Kompresi Data Berdasarkan Perhitungan Distribusi Probabilitas Kemunculan Karakter Orde Dua Dalam Teks Bahasa Indonesia

Yuli Christyono

Abstract: In modern world, the need for capacity data storage and electronic data communications channel is important. Capacity storage media and channel communication in electronic data that we use at this time is not unlimited and it is quite expensive.
Therefor, we need a method to use electronic data storage and communications resource optimally and efficiently. One effort that can do at this time is to perform data compression. Data will be compressed before store or before send, so that storage capacity of the data after compression is smaller than before and economize time to send data after compression. There are many kind of method which used in data compression.This research will study about the theory of probability, calculate distribution probability of character order two of text in indonesian language, coding based on Huffman theory, and design software for data compression.

Key Words: sandi, kompresi, penyandian, huffman, lossy, lossles.

Dalam dunia modern sekarang ini kebanyakan aktivitas manusia selalu berhubungan dengan dokumentasi atau data, tidak terkecuali dunia industri dan dunia pendidikan. Data-data yang ada harus tersimpan dengan rapi dan dapat digunakan setiap saat apabila dibutuhkan. Biasanya data-data tersebut tidak hanya digunakan sendiri, tetapi juga dibutuhkan pihak lain, untuk itu perlu adanya suatu sistem penyimpanan data dan pertukaran data atau komunikasi data yang baik agar dapat menunjang aktivatas manusia. Media komunikasi data dan penyimpanan data yang saat ini sedang berkembang adalah media komunikasi data dan media penyimpanan data elektronik. Kapasitas media komunikasi data dan penyimpanan data elektronik yang digunakan saat ini bukanlah tak terbatas, dan membutuhkan biaya yang cukup mahal. Salah satu upaya yang dapat dilakukan adalah dengan melakukan kompresi terhadap data yang akan disimpan atau sebelum dikirim, dengan demikian dapat menghemat penggunaan media penyimpanan data serta media komunikasi data elektronik.

Tujuan yang akan dicapai dalam penelitian ini adalah untuk implementasi teori pengkodean menjadi suatu perangkat lunak kompresi data. Untuk memperoleh hasil yang optimal ada beberapa hal yang perlu diperhatikan, yaitu tipe file yang dapat dikompres adalah hanya file yang ber-*extension* **.**txt dan karakter yang digunakan adalah yang ada pada papan ketik dan tidak mengenal karakter Tab serta ukuran data maksimum adalah 32 KB.

Teori probabilitas mempelajari rerata gejala massa yang terjadi secara berurutan atau bersama-sama, seperti pancaran elektron, hubungan telepon, deteksi radar, pengendalian kualitas, kegagalan sistem, permainan untung-untungan, mekanika statistik, turbulen, gangguan, laju kelahiran dan kematian serta teori antrian.

$$P(A) = \frac{n_A}{n}$$

catatan : n harus cukup besar

Tafsiran ini tidak tepat, perkataan “dengan kepastian derajat tinggi”, “dekat”, dan “cukup besar” tidak mempunyai arti yang jelas. Meskipun demikian, kekurangtepatan di atas tidak dapat dihindari. Bila mencoba mendefinisikan perkataan “dengan kepastian

Yuli Christyono, (yuli@elektro.ft.undip.ac.id), adalah dosen di Jurusan Teknik Elektro Fakultas Teknik Universitas Diponegoro (Undip) Semarang Jl. Prof. Sudharto, S.H. Tembalang Semarang 50275

tinggi" dalam bentuk probabilitas hanya akan menunda kesimpulan yang tidak dapat dihindarkan bahwa probabilitas, seperti teori fisis yang lain, berhubungan dengan teori fisis hanya dalam bentuk tak eksak (*inexact*). Meskipun demikian, teori probabilitas adalah disiplin eksak yang berkembang secara logis dari aksioma yang didefinisikan secara jelas dan berlaku bila diterapkan pada persoalan nyata.

