

Perbandingan Tiga Langkah Teknik-Teknik Kompresi Teks

¹Adam Puspabhuana

¹STMIK Kharisma, Jl. Pangkal Perjuangan Km.1 By Pass Karawang, Karawang

¹Program Studi Manajemen Informatika

e-mail: ¹ adambhuana@stmik-kharisma.ac.id

Abstract— Data compression is a technique to reduce the size of the data so it could save space on data storage. There are two approaches in compressing data. These approaches are *Lossy* and *Lossless*. In this paper, *Lossless* data compression approach has been applied to reduce the size of text data. There are some algorithms to compress text data such as Bit Reduction, Huffman Coding, and LZW. Previous researchers have made various attempts from these three algorithms by combining two of these algorithms to obtain the better compression ratio and better saving percentage. This paper only focus in the compression technique although the decompression technique has been made into the same program. There are six combinations of text compression techniques in three stages. Combination of LZW and bit reduction gave best compression ratio among the other combination of two stages. In addition, Combination of LZW, bit reduction and Huffman gave best compression ratio among the other combination of three stages. The combination of the three stages has a compression ratio that is better than the combination of two stages. Compared to the combination of two stages, Combination of bit reduction, Huffman and LZW improved compression ratio that is better than other combination of three stages.

Intisari— Kompresi data adalah sebuah teknik untuk mengurangi ukuran data sehingga menghemat ruang pada penyimpanan data. Terdapat dua pendekatan dalam kompresi data. Pendekatan ini adalah *Lossy* dan *Lossless*. Pada artikel ini, pendekatan kompresi data *Lossless* telah diterapkan untuk mengurangi ukuran data teks. Terdapat beberapa algoritma untuk kompresi data teks seperti *bit reduction*, *Huffman coding* dan *LZW*. Para Peneliti sebelumnya telah membuat berbagai upaya dari tiga algoritma dengan menggabungkan dua algoritma ini untuk mendapatkan rasio kompresi (*compression ratio*) yang lebih baik dan persentase penghematan (*saving percentage*) yang lebih baik.

Artikel ini hanya menganalisa teknik kompresi meskipun teknik dekompresi telah dibuat kedalam program yang sama. Terdapat 6 kombinasi teknik kompresi teks ke dalam tiga langkah kompresi. Kombinasi dari *LZW* dan *bit reduction* memberikan rasio kompresi terbaik diantara kombinasi dua langkah lainnya. Kombinasi dari *LZW*, *bit reduction* dan *Huffman* memberikan rasio kompresi terbaik diantara kombinasi tiga langkah lainnya. Kombinasi dari tiga langkah memiliki rasio kompresi yang lebih baik dari kombinasi dua langkah. Dibandingkan dengan kombinasi dua langkah, kombinasi dari *bit reduction*, *Huffman* dan *LZW* telah memperbaiki rasio kompresi dan lebih baik dari kombinasi tiga langkah lainnya.

Kata Kunci—kompresi data, kompresi data *lossy*, kompresi data *lossless*, pengurangan *bit*, kode *huffman*, *lemple zip welch*.

I. PENDAHULUAN

Sebuah teknik kompresi dan dekompresi pada berkas diperlukan untuk menyimpannya ke dalam penyimpanan data maupun untuk memindahkannya melalui jaringan. Kompresi digunakan untuk mengurangi penggunaan sumber seperti ruang data penyimpanan atau kapasitas pengiriman [3]. Untuk mengubah data dari bentuk yang lebih mudah ke dalam data yang terkompres, maka lebih baik menggunakan sebuah program kompresi. Sebaliknya, untuk mengubah kembali data terkompres ke bentuk aslinya, maka digunakan program dekompresi [9].

Penelitian ini akan membahas peningkatan baru dengan mengkombinasi 3 teknik kompresi teks dalam tiga langkah. Teknik-teknik tersebut adalah are bit reduction, Huffman, and LZW. Penelitian ini akan merumuskan masalah menjadi:

“ Akankah hasil rasio kompresi ditingkatkan antara 20% - 40 % (dari data teks berukuran kecil hingga berukuran besar, ketika 3 langkah algoritma yang berbeda diterapkan? ”. Tujuan utama dari penelitian ini adalah untuk meningkatkan rasio kompresi dengan menggabungkan 3 teknik dibandingkan penelitian sebelumnya that telag menggabungkan 2 teknik.

