Pemampatan Data Sebagai Bagian Dari Kriptografi

Muhammad Ismail Faruqi, Adriansyah Ekaputra, Widya Saseno

Laboratorium Ilmu dan Rekayasa Komputasi
Departemen Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : {if13045, if13021, if13002}@students.if.itb.ac.id

Abstrak

Dalam beberapa tahun belakangan ini teknologi informasi telah banyak berperan serta dalam beberapa aspek kehidupan manusia. Banyaknya dokumen yang dihasilkan tiap harinya dapat melebihi jumlah penduduk didunia ini. Penyimpanan dalam media kertas kini sudah mulai ditinggalkan dan beralih pada media lainnya yaitu media elektronik salah satunya berupa dokumen komputer atau *file* komputer. Privasi atau kerahasiaan dokumen menjadi sangat kurang ketika dokumen disimpan dalam bentuk elektronik karena akan sangat mudah digandakan dan di*edit* oleh pihak lain, terutama bila telah tempat dokumen tersebut disimpan terhubung pada jaringan *internet*. Kompresi data dan enkripsi data telah banyak dipakai sebagai alternatif solusi dalam mengatasi permasalahan ini. Makalah ini membahas mengenai dokumen yang telah dikompres (dengan memakai algoritma *huffman*) sebagai dokumen yang merupakan *chipertext* yang cukup susah untuk dipecahkan dan juga merupakan salah satu alternatif dalam menjawab permasalahan mengenai jumlah dokumen yang semakin banyak dan privasi suatu dokumen, merupakan pandangan kami yang melihat adanya kesamaan antara proses kompresi dokumen dengan proses kriptografi suatu dokumen.

***Kata kunci**: kompresi data, enkripsi, algoritma huffman, chipertext*

1. Pendahuluan

Teknologi informasi merupakan sesuatu hal yang berkembang dengan pesat dalam beberapa tahun terakhir ini. Sebagian besar penduduk di dunia ini telah merasakan manfaat atau kemudahan yang dihasilkan dengan memakai teknologi ini. Salah satu keuntungan dari teknologi ini yang sering kita alami adalah kini kita dapat mencari sumber bacaan atau referensi yang kita perlukan tanpa harus pergi jauh dan ke beberapa tempat dimana bacaan tersebut kita perkirakan berada.

Dengan *internet* kini bacaan ataupun dokumen yang ingin kita baca bisa kita dapatkan. Disadari atau tidak, hal yang penting dalam situasi ini adalah banyaknya dokumen yang kini telah dalam wujud atau bentuk elektronik. Secara perlahan media kertas mulai ditinggalkan dan beralih pada media elektronik. Media kertas yang memakan banyak tempat dan memerlukan perawatan yang lebih dibandingkan media elektronik merupakan alasan mengapa media elektronik mulai ditinggalkan.

Teks ataupun tulisan merupakan dokumen yang paling banyak dibuat dibandingkan dengan dokumen gambar ataupun lainnya. Setiap harinya banyak dokumen yang berupa teks atau tulisan yang dihasilkan. Seiring dengan pertumbuhan dari dokumen elektronik ini tibul setidaknya dua masalah ini yaitu, pertama dari banyak dokumen yang dihasilkan beberapa diantaranya merupakan dokumen yang sifatnya rahasia dan pribadi.

Hal ini tidak mengkuatirkan bila dokumen itu digunakan hanya untuk keperluan pribadi saja (tanpa perlu orang lain untuk mengetahui) karena dapat memanfaatkan fungsi *password* pada dokumen dan hanya diri kita yang mengetahui *password* itu, tetapi kini yang menjadi hal yang perlu diperhatikan adalah ketika dokumen yang bersifat rahasia ini diperlukan atau digunakan oleh banyak pihak sehingga *password* yang mengamankan dokumen ini kini harus diketahui oleh beberapa orang. Dengan semakin banyak orang yang mengetahui *password* yang ada maka keamanan dokumen tersebut menjadi berkurang.

Lebih lagi ketika dokumen tersebut harus dikirimkan melalui orang lain yang tidak diharapkan mengetahui isi dokumen itu, diperlukan lebih dari sekedar *password* untuk membuka dokumen itu. Diperlukan suatu penyandian sehingga isi dokumen itu tidak diketahui orang lain.

Permasalahan kedua yang timbul yaitu diperlukan media penyimpanan elektronik yang dapat mecukupi kebutuhan penyimpanan dokumen elektronik tersebut.

Permasalahan tersebut dapat diatasi dengan algoritma pemampatan data dalam hal ini data berupa teks atau tulisan. Salah satu algoritma yang dipakai adalah algoritma *Huffman*.

