

# Impelementasi Kompresi Teks Menggunakan Metode Huffman untuk Menghemat Karakter pada Short Message Service

Evi Mariani Harahap<sup>1</sup>, Dian Rachmawati, S.Si, M.Kom<sup>2</sup>, Herriyance, ST, M.Kom<sup>3</sup>

Program Studi Ekstensi SI Ilmu Komputer, Universitas Sumatera Utara

Jalan Universitas No. 9 Kampus USU Medan 20155

miss3us@student.usu.ac.id<sup>1</sup>

dee230783@gmail.com<sup>2</sup>

Herriyance\_usu@yahoo.com<sup>3</sup>

**Abstrak**— Kajian ini membahas mengenai implementasi metode *Huffman* statis dalam penggunaan SMS yang masih memiliki kelemahan yaitu keterbatasan penulisan teks SMS untuk karakter 7-bit yang hanya sebanyak 160 karakter untuk satu kali kirim. Oleh sebab itu diperlukan proses kompresi dan dekompresi pada teks SMS dengan membuat aplikasi *Huffman* Kompresi, sehingga mampu menjadikan ukuran SMS tersebut seminimal mungkin pada saat dikirimkan. Setiap karakter yang dituliskan akan diubah menjadi bentuk bit sesuai dengan tabel statis *Huffman*, rangkaian bit tersebut akan dikirimkan ke nomor tujuan dengan menggunakan mode pengiriman *BinaryMessage*. Aplikasi ini dibuat dengan menggunakan bahasa Java yaitu *J2ME* dan berjalan pada *mobile phone* berbasis Java *MIDP 2.0* dan konfigurasi *CLDC 1.1*. Pengirim dan penerima bisa berkomunikasi apabila aplikasi sudah terinstal dan dijalankan di *mobile phone* keduanya. Aplikasi ini bertujuan untuk menghemat pemakaian SMS dengan meminimalisasikan ukuran data yang akan dikirimkan. Sehingga dapat menghasilkan rasio penghematan kompresi sekitar 31%.

**Kata Kunci**— Kompresi, Dekompresi, Metode *Huffman*, SMS, Teks, Data.

## I. PENDAHULUAN

### A. Latar Belakang

Manusia sebagai makhluk sosial membutuhkan sarana komunikasi untuk membangun hubungan sosialnya di dalam masyarakat. Komunikasi merupakan usaha penyampaian pesan antar manusia [14]. SMS salah satu sarana komunikasi teknologi seluler yang paling banyak diminati untuk berkomunikasi cepat, murah dan jelas dalam kehidupan sehari-hari. Namun sebuah pesan yang akan dikirimkan hanya memiliki kapasitas maksimal 160 karakter mode 7-bit untuk 1 kali kirim atau 1120 bit. Sehingga pengguna SMS harus memadatkan kata yang digunakan [17].

Kompresi teks merupakan salah satu teknik untuk mengatasi penghematan pemakaian karakter SMS. Dengan adanya proses kompresi terhadap teks SMS, akan terjadi pemampatan terhadap data SMS sehingga dapat menghemat biaya pengiriman SMS (pulsa).

Penulis memilih metode *Huffman* karena memiliki tingkat kompresi terbesar dalam melakukan proses kompresi teks seperti penelitian yang dilakukan pada file teks berukuran asli 172012 *byte*, setelah dikompresi menggunakan metode *Half*

*Byte* ukurannya berubah menjadi 171124 *byte*, dengan metode *Run Length* menjadi 171062 *byte*, sedangkan pada saat dikompresi dengan metode *Huffman* ukurannya berubah menjadi 113056 *byte* [6].

Hal inilah yang mendasari penulis untuk membuat skripsi dengan judul “Implementasi Kompresi Teks Menggunakan Metode *Huffman* Untuk Menghemat Karakter Pada Short Message Service”.

### B. Rumusan Masalah

Bagaimana proses kompresi serta dekompresi teks menggunakan metode *Huffman* untuk menghemat pemakaian SMS serta seberapa besar rasio penghematan kompresi teks SMS yang dapat dihasilkan dengan pengimplementasian metode *Huffman*.

### C. Batasan Masalah

Batasan-batasan masalah dalam penelitian ini adalah:

1) Sistem terbatas hanya pada sistem mengkompresi dan dekompresi jumlah karakter SMS.

2) Batasan maksimal karakter yang dikompresi adalah 1120 karakter.

3) Kompresi yang dilakukan hanya pada teks SMS yang memiliki variasi karakter meliputi alfabet, numerik dan simbol.