Dalam 5 tahun tahun terakhir, terdapat beberapa penelitian, survey dan studi perbandingan. Sebagian besar dari penelitian ini adalah melakukan perbandingan dan telah melakukan peningkatan pada beberapa algortiam kompresi teks diantaranya bit reduction, Huffman, and LZW.

II. TEORI

Pada bagian ini akan dipaparkan teori yang mendasari kompresi teks yang digunakan dalam tiga tahap teknik kompresi teks dengan tiga algoritma yang berbeda.

A. Kompresi dan Dekompresi Teks

Kompresi teks adalah proses pengkodean informasi teks menggunakan bits yang lebih sedikit. Dekompresi teks adalah proses mengembalikan data yang terkompresi ke dalam bentuk yang dapat digunakan kembali [12].

B. Tipe Kompresi Data

Terdapat 2 tipe kompresi data yang diterapkan ke dalam area-area yang berbeda. Tipe-tipe ini adalah *Lossy* compression dan *Lossless* compression. Kompresi data *Lossy* digunakan untuk kompresi berkas gambar yang bertujuan untuk komunikasi maupun penyimpanan [9]. Tipe yang lain

adalah kompresi data *Lossless*, yang digunakan untuk pengiriman dan penyimpanan teks atau berkas biner [9].

Lossy berarti data teks yang terdekompresi tidak persis sama dengan data aslinya. Ketika data yang terdekompresi di sandikan kembali, maka tidak akan sama persis dengan data asli. Data telah hilang. Algoritma-algoritma ini disebut *irreversible compression* [11].

Kompresi data *Lossless* menjalankan algoritma kompresi data untuk kompresi atau *encode* data asli menjadi data terkompres. Ketika data tersebut di dekompresi atau di *decode*, maka data tersebut sama persis dengan data aslinya[9]. Contoh kompresi data yang menggunakan algoritma *lossless* data adalah ZIP. Jenis algoritma-algoritma ini disebut *reversible compression* [11].

C. Definisi-Definisi dasar

Di seksi ini, terdapat beberapa definisi-definisi dasar yang digunakan untuk penelitian ini. Berikut adalah definisi-definisinya:

a. **Compression Size**
Compression Size adalah ukuran berkas yang baru dalam *bits* setelah proses kompresi selesai [1].

b. **Compression Ratio**
Adalah sebuah persentase dari hasil perbandingan ukuran berkas kompresi dalam *bits* dengan ukuran berkas aslinya dalam *bits* kemudian dikalikan dengan 100% [1].
Compression ratio

$$= \left(\frac{C2}{C1} \right) * 100\% \quad (1)$$

C1=Ukuran Sebelum Kompresi,
C2=Ukuran Setelah Kompresi [13].

c. **Saving Percentage**
Saving percentage = $\left(\frac{C1-C2}{C1} \right) * 100\%$ (2)

C1=Ukuran Sebelum Kompresi,
C2=Ukuran Setelah Kompresi [13].

d. **Symbol Probability**
Adalah pembagian frekuensi symbol dalam berkas yang asli dengan jumlah seluruh symbol dalam berkas yang asli [1].

e. **Entropy**
Shannon meminjam definisi *entropy* dari statistik secara fisikal, dimana *entropy* merepresentasikan *randomness* or *disorder* dalam sebuah sistem. Secara khusus, sebuah system diasumsikan. memiliki satu set kemungkinan pernyataan-pernyataan yang tersedia pada waktu tertentu yang terdapat sebuah distribusi probabilitas pada pernyataan-pernyataan tersebut [2].
Entropy is then defined as:

$$H(S) = \sum_{s \in S} p(s) \log_2 \frac{1}{p(s)} \quad (3)$$

Dimana S adalah set kemungkinan pernyataan-pernyataan dan p(s) adalah probabilitas pernyataan s ∈ S [2].