2. Penyadian Data atau Kriptografi

Kriptografi merupakan salah satu dari cara pengamanan data. Pada evolusi pengamanan data berkembang dua cara yaitu *steganography* dan *cryptography*. *Steganography* dikenal sebagai metoda seolah-olah data atau pesan tidak ada [1] sedangkan *cryptography* atau kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti keabsahan, integritas data, serta autentikasi data. Kriptografi tidak berarti hanya memberikan keamanan informasi saja, namun lebih ke arah teknik-tekniknya[2]. Pada setiap kriptografi terdapat kunci yang berguna untuk membuat *plaintext* menjadi sebuah *chipertext* dan mengembalikannya kembali menjadi *plaintext* semula. Kriptografi sendiri sebenarnya terbagi menjadi dua bagian yaitu *transposition (letters arranged)* dan *substitution (letters substituted with other letters)*. *transposition* adalah penyandian dengan merubah susunan alphabet yang digunakan, seperti yang dilakukan oleh Julius Caesar seorang kaisar Romawi untuk menyandikan pesan yang ia kirim kepada para gurbernurnya. Pada *caesar chiper*, tiap huruf disubtitusi dengan huruf ketiga berikutnya dari susunan alphabet. Dalam hal ini kuncinya adalah jumlah pergeseran huruf (yaitu 3)[3]. Selain yang telah digunakan oleh Julius Caesar, kritografi juga telah digunakan oleh bangsa Sparta yang disebut dengan *Spartan Scytale*, bangsa Jerman dengan *Enigma*[1].

Ada empat tujuan mendasar dari ilmu kriptografi ini yaitu :

- Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas.
- Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
- Autentikasi, adalah berhubungan dengan identifikasi, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.
- Non-repudiasi[2].

Notasi untuk menggambarkan proses atau fungsi enkripsi adalah :

$$E_{k1}(P) = C$$

Dimana k1 merupakan kunci yang dipakai dalam proses enkripsi P (*plaintext*) dan menghasilkan C (*chipertext*). Sedangkan proses atau fungsi untuk mengembalikan dari *chipertext* menjadi *plaintext* di notasikan sebagai berikut :

$$D_{k2}(C) = P$$

Dimana k2 merupakan kunci yang dipakai dalam proses dekripsi C (*chipertext*) menjadi P (*plaintext*).

Kedua fungsi diatas memenuhi

$$D_{k2}(E_{k1}(P)) = P$$

Berdasarkan kunci yang digunakan pada proses enkripsi dan dekripsi maka algoritma yang digunakan dalam proses kritografi ini dibagi menjadi dua bagia yaitu *symmetric-key* dan *asymmetric-key*.

Symmetric-key adalah penggunaan kunci yang sama pada proses enkripsi dan proses dekripsi (k1 = k2). Algoritma yang memakai kunci yang sama dalam proses enkripsi dan dekripsi adalah algoritma *DES (Data encryption Standard)*.

Asymmetric-key adalah penggunaan kunci yang berbeda pada proses enkripsi dan dekripsi (k1 <> k2). Algoritma yang memakai metoda *asymmetric-key* ini adalah algoritma *RSA (Rivest-Shamir-Adleman)*.

3. Pemampatan Data atau Kompresi Data

Ide dasar dari pemampatan data adalah mengkodekan kembali setiap karakter didalam dokumen atau pesan dengan kode yang lebih pendek sehingga dapat memperkecil ukuran dokumen. Dokumen atau file yang dibuat dalam komputer atau media elektronik lain seperti *PDA (Personal Digital Asisten)* yang dalam hal ini merupakan dokumen teks atau tulisan merupakan kumpulan kode yang memiliki mewakili setiap karakter dalam dokumen atau pesan. Kode yang biasa dan banyak digunakan adalah kode ASCII *(American Standard Code for Information Interchange)*. Dalam kode ASCII, setipa karakter dikodekan menjadi delapan bit biner. Contoh :

Simbol	*Kode ASCII*
g	01100111
o	01101111
p	01110000
h	01101000
e	01100101
r	01110010

Berdasarkan metode pengkodean yang menggunakan kode ASCII maka akan diperlukan sebanyak 48 bit atau 6 byte untuk merepresentasikan enam huruf atau karakter dalam dokumen. Agar dokumen berukuran kecil maka diperlukan pengkodean yang merepresentasikan karakter didalamnya menjadi lebih sedikit. Sebagai contoh misalkan dalam dokumen tersebut hanya memakai karakter atau huruf yang disebutkan diatas maka karakter tersebut dapat direpresentasikan menggunakan kode dengan tiga bit [4]

Simbol	*Kode binary*
g	000
o	001
p	010
h	011
e	100
r	101

Dengan menggunakan kode diatas maka pesan *'gopher'* yang mengandung enam karakter atau huruf dapat direpresentasikan dengan 18 bit saja atau kurang dari 3 byte. Penurunan yang drastis antara pengkodean dengan memakai kode ASCII dengan memakai kode kedua ini.