4) Aplikasi yang di buat ter-*install* dan dijalankan pada *mobile phone* pengirim dan penerima.

5) Ditujukan pada *mobile phone* berbasis java yang mendukung profil *MIDP 2.0* dan konfigurasi *CLDC 1.1*.

6) Bahasa pemrograman yang akan digunakan untuk membuat aplikasi adalah bahasa *Java2 Micro Edition (J2ME)* dan tidak dibahas secara khusus dan detail.

### D. Tujuan Penelitian

Mengimplementasikan metode *Huffman* dalam melakukan kompresi dan dekompresi pada teks SMS agar ukuran data yang akan dikirimkan dapat seminimal mungkin, serta mengetahui beberapa rasio kompresi yang dapat dihasilkan.

### E. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut:

1) Bagi teknologi adalah dapat mengembangkan *software* aplikasi untuk *mobile phone* berbasis java yang dapat meningkatkan jumlah karakter teks SMS sehingga dapat meminimalisasi biaya pengiriman SMS.

2) Bagi masyarakat adalah untuk menghemat pemakaian SMS dengan meminimalisasikan ukuran data yang akan dikirimkan, sehingga komunikasi melalui SMS lebih leluasa.

3) Adapun bagi penulis bermanfaat untuk mengaplikasikan ilmu yang didapat selama kuliah.

## II. LANDASAN TEORI

### A. SMS (Short Message Service)

SMS merupakan suatu fasilitas untuk mengirim dan menerima suatu pesan singkat berupa teks melalui perangkat komunikasi telepon selular. Pada proses pengiriman SMS, SMS tersebut tidak *langsung* dikirimkan dari mobile phone pengirim ke mobile phone tujuan, akan tetapi terlebih dahulu dikirim ke Short Message Service Center (SMSC), kemudian dengan system store and forward SMS tersebut dikirimkan ke mobile phone tujuan.



Gambar 1 Cara Kerja SMS

Di SMSC pesan disimpan dan dicoba dikirim selama beberapa kali, setelah selang waktu yang telah ditentukan, biasanya 1 hari atau 2 hari, lalu pesan dihapus [8]. SMS bisa pula untuk mengirim gambar, suara dan film, SMS bentuk ini disebut Multimedia Message Service (MMS) [1].

### B. Data

Makna kata data di komputer adalah segala sesuatu yang bisa dikodekan, disimbolkan atau dilambangkan dengan kode-kode, simbol-simbol atau lambang-lambang yang telah disediakan disetiap komputer termasuk suatu teks [15].

### C. Teks

Teks adalah kumpulan dari karakter atau string yang menjadi satu kesatuan berupa teks atau huruf, angka, spasi dan tanda baca lainnya yang mampu menampung karakter dari huruf dan angka tersebut. Pemrosesan teks (*text processing*) adalah salah satu bentuk pengolahan data non-numerik [12].

### D. Kompresi Data

Kompresi data adalah suatu proses mereduksi ukuran suatu data dengan mengubah sekumpulan data tersebut menjadi sekumpulan kode yang dapat menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data [7]. Berdasarkan kompresi yang dihasilkan, metode kompresi dapat dibagi ke dalam dua kategori, yaitu metode kompresi *lossy* dan metode kompresi *lossless* [1,7].

### 1) Metode kompresi *lossy*

Metode kompresi *lossy* adalah data yang dikompresi akan menghasilkan data yang tidak sama dengan aslinya ketika didekompresi, tetapi cukup dan masih bisa digunakan sebatas keperluan. Contoh metode *lossy* adalah metode CS&Q (*coarser sampling and/or quantization*), JPEG, dan MPEG. Kompresi tipe ini cocok diterapkan pada JPEG, MPEG, 3GP, MP3, OGG, dan lain-lain.

### 2) Metode kompresi *lossless*

Metode kompresi *lossless* adalah metode kompresi data yang menghasilkan data yang sama persis ketika didekompresi dengan data aslinya dengan merekonstruksi dari data yang dikompresi. Contoh beberapa metode kompresi *lossless* yaitu LZ77, LZW, metode *Huffman*, *Arithmetic coding*, DMC, GZIP, RAR, dan lain-lain. Kompresi tipe ini cocok diterapkan pada berkas basis data (database), *spread sheet*, berkas *word processing*, dan lain sebagainya.

### E. Metode Huffman

Metode *Huffman* ditemukan oleh David A. Huffman ketika ia melakukan studi Ph.D di MIT. Kode ini dipublikasikan pada tahun 1952 pada tulisannya yang berjudul “*A Method for the Construction of Minimum-Redudancy Codes*” [2].

