

PEMBELAJARAN KOMPRESI TEX DENGAN MENGGUNAKAN METODE SHANON-FANO

Herry Sulendro Mangiri
Universitas Mataram
ch2mangiri@gmail.com

Disetujui: April 2018. Dipublikasikan: April 2018

ABSTRAK

Kompresi merupakan proses perubahan sekumpulan data menjadi suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data. Saat ini terdapat berbagai tipe algoritma kompresi. Kompresi merupakan proses perubahan sekumpulan data menjadi suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data. Saat ini terdapat berbagai tipe algoritma kompresi, antara lain: Huffman, LIFO, LZHUF, LZ77 dan variannya (LZ78, LZW, GZIP), Dynamic Markov Compression (DMC), Block-Sorting Lossless, Run-Length, Shannon-Fano, Arithmetic, PPM (Prediction by Partial Matching), Burrows-Wheeler Block Sorting, dan Half Byte. Ada beberapa faktor yang sering menjadi pertimbangan dalam memilih suatu metode kompresi yang tepat, yaitu kecepatan kompresi, sumber daya yang dibutuhkan (memori, kecepatan PC), ukuran file hasil kompresi, besarnya redundansi, dan kompleksitas algoritma. Metode Shanon-fano yang di pergunakan dalam penulisan ini mengkompresi input stream dengan menggunakan sampel data teks yang berupa tulisan nama. Data teks awal dengan jumlah 176 bits, dengan menggunakan metode ini terdapat beberapa langkah yang di lakukan sehingga pada saat di lakukan kompresi akan menjadi berkurang jumlahnya menjadi 107 bits. Manfaat kompresi ini sama dengan jenis kompresi yang lain yaitu memudahkan dalam menyimpan file dan menghemat memory yang terdapat dalam komputer, sehingga memudahkan komputer dalam bekerja dan tidak membebani dalam pemrosesan data.

Kata Kunci : Kompresi, Metode Shanon-Fano, teks

ABSTRACT

Compression is the process of converting a set of data into a form of code to save the storage needs and time for data transmission. Currently there are various types of compression algorithms. Compression is the process of converting a set of data into a form of code to save the storage needs and time for data transmission. Currently there are various types of compression algorithms, including: Huffman, LIFO, LZHUF, LZ77 and its variants (LZ78, LZW, GZIP), Dynamic Markov Compression (DMC), Block-Sorting Lossless, Run-Length, Shannon-Fano, Arithmetic, PPM (Prediction by Partial Matching), Burrows-Wheeler Block Sorting, and Half Byte. There are several factors that are often considered in choosing an appropriate compression method, ie compression speed, required resources (memory, PC speed), compression file size, redundancy size, and algorithm complexity. The Shanon-fano method used in this paper compresses the input stream by using text data samples in the form of name tags. Initial text data with the number of 176 bits, using this method there are several steps that are done so that at the time of doing the compression will be reduced in number to 107 bits. Benefits of compression is the same with other types of compression is easier to save files and save memory contained in the computer, making it easier for computers in the work and not burdensome in data processing.

Kata kunci: learning mediacalculus, mastery learning, learning independence.

PENDAHULUAN

Sistem komunikasi dirancang untuk mentransmisikan informasi yang dibangkitkan oleh sumber ke beberapa tujuan. Sumber informasi mempunyai beberapa bentuk yang berbeda. Sebagai contoh, dalam radio broadcasting, sumber biasanya sebuah sumber audio (suara atau musik). Dalam TV broadcasting, sumber informasi biasanya sebuah sumber video yang keluarannya berupa citra bergerak. Keluaran dari sumber-sumber ini adalah sinyal analog dan sumbernya disebut sumber analog. Kontras dengan komputer dan tempat penyimpanan data (storage), seperti disk magnetic atau optical, menghasilkan keluaran berupa sinyal diskrit (biasanya karakter binary atau ASCII (American Standard Code for Information Interchange)) dan tentu saja sumbernya disebut sumber diskrit. Baik sumber analog maupun diskrit, sebuah komunikasi digital dirancang untuk mentransmisikan informasi dalam bentuk digital. Sehingga konsekuensinya keluaran dari sumber harus diubah dahulu menjadi bentuk keluaran sumber digital yang biasanya dilakukan pada source encoder, yang keluarannya dapat diasumsikan sebuah digit biner sekuensial. (Rahmad Fauzi, 2003)