D. Teknik-Teknik Kompresi Data

a. Algoritma Bit Reduction

Algoritma ini pertama kali diimplementasikan untuk penggunaannya pada kapasitas aplikasi SMS / *Short Message Service* [3]. *Reduction* disini berarti bahwa algoritma nya mengurangi pengkodean standar 8-bits menjadi 5-bits sistem perngkodean dan kemudian dikemas ke dalam sebuah uruan *Byte* [13]. Langkah-langkahnya adalah [13]:

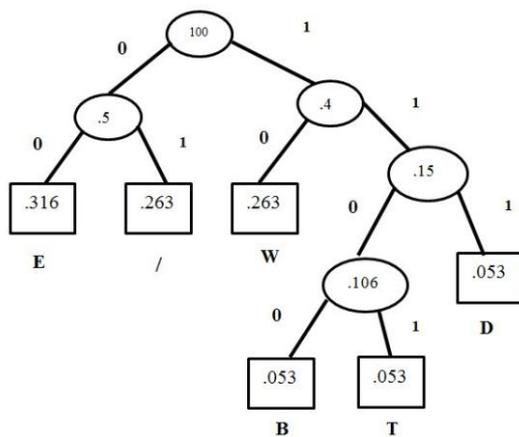
Pilih karakter-karakter yang sering muncul dari berkas teks yang asli dan convert ke dalam kode ASCII. Memperoleh bilangan biner dari kode-kode ASCII untuk setiap karakter. Tempatkan bilangan-bilangan biner ke dalam urutan *Byte* (8-bit array). Hilangkan 3 bits dari depan setiap karakter (bilangan biner). Kemudian atur kembali urutan *Byte* ini. Kemudian teks terakhir akan dikodekan dan kompresi akan dicapai. Untuk dekompresi, bekerja pada kebalikan langkah-langkah kompresi.

b. Algoritma Huffman

Ini adalah algoritma tertua. Algoritma ini dibangun oleh David A. Huffman pada tahun 1952 [1]. Algoritma ini bekerja membangun pohon biner penuh pada simbol-simbol yang berbeda yang terdapat dalam berkas yang aslinya setelah perhitungan probabilitasnya pada setiap symbol dan menempatkan dalam urutan descending [1]. Algoritma bekerja menggunakan distribusi probabilitas alphabet-alphabet dari teks asli untuk membangun *codeword* symbol-simbolnya. Distribusi probabilitas dihitung dari distribusi frekuensi semua karakter dari sumber. Kemudian *codeword* diberikan. *Codeword* terpendek diberikan untuk probabilitas paling tinggi dan *codeword* yang terpanjang diberikan untuk probabilitas paling rendah. Sebagai contoh, ada sebuah *string* “/WED/WE/WEE/WEB/WET”, kemudian akan direpresentasikan probabilitas symbol ke dalam Tabel I berikut [1]. Kemudian hasil berikut dibangun ke dalam pohon biner untuk mendapatkan *codeword* dari setiap symbol. Ditunjukkan pada Gbr 1 [1].

TABEL I
PROBABILITAS SIMBOL PADA STRING

Symbol	Probability
E	6/19 = .316
/	5/19 = .263
W	5/19 = .263
D	1/19 = .053
B	1/19 = .053
T	1/19 = .053



Gbr.1 Pohon Biner untuk String

Kemudian, *codeword* diberikan untuk setiap symbol dari pohon biner diatas, terlihat pada Tabel II [1]. Berkas terkompres untuk string diatas menjadi
 0110001110110000110000001100011000110001101 = 43 bits, selain berkas asli 19 X 8 bits = 152 bits dalam ASCII [1].

TABEL II
 CODEWORDS UNTUK SETIAP SYMBOL DALAM STRING

Symbol	Probability	Codewords
E	.316	00
/	.263	01
W	.263	10
D	.053	111
B	.053	1100
T	.053	1101

c. Algoritma LZW

Algoritma ini dibuat oleh Abraham Lempel dan Jacob Ziv pada tahun 1977. Ini dikenal dengan algoritma LZ [1]. Lalu pada tahun 1984, Terry Welch memodifikasi LZ78 *compressor* untuk implementasi dalam *controller disk* berperforma tinggi [1]. Lalu algoritman LZW dibuat dan digunakan hingga sekarang.