Penggunaan kode yang lebih efisien merupakan ide yang mendasari pemampatan atau kompresi data yang dalam hal ini lebih kepada dokumen berisi teks atau tulisan. Algoritma yang telah ada beberapa diantaranya adalah algoritma *LZW* dan *Huffman*.

Algoritma *Huffman* membuat kode yang baru berdasarkan kekerapan kemunculan karakter dalam dokumen yang akan dimampatkan atau dikompresi. Algoritma ini merepresentasikan karakter yang kerap muncul dalam dokumen dengan kode yang lebih sedikit dibandingkan karakter lain yang jarang muncul dalam dokumen tersebut. Karakter lain yang tidak digunakan dalam dokumen tidak ikut dikodekan. Dengan cara seperti ini maka ukuran dokumen dapat dimampatkan menjadi lebih kecil. Contoh pemampatan pesan dengan memakai algoritma *Huffman* misalnya suatu pesan yang berisi 'ABACCDA'. Pesan ini bila direpresentasikan dalam bit yang memakai kode ASCII maka akan dituliskan sebagai berikut :

```
010000010100000100100000101000001101000
001101000100010000001
```

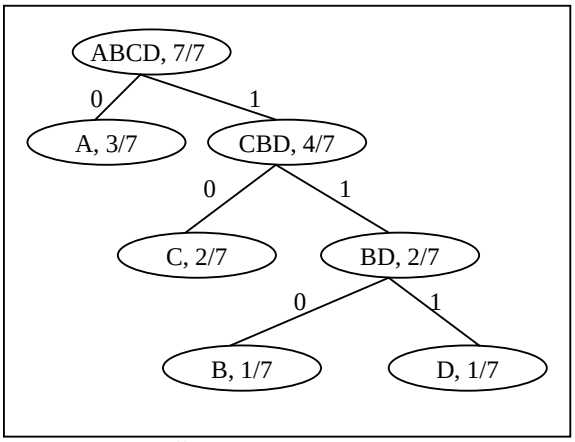
Dengan menggunakan algortima *Huffman* kita dapat mengkodekan pesan diatas dengan rangkaian bit yang lebih sedikit, yaitu :

```
0110010101110
```

Kode yang dipakai diatas didapatkan dari penggunaan algoritma *Huffman* yang menghasilkan tabel sebagai berikut :

Simbol	*Kekerapan*	*Peluang*	*Kode Huffman*
A	3	3/7	0
B	1	1/7	110
C	2	2/7	10
D	1	1/7	111

Algortima *Huffman* merepresentasikan simbol atau karakter yang sering muncul dengan kode yang lebih pendek daripada kode untuk simbol lain. Kode untuk setipa simbol tidak boleh merupakan awalan dari kode yang lain karena hal ini akan menimbulkan abiguitas atau keraguan dalam proses pengembalian atau *decoding* menjadi file semula. Untuk mendapatkan kode *Huffman*, mula-mula kita harus menghitung dulu kemunculan setiap karakter atau simbol dalam dokumen atau pesan yang akan dimampatkan kemudian nantinya kita akan membentuk pohon biner yang menggambarkan kode untuk pemampatan *Huffman* ini. Dengan menghitung kekerapn kemunculan simbol atau karakter maka dapat kita hitung peluang munculnya simbol atau karakter tersebut dalam pesan atau dokumen. Langkah selanjutnya adalah memilih simbol atau karakter yang memiliki peluang yang paling kecil diantara simbol atau karakter lainnya dalam teks (misal dalam contoh tadi adalah simbol *B* dan *D*). Kedua simbol tersebut dikombinasikan sebagai simpul orang tua dari simbol B dan D sehingga menjadi simbol BD dengan peluang 1/7 + 1/7 = 2/7, yaitu jumlah peluang kedua anaknya. Simabol baru ini diperlakukan sebagai simpul baru dan diperhitungkan dalam mencari simbol selanjutnya yang memiliki peluang paling kecil. Langkah selanjutnya adalah dengan memilih kembali dua simbol berikutnya yang memiliki peluang yang paling kecil. Pada contoh ini dua simbol tersebut adalah C (peluang = 2/7) dan BD (peluang = 2/7). Lakukan hal yang sama seperti lankah sebelumnya sehingga dihasilkan simbol baru CBD dengan peluang 2/7 + 2/7 = 4/7. Prosedur yang sama dilakukan pada dua simbol berikutnya yang mempunyai peluang terkecil, yaitu A (peluang = 3/7) dan CBD (peluang = 4/7) sehingga menghasilkan simpul ACBD, yang merupakan akar dari pohon *Huffman* dengan peluang 3/7 + 4/7 = 7/7 [3].