Representasi Kode ASCII menjadi kode standar dalam dunia komputer karena kemudahan perpecahan sebuah pesan besar menjadi huruf-huruf dimana kode ASCII mempunyai panjang 8 bit untuk mengkodekan 256 karakter atau huruf. Namun, kode ASCII ini bisa kurang efisien untuk aplikasi tertentu. Misalnya dalam teks bahasa Indonesia, huruf 'e' (ASCII(e) = 101) sering dipakai. Sebaliknya, huruf Umlaut "Ü" (ASCII = 252) jarang sekali, bahkan tidak pernah

dipakai. Oleh karena itu, kurang bagus bila kode ASCII mengkodekan kedua huruf tersebut sama dengan 8 bit. Hal inilah yang menjadi salah satu ide dasar untuk mengompres teks dengan menggantikan kode ASCII yang dipakai ke kode baru yang lebih baik. (Mueller, 2003)

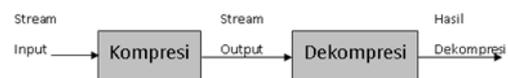
Kompresi merupakan proses perubahan sekumpulan data menjadi suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data. Saat ini terdapat berbagai tipe algoritma kompresi, antara lain: Huffman, LIFO, LZHUF, LZ77 dan variannya (LZ78, LZW, GZIP), Dynamic Markov Compression (DMC), Block-Sorting Lossless, Run-Length, Shannon-Fano, Arithmetic, PPM (Prediction by Partial Matching), Burrows-Wheeler Block Sorting, dan Half Byte.

Masih banyak lagi metode dan algoritma kompresi, namun dalam penulisan ini akan dibahas hanya mengenai metode kompresi dengan menggunakan metode Shanon-Fano.

METODE PENELITIAN

1. Kompresi Data

Kompresi data (data compression) adalah proses mengubah stream data input menjadi stream data output agar menjadi lebih kecil. Proses pembalikan data yang sudah terkompres menjadi data semula disebut dengan dekompresi (decompression). Gambar 1 berikut adalah proses kompresi dan dekompresi.



Gambar 1. Proses Kompresi dan Dekompresi

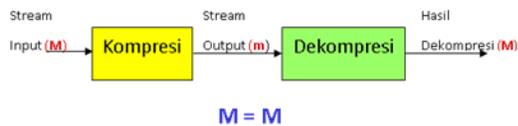
Kompresi data membutuhkan pengidentifikasi dan pengkutifan

terhadap terjadinya redundansi sumber-sumber informasi. Dalam kata lain, kompresi data mencoba mengurangi jumlah bits yang digunakan untuk menyimpan atau mengirimkan (mentransmisikan) informasi. Ada beberapa metode kompresi data yang dapat digunakan, dimana satu sama lain berbeda dalam penggunaannya yang tergantung pada bentuk data apa yang harus dikompresikan. Jenis data dengan metode kompresi yang dipilih akan menghasilkan perbedaan kapasitas atau kualitas hasil kompresi yang diperoleh.

Metode kompresi data dibedakan ke dalam dua tipe utama, yaitu kompresi metode Lossless (Lossless Compression) dan metode Lossy (Lossy Compression).

a. Lossless Compression

Melalui metode ini, stream input informasi yang dikompresi akan mempunyai kapasitas yang sama dengan stream output informasi setelah melalui proses dekomposisi ($M = M$) sebagaimana ditunjukkan pada Gambar 1.2 di bawah. Metode ini lebih cocok untuk mengkompresi data berbentuk databases records, spreadsheets, word processing files atau bentuk teks-teks lainnya