#### 1) Metode Huffman Statis

Metode ini tergantung pada probabilitas dari setiap simbol yang hadir pada suatu data (pesan). Berdasarkan probabilitas tersebut kemudian dibentuk daftar kode untuk setiap simbol dengan ketentuan karakter yang paling sering muncul di dalam data dikodekan dengan kode yang jumlah bitnya lebih sedikit, sedangkan karakter yang jarang muncul dikodekan dengan kode yang jumlah bitnya lebih panjang.

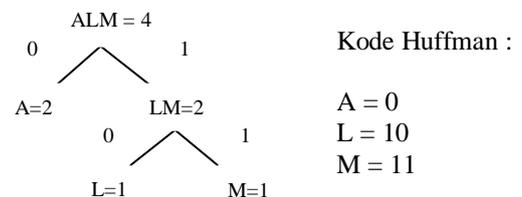
Contoh: Kode ASCII *string* 4 huruf “ALAM” membutuhkan representasi  $4 \times 8 \text{ bit} = 32 \text{ bit}$ , dengan rincian sebagai berikut: A = 01000001 L = 01001100 M = 01000001 M = 01001101. Pada *string* di atas, frekuensi kemunculan A = 2, L = 1 dan M = 1.

Maka Pengerjaan akan seperti berikut:

-L dan M menjadi 1 (LM) sehingga probabilitas menjadi  $1/4 + 1/4 = 2/4$

-LM dan A menjadi 1 (ALM) sehingga probabilitas menjadi  $2/4 + 2/4 = 1$

Setiap simpul yang terletak dengan frekuensi lebih besar diberi nilai 0 dan simpul yang terletak dengan frekuensi lebih kecil diberi nilai 1. Untuk frekuensi yang sama besar maka simpul yang terletak pada cabang kiri diberi nilai 0 dan untuk simpul yang terletak pada cabang kanan diberi nilai 1.



Gambar 2 Pohon Huffman untuk Karakter “ALAM”

Dihasilkan kode baru yaitu A menjadi 1 bit dengan frekuensi kemunculan 2 kali sehingga  $2 \times 1 = 2$ , L:  $1 \times 2 = 2$  dan M:  $1 \times 2 = 2$ . Sehingga hanya membutuhkan representasi 6 bit. Dapat dihemat sebanyak  $32 - 6 = 26$  bit.

### 2) Metode Huffman Dinamis

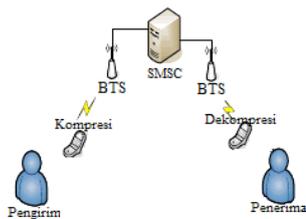
Metode ini merupakan kelanjutan dari metode Huffman statis dengan menambah atau mengurangi suatu proses tertentu pada metode Huffman statis. Ide dasar dari metode ini adalah meringkas tahapan metode Huffman tanpa perlu menghitung jumlah karakter keseluruhan dalam membangun pohon biner.

Metode Huffman dinamis adalah suatu metode dengan kemungkinan kemunculan dari setiap simbol tidak dapat ditentukan dengan pasti selama pengkodean. Hal ini disebabkan oleh perubahan pengkodean secara dinamis berdasarkan frekuensi dari simbol yang telah diolah sebelumnya. Metode ini dikembangkan oleh trio Faller, Gallager dan Knuth, kemudian dikembangkan lebih lanjut oleh Vitter, selanjutnya baca [5].

## III. ANALISIS DAN PERANCANGAN SISTEM

### A. Analisis Kompresi SMS

Aplikasi kompresi ini hanya dapat berjalan antar mobile phone GSM yang mendukung Java MIDP 2.0 dan konfigurasi CLDC 1.1.



Gambar 3 Cara Kerja SMS Kompresi-Dekomposisi

Berdasarkan gambar di atas, maka sistem aplikasi kompresi ini dapat dideskripsikan sebagai berikut:

- Aplikasi terinstal di kedua mobile phone dan dijalankan terlebih dahulu kemudian ketik pesan pada text book yang tersedia, pesan terkompresi dan pengguna dapat melanjutkan proses pengiriman pesan.
- Pesan dikirimkan ke SMSC melalui jaringan GSM dengan melewati beberapa Base Transceiver Station (BTS) terlebih dahulu (Pesan tidak langsung dikirimkan ke mobile phone penerima, melainkan melalui SMSC terlebih dahulu).
- Pada BTS terakhir (paling dekat dengan SMSC) pesan dikirimkan ke SMSC melalui jaringan kabel. Pada SMSC ini pesan yang telah terkompresi disimpan sementara guna kebutuhan informasi seperti delivery report, status pending atau failed.
- Kemudian diteruskan oleh BTS satu kepada BTS lain sampai pada BTS yang melayani jaringan mobile phone penerima.
- Ketika pesan terkirim pada aplikasi mobile phone pengirim maka mobile phone penerima akan

mengetahui adanya pesan masuk pada port yang ditentukan, lalu menampilkan hasil kepada pengguna bahwa ada pesan.