Pohon Huffman untuk pesan "ABACCDA"

Setelah mendapatkan pohon *Huffman* maka tabel kekerapan dan kode *Huffman* dapat dibuat juga.

Proses yang kita lakukan tadi adalah proses encoding dokumen kedalam bentuk dengan ukuran bit yang lebih kecil. Untuk menguraikan kembali pesan atau decoding menjadi bentuk awal sehingga dapar dipergunakan, dibutuhkan data yang menyimpan tentang informasi yang dipakai dalam proses encoding. Informasi ini disimpan dalam

Header dokumen. Header dokumen dapat berisi informsai yang berupa jumlah karakter-yang-dipakai yang terdapat dalam dokumen tersebut, kekerapan kemunculan huruf dalam dokumen, dan juga dapat berupa pohon *Huffman* yang telah dibentuk sebelumnya. Informasi ini sangat penting untuk disertakan pada dokumen yang telah dikompresi karena dengan informasi ini maka dokumen dapat dibentuk menjadi dokumen semula yang dapat dibaca.

4. Analisis

4.1. Kesamaan Antara Penyandian Data Dengan Pemampatan Data

Dari penjelasan pada bab-bab sebelumnya maka dapat disimpulkan terdapat kesamaan antara penyandian dan pemampatan data yaitu dokumen (dalam hal ini dokumen yang berisi teks atau tulisan) akan diubah ke dalam bentuk lain sehingga tidak dapat langsung dipergunakan atau dalam hal ini dibaca. Dalam proses tersebut sama-sama melakukan dua proses yaitu merubah bentuk awal dokumen menjadi bentuk lain yang tidak dapat secara mudah atau langsung dipergunakan dan proses kedua yaitu mengembalikan dokumen yang telah dirubah menjadi bentuk semula.

Proses penyandian data atau kriptografi ditujukan untuk mencegah orang lain untuk mengetahui isi pesan atau dokumen. Dalam proses ini diperlukan kunci untuk mengubah *plaintext* menjadi *chipertext* dan mengembalikan kembali dari *chipertext* menjadi *plaintext*. Pemakaian kunci untuk menjaga kerahasiaan dokumen juga dipakai dalam proses pemampatan data, hanya saja tujuan dari proses ini bukan untuk mencegah orang lain untuk mengetahui isi dokumen atau pesan tetapi lebih kepada tujuan untuk mengurangi ukuran dokumen (dalam hal ini dalam ukuran bit atau byte).

Jadi sebenarnya ketika data atau dokumen dikompresi maka data tersebut tidak dapat lagi dibaca atau dipergunakan lagi secara langsung dan hal itu merupakan salah satu tujuan dari kriptografi dimana data disamarkan.

4.2. Proses Kompresi Data Sebagai Proses Penyandian Data

Hal yang ingin kami angkat dalam makalah ini adalah pandangan kami mengenai proses kompresi data yang ternyata dapat digunakan juga sebagai bagian dari proses kriptografi dengan tujuan melindungi data dari pihak lain. Keuntungan yang didapatkan dari hal ini adalah ukuran data atau dokumen yang menjadi lebih kecil atau sedikit dibandingkan ukuran awalnya sehingga mudah untuk dipindahkan dari suatu tempat (komputer) ke tempat lain (komputer lain) melalui jaringan *internet* tanpa mengurangi segi keamanan dan kerahasiaan data atau dokumen tersebut.

Metoda yang kami diskusikan dalam proses kompresi data atau dokumen adalah algoritma *Huffman* yang mungkin sampai saat ini banyak dipakai dala proses kompresi data berupa teks. Pada proses ini dihasilkan suatu tabel ataupun pohon *Huffman* yang merupakan peraturan atau *rule* dalam merepresentasikan karakater yang berada dalam dokumen menjadi kode dalam binary. Kami perhatikan bahwa hasil dari proses kompresi ini sebenarnya tidak dapat dipergunakan atau dibaca langsung, diperlukan proses pengembalian menjadi data atau dokumen yang semula. Dalam proses ini diperlukan peraturan atau *rule* yang sama pada saat data atau dokumen tersebut dikompresi.