Pertama, sebuah kamus yang memuat *string* dengan single karakter yang berhubungan ke semua input karakter yang memungkinkan (kecuali *clear* dan *top code*). Langkah-langkah kompresi sebagai berikut:

Memindai melalui masukkan *string* hingga menemui salah satu yang tidak ada dalam kamus [6]. Ketika algoritma menemukan sebuah *string*, indeks untuk string kecil pada karakter terakhir (yaitu *substring* yang ada di dalam kamus) kode dihasilkan dari kamus dan mengirimkann kode menjadi *output* [6]. *String* baru (termasuk karakter terakhir) ditambahkan ke dalam kamus dengan kode terakhir yang tersedia [6]. Input karakter yang terakhir digunakan sebagai *string point top scan* berikutnya untuk *substring*.

Kemudian *string* yang lebih panjang berturut-turut didaftarkan dalam kamus dan membuat tersedia untuk *encoding* berikut sebagai nilai single output. Ini bekerja dalam pola berulang kemudian bagian-bagian awal akan terlihat kompresi kecil.

Contoh output untuk *string* terlihat pada Tabel III. Seperti terlihat, table *string* terisi secara cepat, sejak sebuah *string* ditambahkan ke dalam table setiap *code* dikeluarkan [6]. Terlihat pada table 2.3. Layaknya kompresi, dekompresi menambahkan sebuah *string* baru. Kedalam table setiap membaca ke dalam kode baru. Yang perlu dilakukan adalah menerjemahkan setiap kode yang datang ke dalam sebuah *string* dan mengirimkannya menjadi output. Hal penting untuk dicatat adalah table *string* berakhir sama persis dengan table yang dibangun selama kompresi. *String* output identik dengan *string output* dari algoritma kompresi. Dicatat bahwa 256 kode yang pertama didefinisikan untuk menterjemahkan ke dalam karakter *single string*, seperti dalam kode kompresi [6].

TABEL III
 PROSES KOMPRESI

Input String = /WED/WE/WEE/WEB/WET			
Character Input	Code Output	New code value	New String
/W	/	256	/W
E	W	257	WE
D	E	258	ED
/	D	259	D/
WE	256	260	/WE
/	E	261	E/
WEE	260	262	/WEE
/W	261	263	E/W
EB	257	264	WEB
/	B	265	B/
WET	260	266	/WET
EOF	T		

III. METODA PENELITIAN

Tahap pertama dalam penelitian adalah tahap pengumpulan data berupa teks dari beberapa sumber seperti konten teks berita di beberapa halaman situs lokal, konten teks pada beberapa penyedia buku elektronik baik dari halaman situs perpustakaan nasional maupun dari halaman situs lembaga ilmu pengetahuan Indonesia. Seluruh teks yang diambil adalah berbahasa Indonesia. Data teks ini lalu dipindahkan ke dalam bentuk berkas teks dan dikelompokkan menjadi beberapa berkas sesuai besar ukurannya, mulai dari 1,2 KB hingga 1130 KB.

Tahap selanjutnya adalah tahap analisa terhadap beberapa algoritma yang akan diterapkan. Algoritma tersebut adalah *bit reduction*, *Huffman*, dan *LZW*. Untuk *bit reduction*, berdasarkan kode *ASCII* maka akan mengurangi 3 bit dari standar 8 bit setiap karakternya. Karakter yang akan dilakukan kompresi terhadapnya adalah karakter huruf kecil (a-z), karakter angka (0-9) dan seluruh karakter symbol. Untuk dua algoritma lainnya tidak ada perubahan.