- Penerima yang menyetujui pesan untuk dibaca maka sistem melakukan proses dekompresi pada pesan tersebut dan menampilkan hasil kepada pengguna.

### B. Analisis Metode Huffman

Prinsip kerja metode Huffman adalah mengkodekan setiap karakter dalam representasi bit. Representasi bit untuk setiap karakter berbeda satu sama lain berdasarkan frekuensi kemunculan karakter. Semakin sering karakter itu muncul semakin pendek representasi bitnya. Sebaliknya semakin jarang karakter untuk muncul, maka semakin panjang representasi bit untuk karakter tersebut.

Pada penelitian ini, proses pembuatan Huffman tree tidak akan dilakukan di dalam program. Hal ini dilakukan karena penambahan informasi mengenai Huffman tree akan memerlukan tempat tersendiri sehingga proses kompresi menjadi kurang efektif. Hal ini diperkuat dengan hasil penelitian yang menyimpulkan bahwa jika ukuran file yang dikompresi kecil maka file hasil kompresi bisa jadi bukannya semakin kecil tetapi malah semakin besar karena harus menyimpan Huffman tree yang dihasilkan<sup>[4]</sup>.

Untuk menggantikan informasi mengenai Huffman tree tersebut maka dibuat suatu tabel Huffman yang terbentuk dari tree itu sendiri dalam bentuk biner, yang berisi kode-kode Huffman dari karakter-karakter SMS default yang akan digunakan. Tabel ini bersifat statis dan akan digunakan oleh aplikasi, baik aplikasi pengirim maupun penerima, sebagai acuan untuk melakukan proses kompresi atau dekompresi terhadap teks SMS.

### C. Rancangan Pembentukan Tabel Statis Tulis Baru

Aplikasi kompresi ini akan dibangun dengan menggunakan daftar kode Huffman yang telah ada, kode Huffman ini terdiri atas awalan dan badan. Awalan dan badan merupakan pengelompokan agar tabelnya lebih kompleks dan rapi.

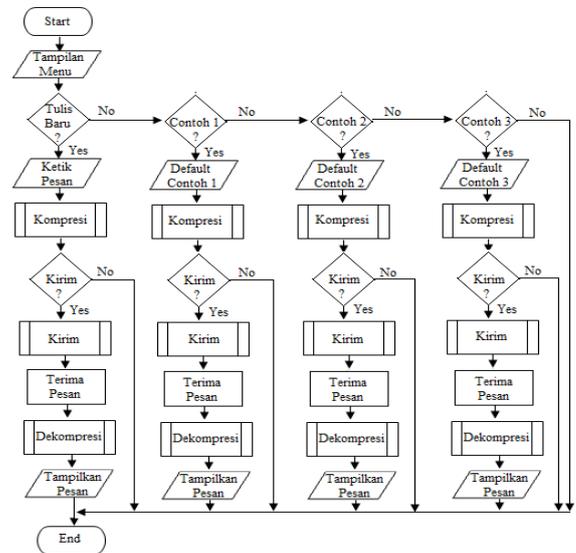
TABLE I  
TABEL STATIS KODE HUFFMAN

Karakter	Awalan	Badan	Ubahan
spasi	101	1	1011
a		0	1010
b	0	00100	000100
c		0101000	00101000
d		0000	000000
e		0001	000001
f		110001	0110001
g		00101	000101
h		01011	001011
i		0011	00011
j		10100	010100
k		10101	010101
l		0100	00100
m		0110	00110
n		111	0111
o		10110	010110
p		10111	010111

