sedangkan pengelolaan file menggunakan Microsoft Office dimana untuk mengamankan file digunakan fasilitas proteksi file yang disediakan oleh Microsoft Office dengan menambahkan password.

Berbagai akses yang dilakukan oleh pihak yang tidak berkepentingan sulit untuk diketahui oleh pengelola file, dikarenakan pengaksesan file tersebut tidak meninggalkan jejak atau kesan bahwa telah terjadi pengaksesan. File sebagai aset dari organisasi yang tersimpan pada setiap media penyimpan komputer rentan untuk diakses karena terbuka. Pengaksesan yang terjadi yang dilakukan dapat berupa membuka, men-copy, mengedit, dan menghapus file.

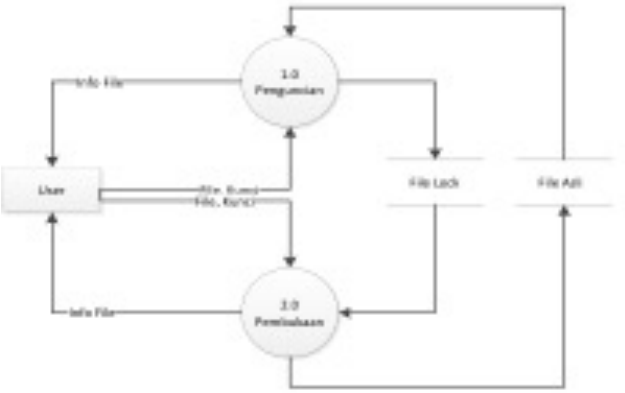
Prilaku pengguna yang sering meninggalkan atau meminjamkan komputer atau bahkan menggunakan komputer secara bersama kepada pihak lain menjadi kunci pembuka terhadap keamanan file. Akibat kebiasaan tersebut walaupun Sistem Operasi memiliki pengaturan hak pakai menjadi tidak berfungsi sebagai mana mestinya, pihak lain bisa mengakses karena komputer dalam keadaan aktif sedangkan tahapan login Sistem Operasi telah terbuka atau terlewati.

6. Perancangan Keamanan File

Berdasarkan hasil analisi tersebut diatas maka dirancang suatu pengamanan file dengan menggunakan metode kriptografi dan medote kompresi dengan menggunakan Metode Zigzag dan Metode Huffman. Rancangan proses dari keamanan file ini terdiri 2 (dua) proses utama, yaitu "Penguncian" dan "Pembukaan".

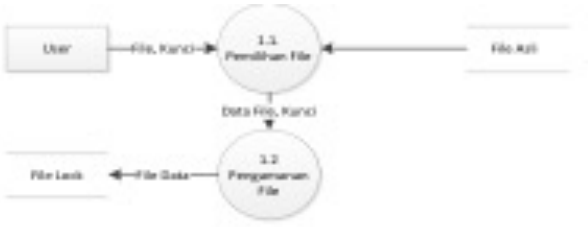
Gambar 3 Diagram Konteks Keamanan File

Diagram kontek Keamanan File ini menggambarkan antara hubungan entitas eksternal User dengan Keamanan file dimana user mengirimkan nama file dan password untuk pengamanan file kepada system dan mendapatkan kembali info dan data file.



Gambar 4 Diagram Alir Data Level 0

Diagram alir data level 0 ini menjelaskan tentang aliran data yang terjadi dalam system keamanan file dengan entitas eksternal user yang mengalir dalam proses penguncian dan pembukaan. Pada proses penguncian user mengirim file dan password ke proses 1.0 Penguncian yang akan membaca isi file yang dikirim, dimana apabila ada kode bahwa file tersebut sudah dikunci maka proses akan memberikan informasi kepada user bahwa "File Telah Terkunci", dan apabila status file tidak terkunci maka akan dilakukan proses penguncian yang hasilnya berupa "file lock". Sedangkan dalam aliran data pada proses 2.0 pembukaan, user akan mengirim file dan password kemudian proses 2.0 Pembukaan akan memberikan informasi status file jika file tidak memiliki kode kunci berupa "File tidak terkunci" dan apabila file memiliki kode terkunci maka proses akan meminta password dimana jika salah akan muncul informasi "Password Salah" dan jika password benar maka file akan dibuka dan akan membuat file baru menjadi "File Asli".