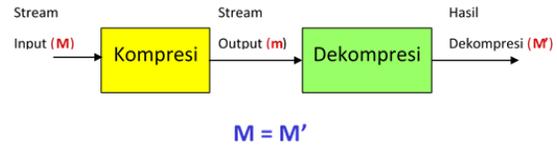


Gambar 2. Blok Diagram Metode Lossless Compression

b. Lossy Compression

Melalui metode ini, stream input informasi yang dikompresi akan mempunyai kapasitas yang hampir sama dengan stream output informasi setelah melalui proses dekomposisi

($M \neq M'$) sebagaimana ditunjukkan pada Gambar 3 di bawah.



Gambar 3. Blok Diagram Metode Lossy Compression

Metode ini lebih efektif untuk mengkompresi data berbentuk grafik, gambar, suara.

Jumlah hasil kompresi yang dapat dicapai melalui penggunaan kedua algoritma ini sangat tergantung terhadap jumlah redundansi dan ekstraksi dari sumber informasi tersebut.

2. Tuntutan Melakukan Kompresi Data

Sistem penyimpanan dokumen secara tradisional dalam bentuk tertulis maupun dalam lemari penyimpanan surat menyurat dirasa sangat tidak efisien. Hal ini dialami terutama ketika kita menginginkan kembali untuk mengambil atau menemukan informasi-informasi terdahulu yang kita sudah lupa pada bagian mana informasi tersebut disimpan.

Untuk mengatasi permasalahan tersebut, maka sistem penyimpanan dokumen sudah digantikan melalui sistem penyimpanan dan akses dokumen secara elektronik dalam sistem komputer. Komputer telah membantu kita untuk mengorganisasikan hal-hal berkaitan dengan dokumentasi secara efisien dan efektif sehingga informasi dapat ditempatkan yang mudah untuk diakses kembali.

Dalam kaitan dengan sistem penyimpanan, kapasitas dari suatu alat penyimpanan (memori komputer) dapat secara efektif ditingkatkan dengan metode kompresi.

Selain ini metode ini bermanfaat juga dalam melakukan komunikasi data guna lebih mempercepat proses pengiriman yang lebih efektif dan efisien. Dalam komunikasi data, bandwidth sebuah jaringan komunikasi digital menjadi lebih efektif melalui sistem kompresi data dalam proses pengiriman akhir dan sistem dekompresi dalam penerimaan akhir sebuah informasi.

Dalam grafika computer, kita tertarik untuk mengurangi ukuran per blok data grafik sehingga akan lebih mengefisienkan tempat penyimpanan dalam memori computer secara fisik. Manfaat lain dari teknik kompresi yaitu untuk mempermudah dan mempercepat waktu download, upload atau proses penyimpanan dan pemindahan dokumen dalam internet atau dalam sistem yang berbeda.

3. Redundansi

Ada 2 istilah yang sering digunakan untuk menilai kualitas sebuah algoritma kompresi :

(a) Faktor Kompresi (Compression Factor)

Yaitu nilai yang bisa dihitung sebagai ukuran stream input dibagi ukuran stream output.

(b) Rasio Kompresi (Compression Ratio)

Yaitu hasil pembagian ukuran stream output dengan ukuran stream input

Istilah penting lain dalam teori informasi adalah redundansi (redundancy) sebuah pesan. Redundansi menunjukkan tingkat optimalitas suatu metode. Makin kecil nilainya, makin optimal pengkodean metode tersebut. Redundansi R bisa dihitung dengan membandingkan entropi pesan dengan entropi maksimal, yaitu :

$$R = \log_2(n) + \sum_{i=1}^n p_i \log_2(p_i) \quad (1)$$

Redundansi terjadi ketika representasi dari sebuah media yang berkapasitas x bytes dan

y bytes ($y < x$), maka x merupakan suatu redundansi relatif terhadap y. Dalam bentuk lain, redundansi terjadi jika representasi dari isi capture suatu media tidak dapat dipahami oleh manusia kemudian dihapus, tetapi tidak akan mempengaruhi isi media aslinya.