q	0	0101001	00101001
r		0111	00111
s		1000	01000
t		1001	01001
u		1101	01101
v		0101010	00101010
w		0101011	00101011
x		1100000	01100000
y		11001	011001
z		1100001	01100001
0	111	0000	1110000
1		0001	1110001
2		0010	1110010
3		0011	1110011
4		0100	1110100
5		0101	1110101
6		0110	1110110
7		0111	1110111
8		1000	1111000
9		1001	1111001
@		1010	1111010
/		1011	1111011
:		1100	1111100
(		1101	1111101
)		1111	1111111
sisa huruf		1110	1111110
.	110	0000	1100000
,		0001	1100001
?		0010	1100010
!		0011	1100011
'		0100	1100100
-		0101	1100101
“		0110	1100110
A		0111	1100111
B		1000	1101000
G		1001	1101001
H		1010	1101010
J		1011	1101011
K		1100	1101100
O		1101	1101101
P		1110	1101110
Y		1111	1101111
C	100	0000	1000000
D		0001	1000001
E		0010	1000010
F		0011	1000011
I		0100	1000100
L		0101	1000101
M		0110	1000110
N		111	100111
Q		11000	10011000
R		0111	1000111
S		1000	1001000
T		1001	1001001
U		1010	1001010
V		11001	10011001
W		1011	1001011
X		11010	10011010
Z	11011	10011011	

#### D. Rancangan Flowchart Program Utama Aplikasi

Menggambarkan tampilan menu utama dari program yang dibangun. Pada Gambar 4 proses dimulai dengan menampilkan menu yang ada yang terdiri dari Tulis Baru, Contoh 1, Contoh 2 dan Contoh 3.

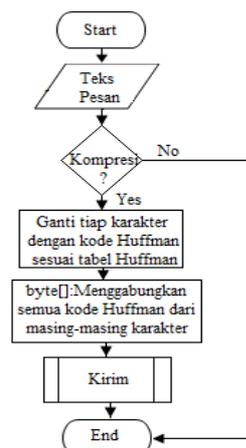


Gambar 4 Flowchart Program Utama Aplikasi

Tulis pesan pada Tulis Baru untuk menulis pesan baru dengan menggunakan tabel statis kode Huffman dan Contoh 1, Contoh 2, Contoh 3 untuk mengirim pesan Contoh 1, Contoh 2 dan Contoh 3 dengan menggunakan tabel default Huffman statis masing-masing, pesan akan dikompresi kemudian kirim pesan yang sudah terkompresi. Pesan diterima kemudian pesan yang akan ditampilkan didekompresi.

#### E. Rancangan Flowchart Proses Kompresi

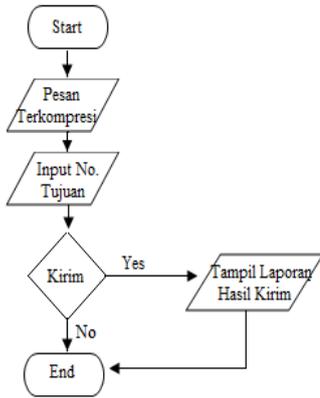
Flowchart ini berfungsi untuk menggambarkan proses kompresi pada aplikasi. Dapat dilihat pada Gambar 5 gambar berikut:



Gambar 5 Flowchart Kompresi

### F. Rancangan Flowchart Kirim

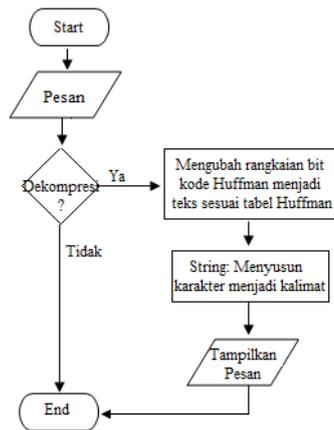
Flowchart Kirim dapat dilihat pada Gambar 6. Pesan yang dikirim adalah pesan yang sudah terkompresi.



Gambar 6 Flowchart Kirim

### G. Flowchart Dekompresi

Berfungsi untuk menggambarkan proses dekompresi pada aplikasi.



Gambar 7 Flowchart Dekompresi

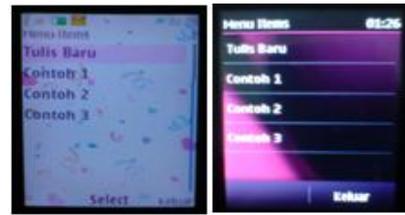
### H. Rasio Kompresi

Penghitungan rasio dilakukan dengan tujuan untuk mengetahui seberapa jauh pengaruh proses kompresi terhadap jumlah karakter SMS yang dihasilkan dari data sebelum dilakukan proses kompresi dengan data setelah dilakukan proses kompresi. Jika jumlah bit sebelum dikompresi disimbolkan dengan Nsebelum dan jumlah bit setelah kompresi disimbolkan dengan Nsesudah, maka rumus yang digunakan dalam penelitian ini adalah dengan persamaan berikut:

$$\text{Rasio} = (1 - (\text{Nsesudah} / \text{Nsebelum})) \times 100\% \dots \text{pers}(1) [8]$$