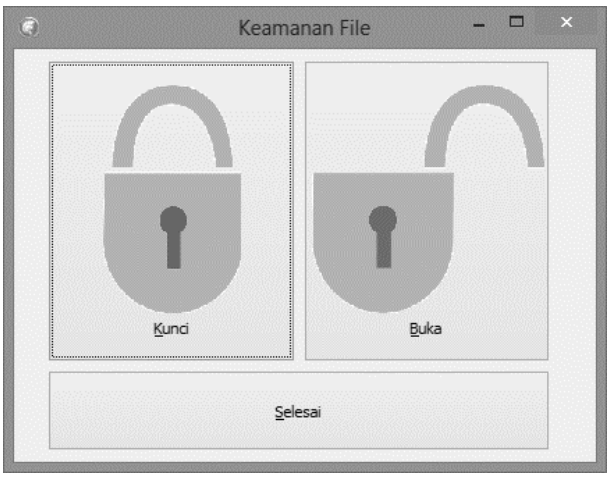


Gambar 5 Diagram Alir Data Level 1 Proses 1

Gambar 6 Diagram Alir Data Level 1 Proses 2

7. Implementasi

Dalam implementasi yang dilakukan perangkat lunak yang diperlukan selain Sistem Operasi Windows adalah Delphi Xe5 sebagai perangkat lunak yang digunakan dalam mengimplementasikan rancangan kedalam pemrograman.

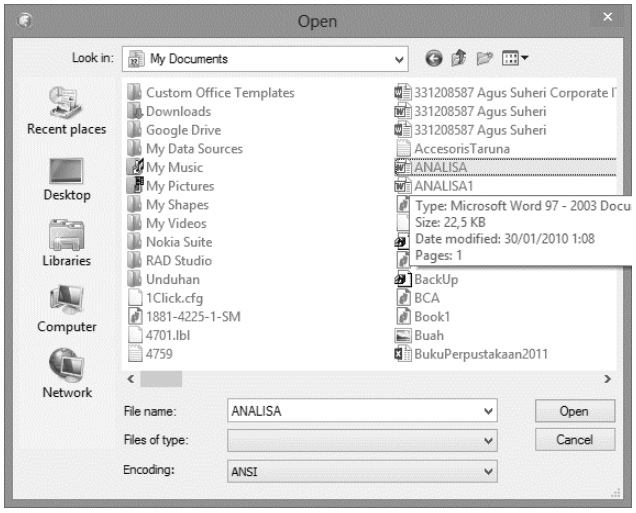


Gambar 7 Tampilan Utama Program

Program pertama kali akan menampilkan seperti pada Gambar 7, dimana user bisa memilih tombol Kunci, Buka, dan Selesai. Jika user memilih tombol Kunci maka akan muncul tampilan seperti gambar 8 dibawah ini.

Gambar 8 Form Penguncian File

Pada saat akan mengunci file, user diminta untuk memilih atau membuka file yang akan dikunci, seperti pada gambar 9.

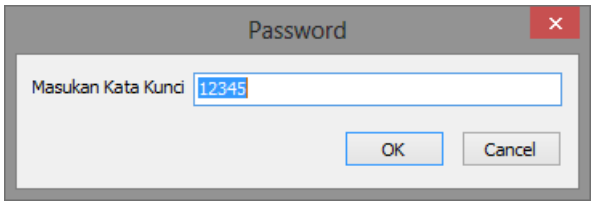


Gambar 9 Memilih File yang akan dikunci

Setelah memilih file yang akan dikunci, maka klik Open, yang selanjutnya sistem akan memeriksa apakah ada tanda "kunci" pada file yang dipilih, apabila ternyata dalam file tersebut ada tanda kunci maka akan muncul pesan seperti gambar 10.

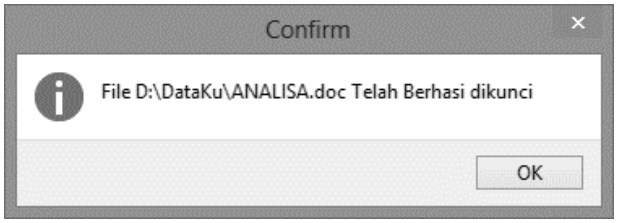
Gambar 10 Pesan Error Apabila File Telah Dikunci

Apabila sistem tidak menemukan tanda "kunci" maka akan muncul dialog untuk memasukan kata kunci untuk file yang akan dikunci seperti pada gambar 11.