Peraturan atau *rule* yang digunakan berbeda antara data atau dokumen yang satu dengan yang lainnya apabila isi pesan tersebut tidak sama. Perbedaan satu symbol saja dapat memuat perbedaan yang cukup jauh antara table *Huffman* data atau dokumen yang satu dengan yang lain. Karakteristik ini yang menjadikan proses kompresi data atau dokumen ini menyerupai sebuah proses kriptografi, hanya saja dalam proses kompresi data atau dokumen informasi mengenai peraturan atau *rule* yang digunakan dalam proses *encoding* disertakan dalam *header* data atau dokumen untuk memudahkan proses *decoding* data atau dokumen tersebut.

Apabila peraturan atau *rule* dalam proses kompresi data atau dokumen kita rahasiakan maka data atau dokumen tersebut sekarang tidak tidak dapat dipergunakan oleh orang yang tidak mengetahui peraturan atau *rule* untuk mengembalikan menjadi data yang siap dipergunakan. Pemakaian peraturan atau *rule* yang sama dalam proses *encoding* dan *decoding* memiliki kesamaan dengan proses kriptografi metoda *symmetric-key*. Maka sekarang proses kompresi data atau dokumen tadi menjadi bagian juga dari proses kriptografi.

Agar data atau dokumen yang telah dikompresi dapat digunakan orang lain maka tabel atau pohon *Huffman* yang dipakai untuk mengkompres data tersebut harus juga diberikan. Data itu akan aman (tidak dapat dibaca, bahkan dipakai) dari orang atau pihak lain selama orang atau pihak tersebut tidak mengetahui pohon atau tabel *Huffman* yang dipakai dalam proses kompresi pada data atau dokumen tersebut.

4.3. Pengembangan

Awalnya kami melihat bahwa dengan menghilangkan *Header-Information* pada data atau dokumen yang telah dikompresi maka data tersebut tidak dapat digunakan oleh orang lain selama orang tersebut tidak mengetahui tabel ataupun pohon *Huffman* yang digunakan pada proses kompresi data atau dokumen tersebut.

Metoda ini ternyata mengarah pada metoda *symmetric-key* pada kriptografi. Tabel atau poon *Huffman* yang dipakai saat proses *encoding* harus

diberitahukan kepada orang lain yang membutuhkan data tersebut untuk kembali di-*decodingkan*.
Metoda seperti ini tidak begitu populer diabandingkan dengan metoda yang menggunakan kunci yang berbeda atau *asymmetric-key*. Kemungkinan untuk pengembangan keamanan yang kami pikirkan adalah penerapan kriptografi metoda *asymmetric-key* untuk mengamankan tabel atau pohon *Huffman* yang dimasukkan dalam *Header-Information* pada data atau dokumen. Dengan metoda seperti ini maka data akan mudah untuk dikirmkan ke orang atau pihak lain dan keamanan data tetap dapat dijaga.
Masalah kurangnya media untuk menyimpan informasi kini juga dapat diatasi. Karena data atau dokumen dapat dikompres terlebih dahulu sebelum disimpan.
Dalam teknologi informasi yang berkembang dengan pesat sekarang ini maka bukan tidak mungkin ditemukan algoritma pengkompresian data yang lebih abik lagi, yang kami harapkan adalah pengembangan dari ide kami ini dengan menggunakan algoritma tersebut.

5. Kesimpulan

Hasil analisis kami memperlihatkan bahwa pemakaian proses kompresi data untuk keamanan dan kemudahan dalam pengiriman data atau dokumen ternyata dapat dipakai. Analisis kami memakai algoritma *Huffman* untuk mengkompresi data atau dokumen dan menghilangkan *Header-Information* pada data atau dokumen untuk mencegah orang lain untuk mengetahui isi dari data atau dokumen tersebut.
Keamanan dari data atau dokumen akan lebih ketat lagi apabila *Header-Information* yang ditaruh didalam data atau dokumen di-encripsi dengan memakai *asymmetric-key*.
Implementasi dari analisis ini memang belum kami buktikan karena keterbatasan kami, tatapi kami yakin bahwa metoda ini dapat diimplementasikan dan merupakan salah satu alternatif jawaban dalam permasalahan mengenai ukuran data atau dokumen dan keamanan suatu data atau dokumen.

Daftar Pustaka

[1] FEB 2005 PENGANTAR KRIPTOGRAFI - INDOCISC
[2] http://id.wikipedia.org/wiki/Kriptografi
[3] Rinaldi Munir, *Diktat Kuliah Matematika Diskrit*, 2003
[4] Huffman Coding A CS2 Assignment