## IV. IMPLEMENTASI SISTEM

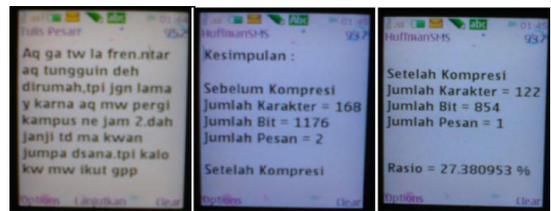
Tampilan aplikasi dapat dilihat pada gambar berikut:



Gambar 8 Tampilan Aplikasi di Mobile Phone Pengirim dan Penerima

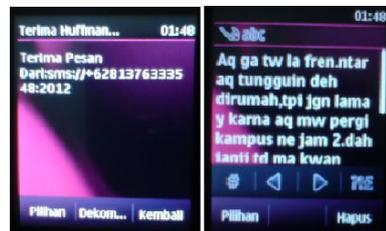
### A. Tampilan penggunaan Tulis Baru

Dihalaman ini pengirim dapat menuliskan pesan yang akan dikirimkan. Contoh teks yang ditulis adalah “Aq ga tw la fren.ntar aq tungguin deh dirumah,tpi jgn lama y karna aq mw pergi kampus ne jam 2.dah janji td ma kwan jumpa dsana.tpi kalo kw mw ikut gpp kok.aq tunggu yo”



Gambar 9 Hasil Kompresi Sebelum dan Sesudah Kompresi Tulis Baru

Pesan akan diterima mobile phone penerima dan pilih Dekompres untuk membaca pesan, hasilnya akan ditampilkan sesuai pesan yang dikirim seperti Gambar 10:



Gambar 10 Terima Pesan untuk Tulis Baru

### B. Tampilan penggunaan Contoh 1

Pesan ini berisikan pesan default “Ilmu Komputer USU 2010”



Gambar 11 Hasil Kompresi Sebelum dan Sesudah Kompresi Contoh 1

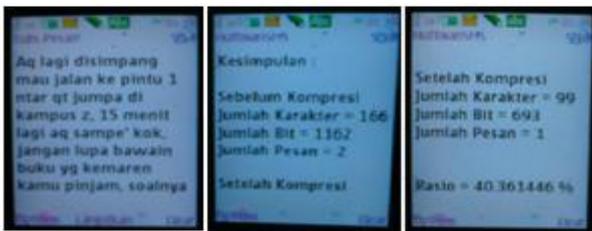
Pesan akan diterima *mobile phone* penerima dan pilih Dekompres 1 untuk membaca pesan Contoh 1, hasilnya akan ditampilkan seperti Gambar 12:



Gambar 12 Terima Pesan Contoh 1

### C. Tampilan penggunaan Contoh 2

Pesan ini berisikan pesan *default* "Aq lagi disimpang mau jalan ke pintu 1 ntar qt jumpa di kampus z, 15 menit lagi aq sampe' kok, jangan lupa bawain buku yg kemaren kamu pinjam, soalnya aq mau belajar."



Gambar 13 Hasil Kompresi Sebelum dan Sesudah Kompresi Contoh 2

Pesan akan diterima *mobile phone* penerima dan pilih Dekompres 2 untuk membaca pesan Contoh 2, hasilnya akan ditampilkan seperti Gambar 14 berikut:



Gambar 14 Terima Pesan Contoh 2

### D. Tampilan penggunaan Contoh 3

Pesan ini berisikan pesan *default* "Undangan Seminar Blogging (bagi pemula). Akan diadakan pada tanggal 15 Mei 2010 bertempat di Hotel Arya Duta, Medan, pada pukul: 10.00 s/d selesai. Undangan ini berlaku untuk pelajar, mahasiswa dan umum. Bagi yang berminat dapat mendaftar lewat SMS ke 0815 2222 222 dengan mengetik: seminar spasi nama spasi alamat. Peserta terbatas."



