

## Algoritma Huffman Dalam Pemampatan Dan Enkripsi Data

Akhmad Pahdi  
STMIK Banjarbaru  
ahmadpahdi@gmail.com

*Abstract - The increasing of computer usage and the opened information in daily life indirectly make a needed of saving data more increasingly, also with data security, with the easier file to read indirectly make data security become more susceptible to know with unknown people, that is why we need a method to do compression and encryption. Huffman's algorithm is one of a famous algorithm to compression in process, we can also say Huffman's Algorithm is an algorithm to do encryption. To compression and data encryption process, there are 3 main phases in the using of Huffman's Algorithm, first the Huffman's creation tree phase, second is change the character series to bit series that is more shorter so we can get smaller file size, and the third phase is change each 8 bit series to become a new character series.*

**Key Word** : Huffman's algorithm, File Compression, File Encryption.

**Abstrak** – Meningkatnya penggunaan komputer dan keterbukaan informasi dalam kegiatan sehari-hari secara tidak langsung membuat kebutuhan akan penyimpanan data semakin meningkat, begitu pula halnya dengan keamanan data, dengan mudahnya file untuk dibaca secara tidak langsung membuat kerahasiaan data menjadi rentan untuk diketahui orang yang tidak seharusnya, oleh karena itu diperlukan sebuah metode untuk melakukan kompresi dan enkripsi data. Algoritma Huffman merupakan salah satu algoritma yang cukup terkenal untuk melakukan kompresi, yang dalam prosesnya, algoritma Huffman juga bisa dikatakan merupakan sebuah algoritma untuk melakukan enkripsi. Untuk proses kompresi dan enkripsi data, terdapat 3 tahapan utama dalam penggunaan algoritma Huffman, pertama tahapan pembentukan pohon Huffman, kedua tahapan mengganti rangkaian karakter menjadi rangkaian bit yang lebih pendek sehingga mendapatkan ukuran file yang lebih kecil, tahapan ketiga merubah setiap 8 rangkaian bit menjadi rangkaian karakter yang baru.

**Kata Kunci** : Algoritma Huffman, Kompresi file, Enkripsi file

### 1. Latar Belakang

Perkembangan teknologi informasi yang pesat telah memberi peran yang sangat penting untuk menjalin pertukaran informasi yang cepat. Kecepatan pengiriman informasi dalam bentuk perpaduan teks, suara dan gambar secara real-time akan menjadi bagian utama dalam pertukaran informasi masa mendatang. Selain itu, kemudahan akses media komunikasi membawa pengaruh terhadap keamanan informasi yang menggunakan media komunikasi sebagai media penyampaian. Informasi menjadi sangat rentan untuk diketahui, diambil atau bahkan dimanipulasi dan disalahgunakan oleh pihak lain yang tidak berhak. Selama pengiriman dan ketika sampai di tujuan, terkadang informasi tersebut harus tetap rahasia dan terjaga keasliannya atau tidak dimodifikasi.

Saat ini pengiriman informasi secara *real-time* terkadang masih mengalami kendala. Di antaranya adalah besarnya jumlah data yang harus dikirim melampaui kecepatan transmisi yang dimiliki oleh perangkat keras yang ada, sehingga masih terdapat delay time yang relatif cukup besar, untuk mentransfer file text dengan

ukuran 27 mb dengan lebar *bandwidth* 265 Kbps maka di perlukan waktu lebih dari 10 menit. Untuk data-data penting penerima informasi harus yakin bahwa informasi tersebut memang benar berasal dari pengirim yang tepat, begitu juga sebaliknya, pengirim yakin bahwa penerima informasi adalah orang yang sesungguhnya. Selain itu penerima tidak ingin pengirim membantah pernah mengirim informasi tersebut, dan jika hal tersebut terjadi penerima perlu membuktikan ketidakbenaran penyangkalan tersebut. Untuk permasalahan-permasalahan kecepatan pengiriman dan keamanan data agar data tidak dapat dibaca dengan mudah tersebut diperlukan suatu metode untuk meningkatkan kecepatan pengiriman serta menjaga keamanan informasi. Salah satu metodenya adalah kompresi dan kriptografi [9]