Tahap yang ketiga adalah tahap menggabungkan ketiga algoritma tersebut ke dalam tiga langkah kompresi. Penelitian sebelumnya telah melakukan proses penggabungan dua algoritma dalam dua tahap kompresi teks. Penelitian ini mengambil tiga algoritma yang berbeda namun ketiga algoritma ini telah menunjukkan hasil kompresi yang terbaik dari beberapa algoritma lainnya pada penelitian-penelitian yang pernah dilakukan sebelumnya. Gabungan dari ketiga algoritma ini menghasilkan enam kombinasi langkah teknik kompresi teks. Kombinasi ini adalah menempatkan ketiga algoritma ini yang akan berperan pada langkah pertama, kedua atau ketiga. Proses kompresi teks nya adalah algoritma pada langkah pertama mengambil *input* teks dari sumber teks yang asli (*dataset*) sehingga menghasilkan karakter hasil kompresi teks algoritma langkah pertama. Pada langkah kedua, algoritma yang berbeda dari langkah pertama mengambil *input* teks dari karakter hasil kompresi algoritma langkah pertama sehingga menghasilkan karakter baru hasil kompresi algoritma langkah kedua. Pada langkah ketiga, algoritma yang digunakan berbeda dengan algoritma langkah pertama dan algoritma langkah kedua. Pada langkah ketiga mengambil *input* teks dari hasil kompresi teks algoritma langkah kedua sehingga menghasilkan karakter hasil kompresi teks yang terakhir.

Setelah ketiga algoritma ini diterapkan ke dalam sebuah program menggunakan bahasa pemrograman PHP maka dilakukan proses kompresi teks. Proses ini pertama kali mengambil sumber teks dari data teks yang telah disiapkan sebelumnya. Setelah terbaca dan tampil pada kotak inputnya maka dilakukan pemilihan algoritma pada langkah pertama, langkah kedua dan langkah ketiga. Algoritma yang dipilih pada langkah pertama, kedua dan ketiga harus berbeda. Kemudian dilakukan proses kompresi teks. Hasil akhir dari setiap kombinasi kompresi teks ini akan menghasilkan karakter hasil ketiga kompresi dalam ukuran bit. Hasil ini lanjutkan dengan pengukuran rasio kompresi, persentase penyimpanan, waktu proses dan hasil entropy nya.

IV. HASIL DAN PEMBAHASAN

Percobaan ini terdiri dari 6 kombinasi tiga langkah kompresi dan dekompresi teks. Kombinasi ini adalah urutan penempatan tiga algoritma. Algoritma-algoritmanya adalah *Bit Reduction*, *Huffman*, dan *LZW*. Percobaan ini melibatkan perbandingan diantara kombinasi tiga langkah dengan kombinasi dua langkah.

A. Kombinasi Pertama

Kombinasi pertama ini adalah *Bit Reduction* pada langkah pertama, *Huffman* pada langkah kedua dan *LZW* pada langkah ketiga. Perbandingan dua langkah (*Bit Reduction* and *Huffman*) dan tiga langkah (*Bit Reduction*, *Huffman* dan *LZW*). Terlihat pada Tabel IV.

TABEL IV
PENGUKURAN KOMBINASI PERTAMA (RATA-RATA)

Combination	Compression Ratio (%)	Saving Percentage (%)	Time Process (Minutes)	Entropy (Bits/Char)
Two Stages	61.8	38.4	0.031	6.41
Three Stages	45.7	54.3	0.034	7.90

B. Kombinasi Kedua

Kombinasi kedua adalah *Bit Reduction* pada langkah pertama, *LZW* pada langkah kedua dan *Huffman* pada langkah ketiga. Perbandingan dua langkah (*Bit Reduction* and *LZW*) dan tiga langkah (*Bit Reduction*, *LZW* dan *Huffman*). Terlihat pada Tabel V.

TABEL V
PENGUKURAN KOMBINASI KEDUA (RATA-RATA)

Algorithms Combination	Compression Ratio (%)	Saving Percentage (%)	Time Process (Minutes)	Entropy (Bits/Char)
Two Stages	34.4	65.6	6.988	6.41
Three Stages	31.4	55.2	7.520	7.84

C. Kombinasi Ketiga

Kombinasi ketiga adalah *Huffman* pada langkah pertama, *LZW* pada langkah kedua dan *Bit Reduction* pada langkah ketiga. Perbandingan dua langkah (*Huffman* and *LZW*) dan tiga langkah (*Huffman*, *LZW*, dan *Bit Reduction*). Terlihat pada Tabel VI.