Model Distribusi Statistik Karakter

Misalkan suatu perpustakaan memiliki banyak buku yang harus dipilih, katakanlah 100 juta buku yang sangat tebal dan tiap buku memiliki 100 juta karakter atau huruf didalamnya. Saat mulai masuk ke perpustakaan itu, memilih dengan cara acak lalu keluar dengan membawa buku yang dipilih.

Secara matematis, buku yang dipilih dinyatakan :

$$X = (X_1, X_2, X_3, X_4, ...)$$

Dengan X mewakili semua buku, X_1 mewakili karakter pertama dalam buku, X_2 mewakili karakter kedua dalam buku, dan seterusnya. Meskipun dalam kenyataannya, panjang buku terbatas, namun secara matematis panjangnya dianggap tak terbatas dengan pertimbangan bahwa buku sangat lama untuk dapat dibayangkan dan berlansung terus menerus. Lebih lanjut secara matematis akan mejadi lebih sederhana dan menarik bila panjang buku dianggap terbatas. Singkatnya, bahwa bila semua karakter dalam buku-buku yang dipilh tersebut dianggap berupa karakter *lower-case* ('a' samapi 'z') atau SPACE. Sumber alphabet A menetapkan pengaturan dari semua nilai yang mungkin dari 26 karakter :

A = {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}

Sekarang bagaimana teknik merancang algoritma kompresi buku yang dipilih harus ditentukan sendiri, karena tidak ada orang yang tahu buku mana yang akan dipilih selanjutnya. Akan tetapi semua tahu bahwa buku yang dipilih berasal dari perpustakaan. Dari perspektif ini, karakter dalam buku $(X_1, i = 1,2,...)$ adalah variabel acak yang nilainya diambil dari alphabet $\boldsymbol{A}$. Seluruh buku, X adalah hanya urutan tak terbatas dari beberapa variabel acak. Jadi X adalah proses acak. Beberapa cara dapat digunakan untuk mengatur model sifat statistik dari buku :

a. *Zero-Order-Model*
   Setiap karakter secara statistik berdiri sendiri dari semua karakter yang lain dan 26 nilai mungkin sama dengan yang terdapat dalam alfabet $\boldsymbol{A}$.
b. *First-Order-Model*
   Dalam bahasa inggris, beberapa huruf terjadi lebih banyak pengulangan dibandingkan dengan huruf yang lain. Sebagai contoh, pada huruf 'a'dan 'e' lebih umum dibanding huruf 'q' dan 'z'. Jadi dalam model ini, karakter masih berdiri sendiri dari yang lain, tetapi distribusi probabilitas dari beberapa karakter adalah menurut *first-order statistical of english text*
c. *Second-Order-Model*
   Dua model sebelumnya diasumsikankan bebas secara statistik dari satu karakter berikutnya sebagai contoh, beberapa kali#at i#i hurufnya kehil#ngan, kan tetapi masih dapat dipahami tulisan apa yang dimaksuddengan melihat konteksnya. Ini menyatakan secara tidak langsung bahwa ada beberapa ketergantungan diantara beberapa karakter. Secara alamiah, karakter yang dekat dengannya akan lebih bergantng dari pada karakter yang berada jauh dari yang lainnya. Dalam model ini, karakter sekarang yakni X_i berubah sesuai dengan karakter X_{i-1} sebelumnya. Sebagai contoh huruf 'u' jarang terjadi (probabilitas = 0,022). Akan tetapi, dengan adanya karakter sebelumnya yaknik 'q', probabilitas dari 'u' dalam karakter sekarang akan lebih besar (probabilitas = 0,995).
d. *Third-Order-Model*
   Ini adalah lanjutan model sebelumnya. Disini, karakter sekarang yakni X_i bergantung pada dua karakter sebelumnya : $(X_i, X_2, .. X_{i-3})$, tetapi secara kondisional berdiri sendiri dari semua karakter sebelumnya. Dalam model ini, distribusi karakter dari X_i berubah menurut (X_{i-2}, X_{i-1}).
e. *General-Model*
   Dalam model ini, buku X berubah-ubah secara acak dan stasioner. Sifat statistik dari model ini terlalu rumit bila diaplikasikan. Model ini hanya menarik dari titik pandang teori.