Algoritma Huffman adalah algoritma yang dapat digunakan untuk melakukan proses pemampatan data untuk menghasilkan ukuran file yang lebih kecil dari ukuran file aslinya, agar ruang penyimpanan dapat lebih dihemat. Algoritma *Huffman* merupakan algoritma yang dinamakan sesuai dengan penemu algoritmanya, Huffman adalah salah satu algoritma kompresi tertua yang

disusun oleh David *Huffman* pada tahun 1952. Algoritma *Huffman* tersebut digunakan untuk membuat kompresi file dengan jenis *loseless compression*, yaitu pemampatan data dimana tidak satu byte pun dari data akan hilang sehingga data tersebut pun dan disimpan sesuai dengan aslinya [5], dengan teknik kompresi *loseless*, informasi pada data hasil kompresi dapat disusun kembali ke bentuk awal data tanpa mengurangi sedikit pun informasi dari file awal.

Enkripsi merupakan upaya untuk mengamankan data/informasi, meskipun enkripsi bukan merupakan satu-satunya cara untuk mengamankan data/informasi. tujuan dari enkripsi adalah untuk menjaga kerahasiaan data dan informasi, untuk menjaga integritas data agar terjamin keaslian/keutuhan data agar tidak dapat dimanipulasi oleh-oleh yang tidak berkepentingan serta tujuan dari enkripsi adalah untuk autentikasi data bahwa benar data yang ada adalah data yang seharusnya [4], proses enkripsi adalah proses merubah data awal (*plaintext*) menjadi data yang sudah tersandikan (*ciphertext*) dan proses sedemikian rupa agar data tidak mudah untuk dibaca oleh yang tidak berkepentingan.

Pemampatan atau kompresi bertujuan merubah ukuran file menjadi file yang lebih kecil ukurannya sedangkan enkripsi adalah untuk menjaga keamanan data/informasi dengan cara merubah *plaintext* menjadi *ciphertext*, algoritma *Huffman* adalah algoritma yang mampu melakukan keduanya, algoritma merubah ukuran data/file menjadi ukuran yang lebih kecil dengan merubah susunan data yang ada menjadi data baru yang sudah tersandikan, dan sebaliknya, algoritma *Huffman* dapat mengembalikan data/file yang sudah diproses ke bentuk awal.

## 2. Kajian Pustaka

Mengecilkan ukuran file digital merupakan tujuan utama dari pemampatan data atau biasa disebut dengan kompresi, dengan kompresi ukuran file yang sebelumnya berukuran besar akan berubah semakin menjadi kecil setelah proses kompresi dilakukan, proses kompresi adalah proses pengkodean informasi menggunakan bit biner pada sebuah atau beberapa file, bit lama akan dikodekan kembali agar menghasilkan bit yang lebih kecil, urgensi dari kompresi adalah pada penghematan media penyimpanan, memperkecil kebutuhan bandwidth ataupun mempercepat proses pengiriman data. [1]

Tipe-tipe file gambar, video audio yang biasa dipergunakan seperti JPEG, PNG, MP3, MPEG

merupakan file-file yang telah mengalami proses kompresi yang termasuk kedalam jenis *Lossy Compression*, dimana terdapat beberapa informasi atau beberapa bit dihilangkan dari file aslinya dan tidak bias dikembalikan ke bentuk aslinya, dengan kata lain file hasil kompresi tidak akan sama dengan file aslinya namun file-file hasil kompresi masih bias dipergunakan. file-file seperti RAR, ZIP, 7-ZIP merupakan tipe kompresi lainnya selain *Lossy Compression*, RAR, ZIP dan 7-ZIP termasuk kedalam jenis *Loseless Compression*, dimana proses kompresi tidak akan menghilangkan informasi apapun dari file aslinya, sehingga file hasil proses kompresi dapat dikembalikan ke bentuk aslinya, biasa disebut dengan proses Kompresi dan proses Dekompresi [6].

*Huffman* merupakan salah satu algoritma yang biasa digunakan untuk proses kompresi data. Salah satu tujuan dari proses kompresi data adalah untuk menjadikan ukuran data lebih kecil dari aslinya, untuk mencapai tujuan tersebut diperlukan terjadi suatu perubahan antara data asli dengan data hasil kompresi, proses kompresi biasanya menghasilkan sebuah data duplikasi dari data asal, hasil duplikasi data kompresi biasanya memiliki bentuk yang berbeda dengan data aslinya. Untuk mengembalikan data hasil kompresi ke data asal kembali diperlukan data yang berkaitan dengan data aslinya [2].