TABEL VI
PENGUKURAN KOMBINASI KETIGA (RATA-RATA)

Algorithms Combination	Compression Ratio (%)	Saving Percentage (%)	Time Process (Minutes)	Entropy (Bits/Char)
Two Stages	34.7	65.3	4.777	7.78
Three Stages	31.9	66.9	5.331	7.91

D. Kombinasi Keempat

Kombinasi keempat adalah *Huffman* pada langkah pertama, *Bit Reduction* pada langkah kedua dan *LZW* pada langkah ketiga. Perbandingan dua langkah (*Huffman* and *Bit Reduction*) dan tiga langkah (*Bit Reduction*, *LZW* dan *Huffman*). Terlihat pada table VII.

TABEL VII
PENGUKURAN KOMBINASI KEEMPAT (RATA-RATA)

Algorithms Combination	Compression Ratio (%)	Saving Percentage (%)	Time Process (Minutes)	Entropy (Bits/Char)
Two Stages	50.2	49.7	0.020	7.65
Three Stages	32.4	64.6	0.250	7.78

E. Kombinasi Kelima

Kombinasi kelima adalah LZW pada langkah pertama, Huffman pada langkah kedua dan Bit Reduction pada langkah ketiga. Perbandingan dua langkah (LZW and Huffman) dan tiga langkah (LZW, Huffman dan Bit Reduction). Terlihat pada Table VIII.

TABEL VIII
PENGUKURAN KOMBINASI KELIMA (RATA-RATA)

Algorithms Combination	Compression Ratio (%)	Saving Percentage (%)	Time Process (Minutes)	Entropy (Bits/Char)
Two Stages	27.6	72.4	5.216	7.91
Three Stages	24.4	74.1	5.270	7.82

F. Kombinasi Keenam

Kombinasi keenam adalah LZW pada langkah pertama, Bit Reduction pada langkah kedua dan Huffman pada langkah ketiga. Perbandingan dua langkah (LZW and Bit Reduction) dan tiga langkah (LZW, Bit Reduction and Huffman). Terlihat pada Tabel IX.

TABEL IX
PENGUKURAN KOMBINASI KEENAM (RATA-RATA)

Algorithms Combination	Compression Ratio (%)	Saving Percentage (%)	Time Process (Minutes)	Entropy (Bits/Char)
Two Stages	25.7	74.3	5.706	7.91
Three Stages	23.5	74.3	6.013	7.67

G. Analisa Percobaan

1. Result Analysis

Berikut adalah rata-rata rasio kompresi, waktu proses dan entropy untuk dua dan tiga kombinasi.

- Kombinasi dua langkah

TABEL X
TWO STAGES COMBINATION (AVERAGE)

Measurements	Two Stage Combinations (Average)					
	BR + Huffman	Huffman + BR	BR + LZW	LZW + BR	Huffman + LZW	LZW + Huffman
Compression Ratio (%)	61.8	50.2	34.4	25.7	34.7	27.6
Processing Time (Minutes)	0.031	0.020	6.988	5.706	4.777	5.216
Entropy (Bits/Symbol)	4.94	4.03	2.53	1.88	2.69	2.09

Tabel X menunjukkan kombinasi dua algoritma seperti LZW dan bit reduction yang memberikan rasio kompresi terbaik yaitu 25.7%. Huffman coding dan bit reduction memberikan waktu proses tercepat yaitu 0.020 menit. LZW dan bit reduction juga memberikan hasil entropy 1.88 %. Terlihat pada Tabel XI berikut.

TABEL XI
PERBANDINGAN DUA LANGKAH

Two Stage		
Best Compression Ratio	25.7	LZW + Bit Reduction
Best Processing Time	0.020	Huffman + Bit Reduction
Best Entropy	1.88	LZW + Bit Reduction

- Kombinasi Tiga Langkah

Tabel XII menunjukkan bahwa kombinasi tiga algoritma seperti LZW, bit reduction dan Huffman memberikan kompresi rasio terbaik yaitu 23.5%. Bit reduction, Huffman coding, dan LZW memberikan waktu tercepat yaitu 0.034 menit. LZW, bit reduction, dan Huffman juga memberikan hasil entropy yaitu 1.88 %. Terlihat pada Table XIII.