Contoh: Penangkapan frekuensi audio di luar jangkauan pendengaran manusia dapat dihindari tanpa banyak mempengaruhi kualitas isinya.

4. Entropi

Claude Shannon pada tahun 1940-an menemukan teori baru yakni rumus yang menghubungkan antara informasi dan probabilitas. Shannon mengatakan bahwa informasi mempunyai hubungan dengan "tingkat kejutan" atau adanya informasi baru dalam sebuah pesan. Misalnya ada stream input yang berisi 1000 huruf 'a' tetapi satu huruf 'b'. Jadi kemungkinan 'a' untuk muncul jauh lebih besar daripada 'b' atau 'c'. Kalau dibaca huruf berikut dari input dan jika 'a' muncul, maka ini tidak mengherankan karena probabilitas 'a' memang besar, dengan kata lain informasi yang disampaikan tidaklah istimewa. Tetapi jika yang muncul adalah 'b' maka ini akan menjadi perhatian karena probabilitas huruf 'b' cukup kecil. Makin besar probabilitas sebuah huruf dalam stream input, makin kecil tingkat informasi huruf tersebut. Sebaliknya, pesan yang mengejutkan pendengar memberikan banyak informasi baru.

Misalkan probabilitas ai disebut pi, maka Entropi (Entropy) dari sebuah huruf ai didefinisikan Shannon sebagai $-p \cdot \log_2(p_i)$. Entropi sebuah huruf bisa digeneralisasi menjadi entropi E untuk seluruh pesan dengan rumus berikut :

$$E = -\sum_{i=1}^n p_i \cdot \log_2(p_i) = \sum_{i=1}^n p_i \cdot \log_2\left(\frac{1}{p_i}\right)$$

(2)

Shannon membuktikan bahwa entropi sebuah pesan mempunyai arti khusus untuk kompresi. Entropi E adalah sama dengan jumlah informasi rata-rata dalam setiap huruf. Kalau sebuah pesan dikodekan, setiap huruf memerlukan rata-rata minimum E bit. Perlu diperhatikan bahwa E tidak pasti bilangan bulat, sehingga ada perbedaan antara teori dan implementasi (dalam sebuah komputer tidak ada kemungkinan untuk memakai bit string dengan jumlah bit yang bukan bulat). Dengan demikian, pengkodean optimal untuk seluruh pesan memerlukan minimal s. E bit, dimana s sama dengan jumlah total huruf dalam pesan. Teori informasi bisa membuktikan bahwa tidak mungkin sebuah pesan dikompres sampai menjadi lebih kecil batasan minimal ini.

Entropi menyatakan suatu ukuran untuk menentukan berapa banyak informasi dikodekan menjadi sebuah pesan. Entropi adalah suatu ukuran ketidakpastian dalam sebuah pesan. Informasi dan ketidakpastian merupakan suatu konsep yang sepadan. Satuan entropi dinyatakan dalam bits per symbol. Satuan ini ditentukan oleh dasar algoritma, bahwa: 2 : binary (bit) dan 10 : desimal (digit).

HASIL DAN PEMBAHASAN

Pada hasil penelitian ini, penulis melakukan kompresi teks dengan menggunakan nama penulis sendiri untuk di aplikasikan dengan metode Shannon-Fano. Metode Shannon-Fano menemukan teknik untuk melakukan kompresi data yang disebut metode (algoritma) Shannon-Fano (1948). Di zamannya, kompresi data menggunakan metode ini sangat bagus terutama untuk kompresi-kompresi data berbentuk teks.

Sebelum menampilkan hasil dari penelitian ini, terlebih dahulu peneliti akan membahas tentang beberapa hal yang mendukung dalam penelitian ini. Metode Shannon-Fano adalah metode untuk menggantikan setiap simbol dengan sebuah alternatif kode biner yang panjangnya ditentukan probabilitas dari simbol tersebut dalam proses kompresinya.

a. Teorema Shannon Lossless Coding

Teorema Shannon Lossless Coding berdasar pada konsep kode blok. Misalkan diberikan input stream informasi berupa karakter sebagai berikut:

S = 'aa bbb cccc dddd eeeee fffffffggggggg'

Sumber informasi tersebut mempunyai panjang sama dengan 40 simbol termasuk karakter spasi atau setara dengan $8 \times 40 = 320$ dalam kode ASCII, sehingga sama panjangnya dengan kode seperti table 1. di bawah ini.