Huffman *Encoding*

Metode kompresi data yang di kembangkan oleh D.A. Huffman adalah metode kompresi data berdasarkan probabilitas dari masing-masing karakter dalam suatu data. Karakter-karakter

dalam data akan disandikan ulang berdasarkan probabilitasnya. Karakter yang mempunyai probabilitas paling besar akan disandikan dengan kode sandi yang pendek, dan karakter dengan probabilitas paling kecil akan disandikan dengan kode yang panjang. Metode penyandian ini yang dikenal dengan nama pohon Huffman. Untuk mendapatkan sandi huffman ada beberapa langkah yang harus ditempuh yaitu :

1. Susun simbol sumber dalam urutan probabilitas menurun (yang paling besar di atas dan yang paling kecil di bawah).
2. Gabungkan 2 simbol paling bawah (paling kecil), beri label "0" dan "1" pada kedua cabang, label "1" untuk yang lebih kecil dan label "0" untuk yang lebih besar, dan jumlahkan probabilitasnya.
3. Perlakukan probabilitas hasil jumlah tadi sebagai probabilitas baru untuk simbol baru.
4. Ulangi langkah ke-2, teruskan sampai selesai (nilainya harus sama dengan satu jika dijumlahkan keseluruhannya).
5. Untuk mencari kata sandi setiap simbol, catat label "0" dan "1" pada langkah ke-2, dan ikuti cabang dari simpul terakhir kembali ke simpul awal.

Teori penyandian Huffman akan mengasilkan panjang rata-rata sandi yang lebih kecil, sehingga akan menghasilkan kompresi data yang lebih baik.

METODE

Metode kompresi yang digunakan dalam adalah metode kompresi data penyandian Huffman berdasarkan probabilitas kemunculan karakter dalam teks Bahasa Indonesia orde dua.

Pengertian **orde dua** dalam perancangan ini adalah dua karakter, dengan demikian pengertian **probabilitas kemunculan karakter dalam teks Bahasa Indonesia orde dua** adalah probabilitas kemunculan dua karakter berurutan dalam teks Bahasa Indonesia, misalnya "aa", "ab", "ac", sampai "az", dan seterusnya.

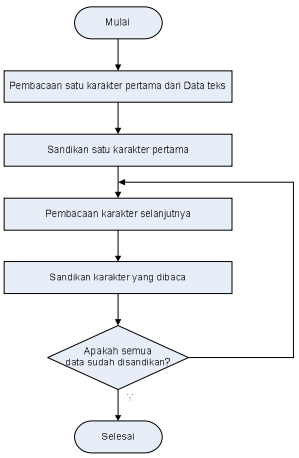
Perhitungan Probabilitas kemunculan karakter dalam teks Bahasa Indonesia orde dua dilakukan dengan menggunakan perangkat lunak.

Perhitungan probabilitas yang dilakukan adalah perhitungan probabilitas untuk 2 tiap dua karakter yang berurutan, misalnya "aa", "ab", "ac" samapi "az" dan seterusnya di dalam teks.

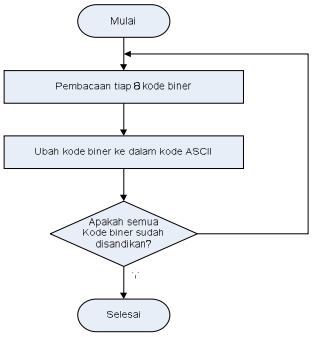
Penentuan kode penyandian dibuat berdasarkan probabilitas yang telah dihitung sebelumnya, sesuai dengan teori pengkodean Huffman yang dikenal dengan istilah Pohon Huffman.

Bedasarkan perhitungan probabilitas dilakukan terhadap semua pasangan karakter yang mengandung karakter "a" seperti "aa", "ab","ac", dan seterusnya.

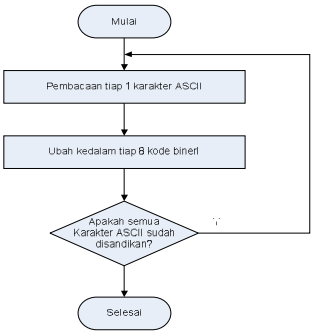
Untuk lebih mempermudah perancangan perangkat lunak dibuat diagram alir seperti gambar di bawah ini.