Gambar 11 Masukan Kata Kunci

Jika telah selesai memasukan kata kunci maka sistem akan menampilkan pesan berhasil mengunci file, seperti pada tampilal 12.

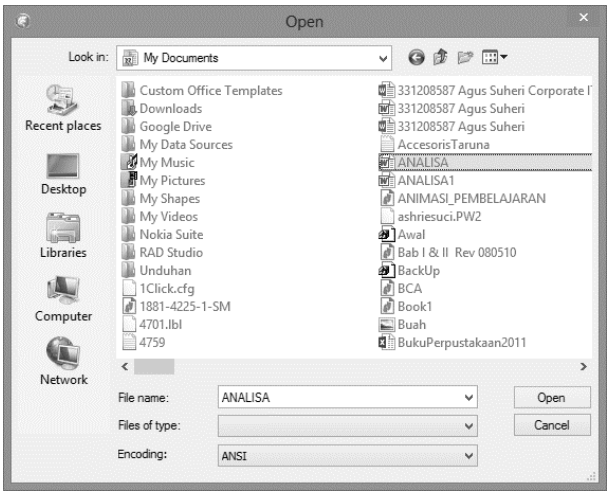


Gambar 12 Pesan Penguncian Berhasil

File yang telah terkunci bisa dibuka oleh user dengan memilih tombol buka, dimana akan memunculkan tampilan seperti gambar 13 berikut ini.

Gambar 13 Form Pembukaan File

User diminta untuk memilih file yang akan dibuka,



Gambar 14 Memilih File yang akan dibuka

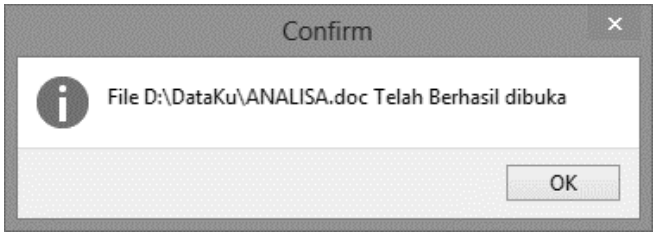
Apabila file yang akan dibuka ternyata belum dikunci maka akan memunculkan pesan kesalahan seperti berikut:

Gambar 15 Pesan Error Apabila File Belum Terkunci

Sedangkan apabila menemukan tanda "kunci" maka sistem akan meminta kata kunci yang menjadi password file yang dikunci.

Gambar 16 Masukan Kata Kunci

Jika passwordnya benar maka akan muncul pesan seperti dibawah ini:



Gambar 17 Pesan File Berhasil dibuka

8. Simpulan

Berdasarkan uraian sebelumnya maka diambil kesimpulan bahwa telah berhasil dibuat aplikasi Keamanan file dengan metode Zigzag dan Huffman sehingga data dalam file tidak dapat dibaca oleh pihak yang tidak memiliki hak akses, dan menerapkan teori klasik kedalam terapan pemrograman untuk keamanan data.

9. Daftar pustaka

[1] Suheri, Agus, (2010) Pengamanan File Dengan Kompresi Huffman dan Penyamaran Data Steganografi, Media Jurnal Informatika, Vol.2, Jurusan Teknik Informatika, Universitas Suryakancana, Cianjur

[2] Ariyus, Dony,(2006) Computer Security, Andi Offset, Jogja,

[3] BA Weyori , S.Akobre and GK ArmahAkobre dan Armah GK, Application of RNS to Huffman's Method of Secured Data Encryption Algorithm, jurnal Meddwell, International Journal of Soft Computing Year:2009, Volume: 4, Issue: 5, Page No.: 197-200 DOI: 10.3923/ijscomp.2009.197.200

[4] Bisson, Jacquelin, Rene Saint-Germain, The BS 7799 / ISO 17799 Standard For a better approach to information security, Callio Technologies, Sherbroke, Canada, 2005

[5] Kendall, Kendall,(2002) System Analisys and Design, Pearson Education, NewJersey,

[6] Schaum,(2002), Schaum'sOutLine of Computer Networking, McGraw Hills, NewYork,

[7] Sinarmata, Janner, (2006) PengamananSistemKomputer, Andi Offset, Jogja

[8] Sommervile, Ian,(2001) Software Engineering, Pearson Education, United Kingdom