TABEL XII
THREE STAGES COMBINATION (AVERAGE)

Measurements	Three Stage Combinations (Average)					
	BR + Huffman + LZW	Huffman + BR + LZW	BR + LZW + Huffman	LZW + BR + Huffman	Huffman + LZW + BR	LZW + Huffman + BR
Compression Ratio (%)	43.0	32.4	31.4	23.5	31.9	24.4
Processing Time (Minutes)	0.034	0.250	7.520	6.013	5.311	5.710
Entropy (Bits/Symbol)	3.44	2.60	2.52	1.88	2.56	1.96

TABEL XIII
PERBANDINGAN TIGA LANGKAH

Three Stage		
Best Compression Ratio	23.5	LZW + Bit Reduction + Huffman
Best Processing Time	0.034	Bit Reduction + Huffman + LZW
Best Entropy	1.88	LZW + Bit Reduction + Huffman

- Perbandingan Dua langkah dan Tiga Langkah

TABEL XIV
PERBANDINGAN DUA LANGKAH DAN TIGA LANGKAH

Two Stage	Three Stage	Comparison Two Stage with Three Stage					
		Status Compression Ratio (%)	Status Processing Time	Status Entropy			
BR + Huffman	BR + Huffman + LZW	Peningkatan	18.78	Penurunan	0.003	Peningkatan	1.50
Huffman + BR	Huffman + BR + LZW	Peningkatan	17.80	Penurunan	0.230	Peningkatan	1.43
BR + LZW	BR + LZW + Huffman	Peningkatan	3.00	Penurunan	0.532	Peningkatan	0.01
LZW + BR	LZW + BR + Huffman	Peningkatan	2.18	Penurunan	0.307	Tidak Berubah	0.00
Huffman + LZW	Huffman + LZW + BR	Peningkatan	2.84	Penurunan	0.534	Peningkatan	0.13
LZW + Huffman	LZW + Huffman + BR	Peningkatan	3.24	Penurunan	0.494	Peningkatan	0.13

Tabel XIV menunjukkan bahwa kombinasi Bit reduction, Huffman (dua langkah) memberikan peningkatan kinerja ketika ditambah satu langkah dengan LZW yaitu 18.78% dan kombinasi LZW, bit reduction (dua langkah) memberikan penurunan kinerja ketika ditambahkan dengan Huffman.

V. PENUTUP

Kombinasi tiga langkah mengalami peningkatan rasio kompresi dan bekerja lebih baik dari pada dua langkah dikarenakan menambahkan satu algoritma pada langkah selanjutnya akan mengkompresi lebih daripada dua langkah, meskipun hanya sedikit bit yang sukses dikompresi daripada ditempatkan pada baris pertama. Kombinasi dua algoritma dalam dua langkah seperti LZW dan Bit Reduction telah menunjukkan kinerja rasio kompresi terbaik. Ketika algoritma ini ditambahkan dengan Huffman pada kombinasi

tiga langkah, algoritma-algoritma ini tetap memberikan kinerja terbaik pada rasio kompresi meskipun hanya sedikit bit. Hasil ini karena LZW pada posisi pertama menghasilkan karakter kompresi baru dengan jumlah bit lebih besar dari 255. Ketika bit reduction ditempatkan pada urutan kedua yang mengambil karakter input dari kompresi LZW maka bit reduction hanya dapat mengenali dan mengkompresi sedikit karakter.

Kombinasi dua algoritma dalam dua langkah seperti bit reduction dan Huffman meningkatkan rasio kompresi secara signifikan. Ketika ditambahkan dengan LZW pada urutan ketiga tidak memberikan halangan dalam mengkompresi karakter-karakter baru yang dihasilkan oleh Huffman pada langkah kedua. Sebagai pengembangan lebih lanjut penelitian ini dapat ditingkatkan dengan kombinasi lebih dari tiga langkah dan dengan algoritma yang sama ataupun menambah algoritma baru. Terdapat sebuah kemungkinan untuk melakukan kompresi dengan dua atau tiga langkah meskipun menggunakan algoritma yang sama pada setiap langkahnya. signifikan. Ketika ditambahkan dengan LZW pada urutan ketiga tidak memberikan halangan dalam mengkompresi karakter-karakter baru yang dihasilkan oleh Huffman pada langkah kedua.