Gambar 15 Hasil Kompresi Sebelum dan Sesudah Kompresi Contoh 3

Pesan akan diterima *mobile phone* penerima dan pilih Dekompres 3 untuk membaca pesan Contoh 3, hasilnya akan ditampilkan seperti Gambar 16 berikut:



Gambar 16 Terima Pesan Contoh 1

### E. Pengujian Rasio

- Pengujian Pertama**  
 Teks : Evi Mariani Harahap  
 Jumlah bit sebelum dikompresi: 133 bit  
 Jumlah bit setelah dikompresi: 98 bit  
 Rasio:  $= (1 - (98/133)) \times 100\%$   
 $= 26.31579 \%$
- Pengujian Kedua**  
 Teks: Alangkah indahnya hidup ini andai dapat kutatap wajahmu  
 Jumlah bit sebelum dikompresi: 385  
 Jumlah bit setelah dikompresi: 268  
 Rasio:  $= (1 - (268/385)) \times 100\%$   
 $= 30.38961 \%$
- Pengujian Ketiga**  
 Teks: Saya adalah Evi Mariani Harahap, NIM saya 101421046, Jurusan ekstensi Ilmu Komputer USU  
 Jumlah bit sebelum dikompresi: 609  
 Jumlah bit setelah dikompresi: 465  
 Rasio:  $= (1 - (465/609)) \times 100\%$   
 $= 23.64532 \%$
- Pengujian Keempat**  
 Teks: Implementasi sistem merupakan prosedur yang dilakukan untuk menyelesaikan perancangan sistem yang ada. Untuk menjalankan sistem, perlu diperhatikan kebutuhan perangkat yang akan mendukung proses pembuatan aplikasi. Perangkat yang dibutuhkan merupakan perangkat keras dan perangkat lunak.  
 Jumlah bit sebelum dikompresi: 2009  
 Jumlah bit setelah dikompresi: 1403

$$\text{Rasio:} = (1 - (1403/2009)) \times 100\% \\ = 30.164259 \%$$

- Pengujian Kelima

Teks: Ilmu Komputer USU 2010  
 Jumlah bit sebelum dikompresi: 154  
 Jumlah bit setelah dikompresi: 87  
 Rasio:  $= (1 - (87/154)) \times 100\%$   
 $= 43.506493 \%$

- Pengujian Keenam

Teks: Aq lagi disimpang mau jalan ke pintu 1 ntar qt jumpa di kampus z, 15 menit lagi aq sampe' kok, jangan lupa bawain buku yg kemaren kamu pinjam, soalnya aq mau belajar.  
 Jumlah bit sebelum dikompresi: 1162  
 Jumlah bit setelah dikompresi: 693  
 Rasio:  $= (1 - (693/1162)) \times 100\%$   
 $= 40.361446 \%$

- Pengujian Ketujuh

Teks: Undangan Seminar Blogging (bagi pemula). Akan diadakan pada tanggal 15 Mei 2010 bertempat di Hotel Arya Duta, Medan, pada pukul: 10.00 s/d selesai. Undangan ini berlaku untuk pelajar, mahasiswa dan umum. Bagi yang berminat dapat mendaftar lewat SMS ke 0815 2222 222 dengan mengetik: seminar spasi nama spasi alamat. Peserta terbatas.  
 Jumlah bit sebelum dikompresi: 2331  
 Jumlah bit setelah dikompresi: 1492  
 Rasio:  $= (1 - (1492/2331)) \times 100\%$   
 $= 35.993134 \%$

## V. PENUTUP

### A. Kesimpulan

Berdasarkan pembahasan pada bab-bab sebelumnya, penulis dapat mengambil kesimpulan sebagai berikut:

1) Dengan melakukan kompresi serta dekompresi pada teks SMS menggunakan metode Huffman dapat menghemat pesan SMS yang akan dikirimkan.

2) Dengan menggunakan tabel statis Huffman Tabel 3.1, rasio pesan teks yang ditulis dapat menghasilkan persentase hingga 31% dan terhadap contoh default yang dibuat, maka rasio pesan teks dapat menghasilkan persentase hingga 44%.

3) Rasio yang dihasilkan pada kompresi teks tidak selalu sama, hal ini dikarenakan tidak semua karakter memiliki panjang bit ubahan yang sama.

4) Aplikasi ini hanya berjalan antar mobile phone yang berbasis java MIDP 2.0 dan CLDC 1.1, dan kedua user baik pengirim maupun penerima harus meng-install dan menjalankan aplikasi ini.

### B. Saran

Untuk pengembangan lebih lanjut, terdapat beberapa saran yang harus dipertimbangkan oleh pengembang selanjutnya:

1) Diharapkan pengembang selanjutnya dapat membuat aplikasi kompresi teks SMS dengan mengkombinasikan beberapa metode kompresi teks yang ada.