Proses perubahan data dari hasil kompresi menggunakan algoritma *Huffman* data dianalogikan dengan proses enkripsi dalam bidang kriptografi, namun harus dilakukan beberapa adaptasi dalam hal implementasinya. Penerapan algoritma *Huffman* dalam bidang kriptografi dapat dilakukan dengan beberapa metode, namun metode tersebut dapat menghilangkan makna kompresi yang ditujukan untuk demi meningkatkan keamanan, akan tetapi, secara umum yang menjadi kunci untuk melakukan enkripsi dan dekripsi dalam algoritma *Huffman* adalah pohon *Huffman* yang dibentuk [2]

Algoritma kompresi *Huffman* dinamakan sesuai dengan nama penemunya yaitu David *Huffman*, seorang profesor di MIT (*Massachusetts Institute of Technology*). Kompresi *Huffman* merupakan algoritma kompresi *lossless Compression* dimana data yang telah dikompresi dapat dikembalikan ke data seperti semula, algoritma *Huffman* sangat ideal untuk digunakan mengkompresi teks atau *file program*. Hal ini yang menyebabkan mengapa algoritma ini banyak dipakai dalam program kompresi [8].

Kode *Huffman* di kenal sebagai kode yang digunakan untuk melakukan proses kompresi data. Kode ini dibentuk dengan memanfaatkan frekuensi kemunculan string karakter ASCII pada data yang akan dikompresi. Kode yang diperoleh ialah berupa bilangan bit biner yang kemudian diubah ke dalam karakter yang sesuai dengan bilangan biner tersebut. Menurut penelitian Satrio Ajie Wijaya, Algoritma Kriptografi *Huffman-Monoalphabetic* merupakan suatu algoritma monoalphabetic yang memanfaatkan kode *Huffman* sebagai pembentuk kunci untuk melakukan enkripsi. Yang membedakan antara algoritma kriptografi monoalphabetic lainnya dengan algoritma *Huffman-monoalphabetic* adalah substitusi karakter dilakukan dengan memanfaatkan kode *Huffman* tersebut yang dibentuk dari plaintext. Kode *Huffman* tersebut kemudian digunakan sebagai kunci pada algoritma ini, pebsubstitusian huruf dilakukan bukan dengan memanfaatkan karakter, melainkan dengan menggunakan bilangan biner yang diperoleh dari kode *Huffman* [11].

Keamanan data adalah menjadi aspek yang sangat penting dalam sistem teknologi informasi. Transaksi tukar menukar informasi sering kali dilakukan dengan sistem jaringan yang terkoneksi ke dunia global, mengamankan data sangat perlu untuk diperhatikan, pengamanan data dapat dilakukan dengan cara melakukan enkripsi terhadap data agar data tidak mudah untuk diakses, dibuka apalagi untuk dirubah isinya oleh orang yang tidak berkepentingan [3]

Kriptografi merupakan salah satu cabang ilmu dari dunia teknik informatika, kriptografi merupakan cabang ilmu untuk mengamankan data untuk menjaga kerahasiaan agar tidak dapat disalahgunakan oleh orang yang tidak berkepentingan, proses kriptografi dapat dilakukan dengan memanfaatkan metode-metode tertentu. Dalam menjaga kerahasiaan data data asli atau *Plaintext* akan dirubah menjadi data yang disandikan atau *ciphertext*, Hasil deskripsi dari *ciphertext* akan menghasilkan *plaintext* yang sama dengan *plaintext* sebelum dienkripsi. Sedangkan kunci yang digunakan untuk mengenkripsi *plaintext* sama dengan kunci yang digunakan untuk mendeskripsi *ciphertext*. Apabila kunci yang digunakan untuk mengenkripsi tidak sama dengan kunci yang digunakan untuk mendeskripsi, maka hasil dari deskripsi tidak akan sama dengan plaintext semula seperti sebelum dienkripsi [7].