Tabel 1. Contoh Penerapan Teorema Shannon Lossless Coding

| Karakter | Kode Bits (3 bits) | Karakter | Kode Bits (3 bits) |
|----------|--------------------|----------|--------------------|
| A | 000 | E | 100 |
| B | 001 | F | 101 |
| C | 010 | G | 110 |
| D | 011 | Space | 101 |

Tabel di atas menunjukkan bahwa setelah melalui kompresi, sumber informasi berupa karakter tersebut menjadi berjumlah $3 \text{ bits} \times 40 = 120 \text{ bits}$.

b. Algoritma Shannon-Fano

Algoritma Shannon-Fano termasuk algoritma statistik yang menghitung kode bebas prefiks. Algoritma ini dibagi menjadi dua yaitu :

1. Algoritma Shannon-Fano Tak Adaptif
2. Algoritma Shannon-Fano Adaptif

Algoritma Sannon-Fano termasuk algoritma statistik karena menggunakan data statistik dari kemunculan huruf dalam suatu file.

a). Algoritma Shannon-Fano Statik (Tak Adaptif)

Kompresi dengan algoritma ini dilakukan dengan langkah-langkah sebagai berikut :

1. Membuat tabel statistik untuk kemunculan huruf.
2. Mengurutkan data pada tabel dengan
 - a. Meletakkan huruf dengan kemunculan tertinggi pada posisi paling atas, kemudian diikuti yang lebih kecil sampai selesai.
 - b. Jika terdapat statistik kemunculan yang sama maka dilakukan pengurutan sesuai tabel ASCII.
3. Menghitung kode masing-masing huruf dengan memecah tabel menjadi dua bagian dengan pembagian masing-masing subtabel mempunyai kemunculan yang sama, atau mendekati.
4. Jika tidak dapat dibagi dalam jumlah kemunculan yang sama maka subtabel yang atas diberi kemunculan yang lebih kecil daripada subtabel dibawahnya.
5. Memberikan kode 0 untuk subtabel sebelah atas dan kode 1 untuk subtabel sebelah bawah.
6. Melakukan langkah 3 dan langkah 4 dan langkah 5 untuk masing-masing subtabel sampai subtabel-subtabel tersebut tidak dapat dibagi lagi dalam subtabel yang lebih kecil.

b). Algoritma Shannon-Fano Dinamik (Adaptif)

Mengkompresi input stream menggunakan metode Shannon-Fano Dinamik (Adaptif) dapat dilakukan dengan tahapan sebagai berikut:

- 1) Menggunakan tabel untuk menyimpan statistik dari huruf pertama sampai posisi aktual

- 2) Metode ini menggabungkan perhitungan statistik dan kode dalam pengkodean pesan.
- 3) Pada saat pengisian tabel diurutkan berdasarkan runtun input, kemudian disusun berdasarkan kode ASCII
- 4) Setiap ada pemunculan huruf yang baru maka jumlah untuk escape bertambah satu, sedangkan jika huruf tersebut pernah muncul maka jumlah escape tidak bertambah, yang bertambah hanya pada huruf yang muncul kembali
- 5) Pada saat awal tabel dianggap masih kosong , karena itu digunakan satu huruf khusus yang disebut sebagai *escape*.