2) Diharapkan pengembang selanjutnya dapat membuat aplikasi dengan penghematan rasio pesan teks yang ditulis menghasilkan persentase lebih dari 31% seperti hasil rasio pada penggunaan contoh default yang dibuat hingga 44%.

3) Diharapkan pengembang selanjutnya dapat membuat aplikasi yang tidak hanya berjalan pada mobile phone berbasis java MIDP 2.0 dan CLDC 1.1, tetapi pada semua merek dan tipe mobile phone yang mendukung teknologi java.

4) Dalam aplikasi ini user tidak dapat menyimpan pesan yang dikirimkannya dan menyimpan pesan yang masuk. Jadi disarankan bagi pengembang selanjutnya untuk membuat fasilitas menyimpan pesan yang dikirimkan dan fasilitas menyimpan pesan yang masuk oleh user.

## DAFTAR PUSTAKA

- [1] Baldwin, Richard G. *Understanding the Huffman Data Compression Algorithm in Java*. [http://www.developer.com/java/other/article.php/10936\\_3603066\\_2](http://www.developer.com/java/other/article.php/10936_3603066_2). Akses tanggal 26 Mei 2012.
- [2] Kandaga, Tjatur. 2006. *Analisis Penerapan Kompresi dan Dekompresi Data dengan Menggunakan Metode Statistik dan Kamus*. Bandung. Maranatha Vol-2 No. 2: Jurnal Informatika.
- [3] Lamhot, S. *Kompresi Teks Menggunakan Algoritma Huffman*. Media Unika Tahun 20 No.71 Edisi ke-2: Jurnal.
- [4] Liliana, Lipesik, V.J. 2006. *Pembuatan Perangkat Lunak Untuk Kompresi File Text Dengan Menggunakan Huffman Tree*. [http://portfolio.petra.ac.id/user\\_files/03-024/kompresiFile.doc](http://portfolio.petra.ac.id/user_files/03-024/kompresiFile.doc). Akses Tanggal 12 Maret 2012.
- [5] Merdiyan, Meckah. dkk. 2005. *Implementasi Algoritma Run Length, Half Byte dan Huffman Untuk Kompresi file*. Seminar Nasional Aplikasi Teknologi Informasi. ISBN: 979-756-061-6. Yogyakarta.
- [6] Mohammad. A, Haroon. A. 2011. *Data Compression Techniques on Text Files: A Comparison Study*. Jordan. International Journal of Computer Applications (0975-8887) Volume 26-No.5.
- [7] Silalahi, P, Adisantoso. dkk. *Perbandingan Algoritma Huffman Statik dengan Algoritma Huffman Adaptif pada Kompresi Data Teks*. Institut Pertanian Bogor. Jurnal.
- [8] Purwanto, Heri dan Sari A.K. *Aplikasi Kompresi SMS Teks (Short Message Service) Dengan Menggunakan Algoritma Huffman Kanonik dan LZW (Lempel-Ziv-Welch)* <http://www.herpur.web.ugm.ac.id/file/kompresi-SMS.pdf>. Akses tanggal 12 Maret 2012.
- [9] Putra, Darma. 2010. *Pengolahan Citra Digital*. Yogyakarta: Andi
- [10] Raharjo, Budi dkk. 2007. *Tuntunan Pemrograman Java untuk Handphone*. Bandung: Informatika.
- [11] Rozidi, Imron. 2009. *Membuat Sendiri SMS Gateway (ESME) Berbasis Protokol SMPP*. Yogyakarta: Andi.
- [12] Suarga. 2006. *Algoritma dan pemrograman*. Yogyakarta: Andi.
- [13] Sutedjo, B. 2006. *Perencanaan dan Pembangunan Sistem Informasi*. Yogyakarta: Andi.
- [14] Vardiansyah, Dani. 2004. *Pengantar Ilmu Komunikasi*. Bogor: Ghalia Indonesia.
- [15] Wahyudi, Bambang. 2004. *Pengantar Struktur Data & Algoritma*. Yogyakarta: Andi.
- [16] Whitten, Jeffery L. dkk. 2004. *Metode Desain dan Analisis Sistem*. Edisi 6. Terjemahan oleh Tim Penerjemah ANDI. Yogyakarta: Andi.
- [17] Yuku. 2007. *Kompresi Teks SMS*. [http://www.kejut.com/kompresi\\_sms](http://www.kejut.com/kompresi_sms). Akses Tanggal 24 Maret 2012.