Untuk mengamankan data, dibutuhkan suatu solusi dalam pelaksanaannya. Salah satu metode

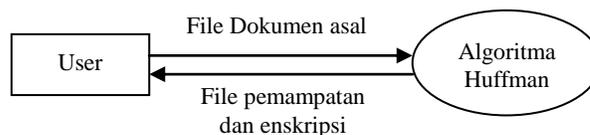
yang digunakan untuk menjaga kewanaman data tersebut adalah kriptografi dengan berbagai algoritmanya. dalam prosesnya dilakukan penyandian data teks, sehingga hanya orang-orang tertentu saja yang dapat memahami dan dapat mengakses data tersebut. Sehingga keamanan data tersebut dapat dijamin kebenarannya. Hasil yang diperoleh dari penelitian ini adalah bahwa dalam menggunakan kriptografi, hanya kunci yang tetap dijaga kerahasiaannya dan hanya orang yang bersangkutan saja yang mengetahuinya. Sedangkan algoritmanya dapat diketahui oleh siapapun tanpa mempengaruhi keamanannya [4].

### 3. Proses Kompresi dan Enkripsi

Kompresi data secara umum merupakan proses yang dapat mengubah sebuah data masukan (sumber data asli) ke data yang lain (data keluaran) yang memiliki ukuran yang lebih kecil, tujuan dari kompresi data adalah untuk merepresentasikan suatu data digital dengan sesedikit mungkin bit, tetapi tetap mempertahankan kebutuhan minimum untuk membentuk kembali data aslinya. Sedangkan Enkripsi merupakan bagian dari algoritma kriptografi yang akan merahasiakan informasi dengan menyandikannya kedalam bentuk yang tidak dimengerti lagi maknanya.

File akan dirubah strukturnya dengan proses pemampatan menggunakan algoritma Huffman dan kemudian dienkripsi menggunakan algoritma Huffman yang digunakan. Proses pengembalian file yang sudah dimampatkan dan proses deskripsi menggunakan algoritma Huffman akan mengembalikan file yang telah di dimampatkan dan di enkripsi ke bentuk file semula dan juga akan mengembalikan ukuran file ke ukuran semula.

Pada sistem yang didalamnya terkandung algoritma Huffman, user memasukkan file inputan, berupa file dokumen berbasis teks, selanjutnya sistem akan memproses file inputan tersebut dan akan memberikan output berupa file baru yang sudah termampatkan dan terenkripsi, sistem ini juga akan dapat mengembalikan file yang termampatkan dan terenkripsi ke bentuk semula.



Gambar 4.1. Diagram Konteks

Pada diagram konteks digambarkan proses umum yang terjadi dalam system. Terdapat 2 komponen utama yaitu *User* dan *Sistem*. *User* akan memberikan masukkan berupa file digital, sedangkan sistem yang dalam hal ini adalah algoritma Huffman akan mengkompresi dan mengenkripsi ataupun sebaliknya sistem akan mendekomposisi dan mendeskripsi file tersebut.

Kode Huffman pada dasarnya merupakan kode prefiks (prefix code). Kode prefiks adalah himpunan yang berisi sekumpulan kode biner yang dalam hal ini tidak ada kode biner yang menjadi awal bagi kode biner yang lain. Kode prefiks biasanya direpresentasikan sebagai pohon biner yang berlabel, dimana setiap sisi diberi label 0 (cabang kiri) atau 1 (cabang kanan). Rangkaian bit yang terbentuk pada setiap lintasan dari akar ke daun merupakan kode prefiks untuk karakter yang berpadanan. Pohon biner ini biasa disebut pohon Huffman. Langkah-langkah pembentukan pohon Huffman sebagai berikut :

1. Baca semua karakter di dalam data untuk menghitung frekuensi kemunculan setiap karakter. Urutkan karakter dari yang paling kecil frekuensi kemunculannya.
2. Gabungkan dua buah pohon yang mempunyai frekuensi terkecil pada sebuah akar. Akar mempunyai frekuensi yang merupakan jumlah dari frekuensi dua buah pohon penyusunnya.
3. Ulangi langkah 2 sampai hanya tersisa satu buah pohon Huffman. Agar pemilihan dua pohon yang akan digabungkan berlangsung cepat, maka semua pohon yang ada selalu terurut menaik berdasarkan frekuensi.
4. Kodekan tiap karakter dengan susunan bit biner hasil dari pohon Huffman
5. Ganti susunan karakter (kalimat) awal dengan susunan bit biner yang dihasilkan dari pohon biner dilangkah ke 2