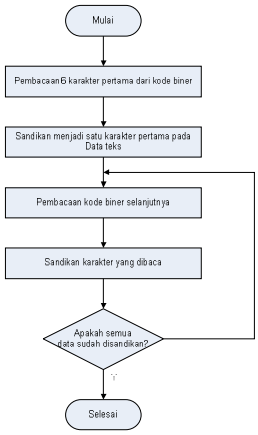
Gambar 1 Diagram Alir Penyandian I



Gambar 2 Diagram Alir Penyandian II



Gambar 3 Diagram Alir Pengawasandian I



Gambar 4 Diagram Alir Pengawasandian II

PEMBAHASAN

Pengukuran Lama Eksekusi

a. Lama eksekusi penyandian I
   Waktu yang dibutuhkan untuk mengeksekusi perintah penyandian I adalah 1 detik dengan panjang data adalah 1690 byte.
   Dengan demikian dapat dihutung panjang rata-rata data yang dapat dieksekusi dalam waktu 1 detik adalah :

$$\bar{r} = \frac{1690}{1} = 1690 \text{ byte}$$

b. Lama eksekusi penyandian II
   Waktu yang dibutuhkan untuk mengeksekusi perintah penyandian II adalah 0 detik (kurang dari 1 detik sehingga diasumsikan = 0.5 detik) dengan panjang data adalah 5657 bit.
   Dengan demikian dapat dihutung panjang rata-rata data yang dapat dieksekusi dalam waktu 1 detik adalah:

$$\bar{r} = \frac{5657}{0.5} = 11314 \text{ bit}$$

c. Lama eksekusi pengawasandian I
   Waktu yang dibutuhkan untuk mengeksekusi perintah pengawasandian I adalah 5 detik dengan panjang data adalah 732 byte.
   Dengan demikian dapat dihutung panjang rata-rata data yang dapat dieksekusi dalam waktu 1 detik adalah :

$$\bar{r} = \frac{732}{5} = 146.4 \text{ byte}$$

d. Lama eksekusi pengwasandian II
   Waktu yang dibutuhkan untuk mengeksekusi perintah pengawasandian II adalah 3 detik dengan panjang data adalah 5657 bit.
   Dengan demikian dapat dihutung panjang rata-rata data yang dapat dieksekusi dalam waktu 1 detik adalah :

$$\bar{r} = \frac{5657}{3} = 1885.66 \text{ bit}$$

Faktor Kompresi

Dari hasil kompresi beberapa file data didapatkan hasil sebagai berikut :

Tabel 1 Hasil kompresi beberapa file data

No	Nama File	Kapasitas		Faktor kompresi
		Sebelum	Sesudah	
1	Sample 1	5,18 KB	2,40 KB	54, 910 %
2	Sample 2	2,50 KB	1, 17 KB	55,365 %
3	Sample 3	3,84 KB	1,82 KB	55,089 %
4	Sample 4	3,95 KB	1,81 KB	56,427 %
5	Sample 5	2,84 KB	1,24 KB	57,216 %

Dari Tabel di atas dapat dijelaskan bahwa ukuran kapasitas file tidak berpengaruh pada faktor kompresi, tetapi yang berpengaruh pada faktor kompresi adalah karakteristik data teks dalam file. Apabila probabilitas kemunculan karakter orde dua dalam data teks pada file mendekati angka-angka probabilitas yang digunakan dalam tabel kompresi, maka faktor kompresi akan menjadi lebih besar, tetapi apabila probabilitas kemunculan karakter orde dua dalam data teks pada file jauh dari angka-angka probabilitas yang digunakan dalam tabel kompresi, maka faktor kompresi akan menjadi lebih kecil.