Sebagai pengembangan lebih lanjut penelitian ini dapat ditingkatkan dengan kombinasi lebih dari tiga langkah dan dengan algoritma yang sama ataupun menambah algoritma baru. Terdapat sebuah kemungkinan untuk melakukan kompresi dengan dua atau tiga langkah meskipun menggunakan algoritma yang sama pada setiap langkahnya.

UCAPAN TERIMA KASIH

Ucapan terima kasih yang sebesar-besarnya diberikan kepada Research Center Universitas Presiden yang telah memberikan bantuan yang sangat berarti bagi penelitian ini. Juga kepada semua pihak yang terlibat secara langsung maupun tidak langsung yang tidak dapat disebutkan satu persatu disini disampaikan penghargaan yang setinggi-tingginya.

REFERENSI

- [1] Altarawneh, Haroon, and Mohammad Altarawneh. "Data Compression Techniques on Text Files", *International Journal of Computer Applications*, vol. 26, no. 5. hal. 42-54, 2011.
- [2] Blelloch, Guy E. *Introduction to Data Compression*, Edisi Pertama, Carnegie Mellon University: Computer Science Department, 2013, Web. 26 Jan. 2016.
- [3] Brar, Rupinder Singh, and Bikramjeet Singh. "A Survey on Different Compression Techniques and Bit Reduction Algorithm for Compression of Text/Lossless Data", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no.3. hal. 579-582, 2013.
- [4] Chakraborty, Debashish, Sandipan Bera, Anil Kumar Gupta, and Soujit Mondal. "Simple Data Compression by Differential Analysis Using Bit Reduction and Number System Theory", *International Journal on Information Technology*, vol. 1, no.3. hal. 16-18, 2011.
- [5] Cormack, G. V., and R. N. S. Horspool. "Data Compression Using Dynamic Markov Modelling", *The Computer Journal*, vol. 30, no. 6. hal. 542-550, 1987.
- [6] Dheemanth, H.N. "LZW Data Compression", *American Journal of Engineering Research*, vol. 3, no. 2. hal. 22-26, 2014.

- [7] Hasan, Md. Rubaiyat. "Data Compression Using Huffman Based LZW Encoding Technique", *International Journal of Scientific & Engineering Research*, vol. 2, no. 11. hal. 1-7, 2011.
- [8] Held, Gilbert, and Thomas R. Marshall. *Data and Image Compression*. Edisi Keempat, England: John Wiley and Sons Ltd.1996.
- [9] Kaur, Amarjit, Navdeep Singh Sethi, and Harinderpal Singh. "A Review on Data Compression Techniques", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 1. hal. 769-773, 2015.
- [10] Kodituwakku, S.R., and U. S. Amarasinghe. "Comparison of Lossless Data Compression Algorithms for Text Data", *Indian Journal of Computer Science and Engineering*, vol. 1, no. 4. hal. 416-425, 2016.
- [11] Pathak, Shivani, Shradha Singh, Smita Singh, Mamta Jain, and Anand Sharma. "Data Compression Scheme of Dynamic Huffman Code for Different Languages", *International Conference on Information and Network Technology Press Singapore*, vol.4. hal. 201-206, 2011.
- [12] Raja, P., and D. Saraswathi. "An Effective Two Stage Text Compression and Decompression Technique for Data Communication", *International Journal of Electronics and Communication Engineering*, vol. 4, no. 2. hal. 233-241, 2011.
- [13] Sharma, Neha, Dr. Paramjeet Singh, and Dr. Shaveta Rani. "An Improved Dynamic Bit Reduction Algorithm for Lossless Text Data Compression", *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 4, no. 7. hal. 1023-1029, 2014.
- [14] Suarjaya, I Made Agus Dwi. 2012. "A New Algorithm for Data Compression Optimization", *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 8. hal. 14-17, 2012.