Algoritma Shanon-Fanno Adaptif (Dinamik)

Input Stream :

HERRY_SULENDRO_MANGIRI

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 0 | 0 |

Input Huruf “ H “

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 1 | 0 |
| H | 1 | 1 |

Input Huruf “ E “

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 2 | 0 |
| E | 1 | 10 |
| H | 1 | 11 |

Input Huruf “ R “

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 3 | 0 |
| E | 1 | 10 |
| H | 1 | 110 |
| R | 1 | 111 |

Input Huruf “ R “

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 4 | 0 |
| E | 1 | 10 |

| | | |
|---|---|-----|
| H | 1 | 100 |
| R | 2 | 111 |

Input Huruf “ Y “

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 5 | 0 |
| E | 1 | 100 |
| H | 1 | 101 |
| R | 2 | 110 |
| Y | 1 | 111 |

Input Spasi “ – “

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 6 | 0 |
| Spasi | 1 | 100 |
| E | 1 | 1010 |
| H | 1 | 1011 |
| R | 2 | 110 |
| Y | 1 | 111 |

Input Huruf “S “

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 7 | 0 |
| Spasi | 1 | 100 |
| E | 1 | 1010 |
| H | 1 | 1011 |
| R | 2 | 110 |
| S | 1 | 1110 |
| Y | 1 | 1111 |

Input Huruf U

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 8 | 0 |
| Spasi | 1 | 100 |
| E | 1 | 1010 |
| H | 1 | 1011 |
| R | 2 | 1100 |
| S | 1 | 1101 |
| U | 1 | 1110 |
| Y | 1 | 1111 |

Input Huruf “ L “

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 9 | 0 |
| Spasi | 1 | 1000 |

| | | |
|---|---|------|
| E | 1 | 1001 |
| H | 1 | 1010 |
| L | 1 | 1011 |
| R | 2 | 1100 |
| S | 1 | 1101 |
| U | 1 | 1110 |
| Y | 1 | 1111 |

Input Huruf “ E “

| Huruf | Jumlah | Kode |
|--------|--------|------|
| Escape | 10 | 0 |
| Spasi | 1 | 1000 |
| E | 2 | 1001 |
| H | 1 | 1010 |
| L | 1 | 1011 |
| R | 2 | 1100 |
| S | 1 | 1101 |
| U | 1 | 1110 |
| Y | 1 | 1111 |

Input Huruf “ N “

| Huruf | Jumlah | Kode |
|--------|--------|-------|
| Escape | 11 | 0 |
| Spasi | 1 | 1000 |
| E | 2 | 1001 |
| H | 1 | 1010 |
| L | 1 | 10110 |
| N | 1 | 10111 |
| R | 2 | 1100 |
| S | 1 | 1101 |
| U | 1 | 1110 |
| Y | 1 | 1111 |

Input Huruf “ D “

| Huruf | Jumlah | Kode |
|--------|--------|-------|
| Escape | 12 | 0 |
| Spasi | 1 | 10000 |
| D | 1 | 10001 |
| E | 2 | 1001 |
| H | 1 | 1010 |
| L | 1 | 1011 |
| N | 1 | 1100 |
| R | 2 | 1101 |
| S | 1 | 1110 |
| U | 1 | 11110 |
| Y | 1 | 11111 |

Input Huruf “ R “

| Huruf | Jumlah | Kode |
|--------|--------|-------|
| Escape | 13 | 0 |
| Spasi | 1 | 10000 |
| D | 1 | 10001 |
| E | 2 | 1001 |
| H | 1 | 10100 |
| L | 1 | 10101 |
| N | 1 | 1011 |
| R | 3 | 110 |
| S | 1 | 1110 |
| U | 1 | 11110 |
| Y | 1 | 11111 |

Input Huruf “ O “

| Huruf | Jumlah | Kode |
|--------|--------|-------|
| Escape | 14 | 0 |
| Spasi | 1 | 10000 |
| D | 1 | 10001 |
| E | 2 | 1001 |
| H | 1 | 1010 |
| L | 1 | 10110 |
| N | 1 | 10111 |
| O | 1 | 1100 |
| R | 3 | 1101 |
| S | 1 | 1110 |
| U | 1 | 11110 |
| Y | 1 | 11111 |