Perbandingan Hasil Kompresi dengan Perangkat Lunak Lain

Dari hasil kompresi beberapa file data dengan perangkat lunak hasil perancangan dan dengan perangkat lunak kompresi lain dalam hal ini adalah WinRAR, didapatkan hasil sebagai berikut:

Tabel 2 Perbandingan Hasil Kompresi

Nama File	Ukuran Sebelum	Ukuran Sesudah		Selisih (KB)
		Orde 2	Win RAR	
Sample 1	5,18 KB	2,40 KB	2,15 KB	W(0,25)
Sample 2	2,50 KB	1, 17 KB	1,10 KB	W(0,07)
Sample 3	3,84 KB	1,82 KB	1,64 KB	W(0,18)
Sample 4	3,95 KB	1,81 KB	1,70 KB	W(0,09)
Sample 5	2,84 KB	1,24 KB	1,34 KB	TA(0,1)

Dari Tabel di atas terlihat bahwa untuk file sample 1, sample 2, sample 3 dan sample 4, hasil kompresi dengan WinRAR menghasilkan ukuran

file yang lebih kecil, meskipun selisihnya cukup kecil. Sedangakn untuk file sample 5, hasil kompresi dengan perangkat lunak hasil perancangan menghasilkan ukuran file yang lebih kecil dengan selisih 0,1 KB. Dari tabel 2 di atas dapat juga dijelaskan bahwa, ukuran kapasitas file tidak berpengaruh pada faktor kompresi, tetapi yang berpengaruh pada faktor kompresi adalah karakteristik data teks dalam file.

KESIMPULAN

Dari hasil pengujian dan analisis yang telah dilakukan, maka dapat diambil kesimpulan *pertama* : kompresi data dapat menghasilkan faktor kompresi yang lebih baik apabila, karakteristik probabilitas data teks mendekati karakterisktik probabilitas dari total sampel yang digunakan untuk menghitung sandi Huffman, yang nantinya digunakan pada tabel sandi. *Kedua* terdapat perbedaan pada data hasil pengawasandian dengan data asli yaitu, huruf kapital (*upper case*) pada data asli, berubah menjadi huruf kecil (*lower case*) pada data hasil pengawasandian. *Ketiga* waktu eksekusi penyandian II (0.5 detik) lebih pendek dari pada waktu eksekusi penyandian I (1 detik). Hal ini terjadi karena pada proses penyandian I terjadi perulangan penambahan karakter biner pada teks 2 yang semakin lama- semakin panjang sehingga proses penyandian I menjadi lebih lama, berbeda dengan pada proses penyandian II yang hanya melakukan pencuplikan tiap 8 bit dan menyandikan kedalam kode ASCII. *Keempat* waktu eksekusi Pengawasandian II (3 detik) lebih pendek dari pada waktu eksekusi pengawasandian I (5 detik). Hal ini terjadi karena pada proses pengawasandian I terjadi perulangan penambahan karakter biner pada teks 2 yang semakin lama- semakin panjang sehingga proses pengawasandian I menjadi lebih lama, berbeda dengan pada proses pengawasandian II yang hanya melakukan pencuplikan tiap 1 bit dan mencocokan dengan tabel sandi dan melakukan pengawasandian terhadap data.

SARAN

Setelah penelitian ini ada beberapa topik yang dapat dijadikan penelitian selanjutnya, yaitu *pertama* : dapat dikembangkan kompres data yang tidak hanya mengenal huruf kecil (*lower case*) tetapi juga mengenal huruf besar (*upper case*), sehingga tidak ada data yang hilang saat dilakukan kompresi data. *Kedua* sistem ini dapat dikembangkan untuk distribusi probabilitas kemunculan karakter orde tiga atau lebih untuk mendapatkan faktor kompresi yang jauh lebih baik.

DAFTAR RUJUKAN

Associate Profesor. 2003, *Information Theory, Coding and Cryptography, International Edition, Department of Electrical Engineering, Indian Insitute Technology*, Delhi.

Budiono; Wayan, Koster. Januari 2001, *Teori dan Aplikasi Statistika dan Probabilitas*, PT. Remaja Rosdakarya, Bandung.

Dodd, Annabel Z. 2002, *The Essential Guide to Telecommunications*, Andi Yogyakarta, Yogyakarta.