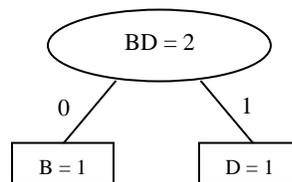
Enkripsi merupakan bagian dari algoritma kriptografi yang akan merahasiakan informasi dengan menyandikannya kedalam bentuk yang tidak dimengerti lagi maknanya. proses mengganti susunan karakter menjadi susunan bit merupakan proses enkripsi yang dimiliki oleh algoritma Huffman, selain untuk membuat file yang lebih kecil dari bentuk awal proses penggantian susunan karakter dengan susunan bit merupakan proses enkripsi.

Dasar pemikiran algoritma kompresi Huffman adalah bahwa setiap karakter ASCII biasanya diwakili oleh 8 bits. Jadi misalnya suatu file berisi deretan karakter "ABACAD" maka ukuran file tersebut adalah 6 x 8 bits = 48 bit atau 6 bytes.

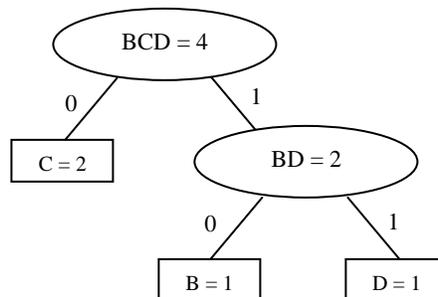
Jika setiap karakter tersebut di beri kode lain secara sembarang, misalkan A=1, B=00, C=010, dan D=011, berarti hanya perlu file dengan ukuran 11 bits (10010101011). Untuk menentukan kode-kode pengganti dari karakter ASCII yang tersebut dengan kriteria bahwa kode harus unik dan karakter yang sering muncul dibuat kecil jumlah bitnya, hal tersebut dapat menggunakan algoritma Huffman.

Sebagai contoh, dalam kode ASCII string 7 huruf "ABACCCA" akan membutuhkan ruang sebesar 56 bit atau 7 byte, karena setiap 1 huruf menempati 8 bit atau 1 byte, maka susunan huruf "ABACCCA" akan mempunyai ukuran file sebesar 7 byte. Dengan algoritma Huffman akan dibentuk pohon binier Huffman, setiap karakter akan diganti dengan kode biner sebagai berikut :

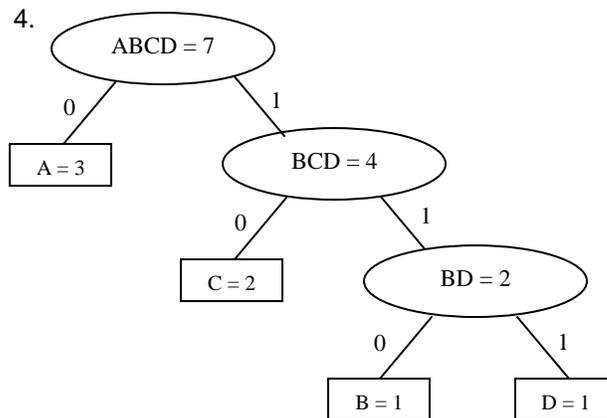
1. A = 3      B = 1      C = 2      D = 1
2. A = 3      C = 2



3. A = 3



- 4.



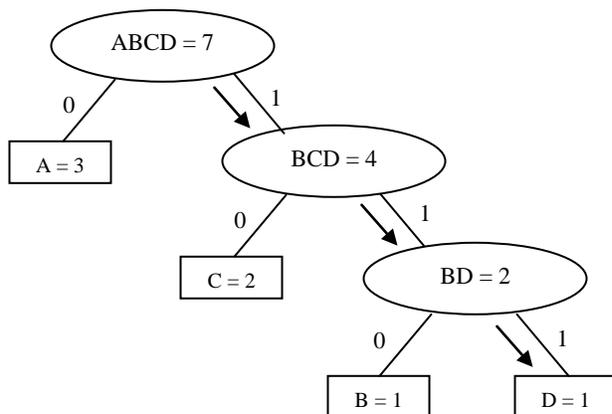
Gambar 4.2. Pohon Huffman untuk Karakter ABACCCA