Input Spasi “ – “

| Huruf | Jumlah | Kode |
|--------|--------|-------|
| Escape | 15 | 0 |
| Spasi | 2 | 1000 |
| D | 1 | 10010 |
| E | 2 | 10011 |
| H | 1 | 1010 |
| L | 1 | 10110 |
| N | 1 | 10111 |
| O | 1 | 1100 |
| R | 3 | 1101 |
| S | 1 | 1110 |
| U | 1 | 11110 |
| Y | 1 | 11111 |

Input Huruf “ M “

| Huruf | Jumlah | Kode |
|--------|--------|-------|
| Escape | 16 | 0 |
| Spasi | 2 | 1000 |
| D | 1 | 10010 |
| E | 2 | 10011 |
| H | 1 | 1010 |
| L | 1 | 10110 |
| M | 1 | 10111 |
| N | 1 | 11000 |
| O | 1 | 11001 |
| R | 3 | 1101 |
| S | 1 | 1110 |
| U | 1 | 11110 |
| Y | 1 | 11111 |

Input Huruf “ A “

| Huruf | Jumlah | Kode |
|--------|--------|--------|
| Escape | 17 | 0 |
| Spasi | 2 | 1000 |
| A | 1 | 10010 |
| D | 1 | 10011 |
| E | 2 | 1010 |
| H | 1 | 10110 |
| L | 1 | 101110 |
| M | 1 | 10111 |
| N | 1 | 11000 |
| O | 1 | 11001 |
| R | 3 | 1101 |
| S | 1 | 1110 |
| U | 1 | 11110 |
| Y | 1 | 11111 |

Input Huruf “ N “

| Huruf | Jumlah | Kode |
|--------|--------|--------|
| Escape | 18 | 0 |
| Spasi | 2 | 1000 |
| A | 1 | 10010 |
| D | 1 | 10011 |
| E | 2 | 1010 |
| H | 1 | 10110 |
| L | 1 | 101110 |
| M | 1 | 101111 |
| N | 2 | 11000 |
| O | 1 | 11001 |
| R | 3 | 1101 |

| | | |
|---|---|-------|
| S | 1 | 1110 |
| U | 1 | 11110 |
| Y | 1 | 11111 |

Input Huruf “ G “

| Huruf | Jumlah | Kode |
|--------|--------|-------|
| Escape | 19 | 0 |
| Spasi | 2 | 10000 |
| A | 1 | 10001 |
| D | 1 | 10010 |
| E | 2 | 10011 |
| G | 1 | 10100 |
| H | 1 | 10101 |
| L | 1 | 10110 |
| M | 1 | 10111 |
| N | 2 | 11000 |
| O | 1 | 11001 |
| R | 3 | 1101 |
| S | 1 | 1110 |
| U | 1 | 11110 |
| Y | 1 | 11111 |

Input Huruf “ I “

| Huruf | Jumlah | Kode |
|--------|--------|--------|
| Escape | 20 | 0 |
| Spasi | 2 | 10000 |
| A | 1 | 10001 |
| D | 1 | 10010 |
| E | 2 | 10011 |
| G | 1 | 10100 |
| H | 1 | 10101 |
| I | 1 | 10110 |
| L | 1 | 10111 |
| M | 1 | 11000 |
| N | 2 | 11001 |
| O | 1 | 1101 |
| R | 3 | 1110 |
| S | 1 | 11110 |
| U | 1 | 111110 |
| Y | 1 | 111111 |

Input Huruf “ R “

| Huruf | Jumlah | Kode |
|--------|--------|-------|
| Escape | 21 | 0 |
| Spasi | 2 | 10000 |

| | | |
|---|---|--------|
| A | 1 | 10001 |
| D | 1 | 10010 |
| E | 2 | 10011 |
| G | 1 | 101000 |
| H | 1 | 101001 |
| I | 1 | 10101 |
| L | 1 | 10110 |
| M | 1 | 10111 |
| N | 2 | 11000 |
| O | 1 | 11001 |
| R | 4 | 1101 |
| S | 1 | 1110 |
| U | 1 | 11110 |
| Y | 1 | 11111 |