Tabel 4.1 Kode pengganti berdasarkan pohon Huffman

Karakter	String Biner Pengganti
A	0
B	110
C	10
D	111

Dari pohon biner di atas, maka susunan karakter "ABACCCA" yang menggunakan ruang sebesar 56 bit (7 bit) akan digantikan dengan susunan bit biner 01100110101110 yang cuma akan menggunakan ruang sebesar 13 bit atau sekitar 2 byte, dan menghasilkan file baru yang sudah terenkripsi dengan kunci enkripsi adalah tabel pengganti karakter karakter yang dihasilkan dari proses pembentukan pohon Huffman.

Proses dekompresi atau mengembalikan file yang sudah dimampatkan ke bentuk file awal, juga menggunakan pohon biner pada proses kompresi dengan cara menelusuri pohon biner tersebut mulai dari akar.



Gambar 4.3. Proses dekompresi

Sebagai contoh kita medekompresi string bernilai 111, setelah ditelusuri menggunakan pohon biner mulai dari akar, maka diketahui string bernilai 111 adalah karakter D (lihat gambar 3.6). Cara dekompresi kedua adalah dengan cara memperhatikan tabel kode Huffman hasil dari pohon biner Huffman (lihat tabel 3.1), maka proses dekompresi dari susunan biner 0110110101110 dapat dilakukan dengan mudah, prosesnya sebagai berikut, baca susunan biner 0110110101110, bit pertama dari susunan tersebut adalah 0 yang merupakan pengganti dari karakter A, baca bit berikutnya, yaitu 1, tidak ada kode Huffman 1 pada tabel maka baca bit

berikutnya sehingga menjadi 11, 11 juga tidak ditemukan, maka baca lagi bit berikutnya sehingga menjadi 110, rangkaian bit 110 diketahui adalah pengganti dari karakter B, baca bit berikutnya yaitu 0 yang merupakan pengganti dari karakter A, lakukan pembacaan bit sampai bit terakhir sehingga proses dekompresi akan menghasilkan susunan karakter semula yaitu ABACCCA..

#### 4. Hasil Pembahasan

File yang diuji coba dalam sistem adalah file dokumen yang berbasis teks, dimana isi dari file dokumen tersebut di dominasi oleh susunan-susunan string karakter yang membentuk sebuah kalimat, file yang di uji coba adalah file Microsoft Excel 2003, Microsoft Excel 2007, Microsoft Word 2003, Microsoft Word 2007, Microsoft Power Point 2003 dan file sql, daftar file dimasukkan ke dalam sistem yang sudah di sertakan algoritma Huffman didalamnya, file sampel akan di dapatkan sekaligus di enkripsi agar ukuran file menjadi leih kecil dari ukuran file sebelumnya serta hasil dari proses pemampatan akan menghasilkan file baru yang sudah terenkripsi.

Nama File	Ukuran File Asal	Ukuran File Baru	Persentase Kompresi
E:\ceang.html	0,29 KB	0,24 KB	17,47 %
E:\tes.html	1,24 KB	0,85 KB	30,45 %
E:\14101585-2.PB.pdf	179,10 KB	172,71 KB	1,33 %
E:\115314040_full.pdf	3,037,03 KB	3,037,05 KB	- %
E:\Bahan.dl	24,50 KB	10,25 KB	58,15 %
E:\Praktek1.xlsx	9,41 KB	8,35 KB	11,28 %
E:\Algoritma.doc	32,00 KB	16,19 KB	49,42 %
E:\Huffman.doc	34,00 KB	17,45 KB	48,68 %
E:\Penelitian.doc	15,50 KB	8,88 KB	42,69 %
E:\komputer.txt	0,34 KB	0,20 KB	41,57 %
E:\bunian.txt	1,33 KB	0,77 KB	41,99 %
E:\Vitrakibergasbau.txt	0,88 KB	0,51 KB	42,83 %
E:\Unggan 1.ppt	203,50 KB	136,46 KB	32,94 %
E:\Semester 2.pptx	101,98 KB	96,85 KB	4,94 %
E:\gauri.sql	454,26 KB	248,03 KB	45,40 %
E:\viral.mysql.utf8.sql	102,00 KB	68,01 KB	33,32 %

Gambar 4.4. Proses Kompresi dan Enkripsi

Persentasi kompresi sangat bervariasi, efektivitas kompresi tergantung dari isi file tersebut serta susunan karakter yang terkandung didalam file tersebut, semakin besar file tersebut tetapi variasi susunan karakter didalam file tersebut sedikit maka hasil kompresi akan bisa semakin maksimal, hal sebaliknya apabila file tersebut kecil dan memiliki sedikit karakter sedangkan variasi karakter banyak, maka hasil kompresi akan kurang maksimal, dan apabila dalam file tersebut terdapat sebuah objek yang bukan merupakan sebuah teks maka hasil kompresi juga akan kurang maksimal.

Tabel 4.2. informasi file hasil pengujian

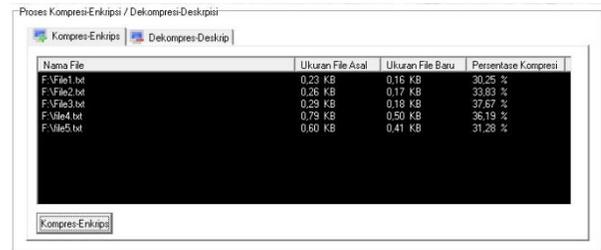
No	Nama File baru	File Lama (KB)	File Baru (KB)	Kompresi (%)
1	cang.phd	0.29	0.24	17.47
2	tes.phd	1.24	0.86	30.45
3	1410-1585-2PB.phd	179.10	176.71	1.33
4	115314040.phd	3037.03	3037.05	0
5	Bahan.phd	24.50	10.25	58.15
6	Praktek1.phd	9.41	8.35	11.28
7	Algoritma.phd	32	16.18	49.42
8	Huffman.phd	34	17.45	48.68
9	Penelitian.phd	15.50	8.88	42.69
10	komputer.phd	0.34	0.20	41.57
11	liburan.phd	1.33	0.77	41.90
12	stmikbanjarbaru.phd	0.88	0.51	42.83
13	Uangan.phd	203.50	136.46	32.94
14	Semester2.phd	101.88	96.85	4.94
15	gaji_rs.phd	454.26	248.03	45.40
16	install.mysql.utf8.phd	102	68.01	33.32

Pada hasil uji coba file dokumen yang disimpan dengan format Microsoft office 2007 dan Microsoft Office 2013 memiliki persentasi kompresi yang sangat kecil, berbeda dengan file dokumen yang disimpan dengan format Microsoft office 2003 persentasi kompresi yang dihasilkan cukup maksimal, hal tersebut dikarenakan pada file Microsoft Office 2007 file tersebut mempunyai format OpenXML dimana file openXML merupakan file yang sudah terkompresi. Secara umum persentase hasil kompresi sangat maksimal apabila file tersebut adalah file dengan ukuran besar dan file tersebut hanya terdiri dari susunan karakter.

Enkripsi adalah proses untuk mengamankan suatu informasi dengan cara mengacaknya, mengubahnya dari teks terang atau teks asal (*plaintext*) menjadi teks tersandi (*ciphertext*), sehingga tidak dapat dibaca pihak lain tanpa pengetahuan khusus.

Secara teori, proses enkripsi yang dilakukan oleh algoritma Huffman adalah dengan mengganti susunan karakter menjadi susunan bit biner, kemudian dari susunan bit biner tersebut setiap delapan bit nya di rubah kembali menjadi susunan karakter baru sesuai dengan tabel ascii.

Berikut beberapa hasil enkripsi yang dilakukan oleh algoritma Huffman yang dimasukkan kedalam sistem :



Gambar 4.5. Proses Kompresi dan Enkripsi

File sampel yang dijadikan bahan uji coba adalah file bertipe txt, file sample dari file 1 sampai dengan file 5 berhasil di mampatkan sekaligus file asli dirubah menjadi file baru dimana file yang baru tidak bias di baca secara langsung, hal ini sesuai dengan prinsip enkripsi, file akan bias dibaca kembali melalui proses deskripsi sekaligus proses dekompresi.