Input Huruf “ I “

| Huruf | Jumlah | Kode |
|--------|--------|--------|
| Escape | 22 | 0 |
| Spasi | 2 | 10000 |
| A | 1 | 100010 |
| D | 1 | 100011 |
| E | 2 | 10010 |
| G | 1 | 10011 |
| H | 1 | 10100 |
| I | 2 | 10101 |
| L | 1 | 10110 |
| M | 1 | 10111 |
| N | 2 | 11000 |
| O | 1 | 11001 |
| R | 4 | 1101 |
| S | 1 | 1110 |
| U | 1 | 11110 |
| Y | 1 | 11111 |

Hasil Kompresi :

Stream Input :

Jumlah Output = 107 bits

Sebelum di kompresi = 176 bits

Dari hasil kompresi menggunakan metode shanon-fano, menunjukkan jumlah bits sebelum di lakukan kompresi sejumlah 176 bits dan setelah di lakukan kompresi 107 bits.

PENUTUP

a. Kesimpulan

Setelah mengemukakan tentang teknik kompresi data menggunakan metode Shannon-Fano Statik dan Dinamik disertai contoh-contoh aplikasinya, maka dapat dituliskan beberapa kesimpulan berikut ini:

1. Untuk input stream yang sama dikerjakan menggunakan metode Shannon-Fano Statik dengan cara mengurutkan simbol berdasarkan alphabet atau berdasarkan simbol yang mempunyai frekuensi tertinggi akan menghasilkan jumlah bits output yang sama.
2. Untuk input stream yang sama dikerjakan menggunakan metode Shannon-Fano Statik dan Shannon-Fano Dinamik akan menghasilkan jumlah bits output yang berbeda.
3. Untuk input stream yang sama, jumlah bits output hasil kompresi menggunakan metode Shannon-Fano Statik lebih sedikit dibanding jumlah bits output

hasil kompresi menggunakan metode Shannon-Fano dinamik.

b. Saran

Masih banyak Metode kompresi yang dapat di pergunakan untuk mengkompresi teks, sehingga di seuaikan dengan kapasitas komputer yang di gunakan dan juga jenis teks yang akan di kompresi. Ada beberapa faktor yang sering menjadi pertimbangan dalam memilih suatu metode kompresi yang tepat, yaitu kecepatan kompresi, sumber daya yang dibutuhkan (memori, kecepatan PC), ukuran file hasil kompresi, besarnya redundansi, dan kompleksitas algoritma. Tetapi metode ini hampir tidak pernah digunakan lagi setelah kemudian muncul metode-metode baru seperti metode Huffman.

Di sini penulis mencoba memberikan wawasan tentang bagaimana melakukan kompresi teks dengan menggunakan metode Shanon-Fano, yang selama ini proses kompresi hanya di lakukan di dalam komputer tanpa mengetahui proses kompresi huruf demi huruf secara tertulis.

DAFTAR PUSTAKA

- Fauzi, R., 2003. Analisis Beberapa Teknik Coding. USU digital Library
- Halsall, F., 2001. Multimedia Communication. Pearson Education Limited, England
- Mueller, V., 2003. Kompresi Data dan Ilmu Bahasa Sandi. Program Pascasarjana Teknik Elektro Universitas Gadjah Mada. Yogyakarta.
- Practical Huffman Coding,. <http://www.compressconsult.com/huffman/>
- Vitter, J.S., 1987. Design and Analysis of Dynamic Huffman Codes. Journal of the Association for Computing Machinery. Vol. 34, No. 4,.
- Vitter, J.S., 1989. Dynamic Huffman Coding. ACM Transactions on Mathematical Software. Vol. 15, No. 2, June 1989, Pages 158-167.
- Matthew Valenti, Convolutional Codes: Pengkodean and Performance Analysis, April 21, 2003.
- Rorabaugh, C. Britton, Error Coding Cookbook : Practical C/C++ Routines and Recipes for Error Detection and Correction, McGraw-Hill, 1996