Tabel 4.3. Tampilan proses Komresi & Enkripsi

File sebelum di kompres & enkripsi	File setelah di kompres & enkripsi	Kompresi
File1.txt :  Enkripsi adalah proses untuk mengamankan suatu informasi dengan cara mengacaknya, mengubahnya dari teks terang atau teks asal ( <i>plaintext</i> ) menjadi teks tersandi ( <i>ciphertext</i> ), sehingga tidak dapat dibaca pihak lain tanpa pengetahuan khusus.	File.txt.phd  ^-üh:Ûtny.Emsfjchi tdgprl(),boxeukaî diª...ÖOo.ljË#½k>ð é±óz,+  ½Jîè4Ê#Åx>ðÅoi~`dè ^,Ó«:f³cé'ýørifÿß, Uš+buÑ6ÿÖ1,SùHê - ~ENËéG»c-©ÈùMß N/'i^¼«Éo7W8žr— Až,\#·¼~	30%

## 5. Penutup

Berdasarkan hasil percobaan dari sistem yang dibangun serta berdasarkan hasil yang didapatkan dalam uji implementasi diatas, maka dapat diambil beberapa kesimpulan :

1. Algoritma Huffman adalah salah satu algoritma yang dapat di gunakan untuk melakukan kompresi sekaligus enkripsi file, ukuran file hasil dari proses menjadi lebih kecil sekaligus terenkripsi menjadi susunan karakter yang baru dan tidak dapat dibaca dengan mudah.
2. Hasil proses kompresi yang dilakukan dengan aplikasi yang dibuat sangat maksimal apabila file dokumen berbasis teks tersebut berukuran besar yang hanya berisi susunan karakter.

Aplikasi ini akan sangat tidak maksimal apabila di dalam file dokumen berbasis teks tersebut terdapat sebuah objek yang bukan merupakan teks.

## 6. Pustaka

- [1] Adhayanti, N. (2015). *Kompresi Data dan Teks*.
- [2] Adrisatria, Y. (2015). *PENERAPAN ALGORITMA HUFFMAN DALAM DUNIA KRIPTOGRAFI*. Bandung: Program Studi Teknik Informatika, Institut Teknologi Bandung.
- [3] Eka, A. A. (2013). *Teknik Pengamanan Data pada Sistem Proses Surat dengan Metode Blowfish di Badan Kepegawaian Daerah Jawa Tengah*. Semarang: Fakultas Ilmu Komputer Universitas Dian Nuswantoro.
- [4] GIRSANG, T. T. (2010). *ANALISIS KERAHASIAAN DATA MENGGUNAKAN ALGORITMA VIGENERE CIPHER DALAM SISTEM PENGAMANAN DATA*. Medan: UNIVERSITAS SUMATERA UTARA.
- [5] Neta, M. R. (2013). Perbandingan Algoritma Kompresi Terhadap Objek Citra Menggunakan JAVA. *SEMANTIK 2013*, 224-230.
- [6] Oky Dwi Nurhayati, S. M. (n.d.). *Kompresi Data*. Semarang: Universitas Diponegoro.
- [7] Patricia Handoko, d. (n.d.). Teknik Keamana Data Menggunakan Kriptografi dengan Algoritma Vigene Cipherdan Steganografi dengan Metode End Of File

(EOF). *Fakultas Ilmu Komputer Universitas Dian Nuswantoro*, 1-7.

- [8] Purwanto, H. (n.d.). *PENERAPAN ALGORITMA HUFFMAN PADA KOMPRESI FILE WAVE*.
- [9] Suriski Sitinjak, d. (2010). APLIKASI KRIPTOGRAFI FILE MENGGUNAKAN ALGORITMA BLOWFISH. *semnasIF*, 78-86.
- [10] Timothy John Pattiasina, S. M. (n.d.). ANALISA KODE HUFFMAN UNTUK KOMPRESI DATA TEKS. *IKADO-TECH*, 1-12.
- [11] Wijaya, S. A. (n.d.). *Peancangan Algoritma Kriptografi Huffman-Monoalphabetic*. Bandung: Jurusan Teknik Informatika ITB.