

# Implementasi Kode Huffman dalam Aplikasi Kompresi Teks pada Layanan SMS

Nadhira Ayuningtyas (13506048)

Jurusan Teknik Informatika ITB, Jalan Ganesha 10, Bandung, email: if16048@students.if.itb.ac.id

**Abstract** – Makalah ini membahas kompresi teks dalam layanan SMS menggunakan algoritma Huffman. Layanan SMS merupakan cara pengiriman pesan singkat dalam waktu cepat dengan harga relative murah. Pesan singkat ini akan dikirimkan dengan kapasitas maksimal 140 byte untuk setiap pesamya. Untuk mengatasi kesulitan penyampaian pesan dengan jumlah karakter banyak ataupun karakter dengan bit besar, dikembangkan suatu aplikasi untuk melakukan kompresi teks dengan menggunakan algoritma Huffman.

Aplikasi kompresi SMS merupakan serangkaian proses yang berupa kompresi, pengiriman, penerimaan dan penampilan. Proses kompresi itu sendiri merupakan suatu proses encoding yang akan menjadikan suatu data atau teks memiliki bit lebih kecil dari biasanya.

Proses encoding dengan kode Huffman merupakan kode yang dibentuk berdasarkan frekuensi kekerapan yang muncul untuk suatu karakter. Karakter dengan kekerapan lebih besar akan memiliki kode yang lebih pendek dibandingkn karakter dengan kekerapan kecil. Pembentukan kode Huffman dimulai dengan membentuk suatu pohon Huffman yang akan membantu dalam pembuatan kode Huffman. Proses ini dilanjutkan dengan melakukan proses encoding terhadap karakter tersebut sesuai dengan kode Huffman yang ada. Setelah tersampaikan, proses decoding dilakukan agar pesan dapat dibaca kembali.

**Kata Kunci:** Kode Huffman, SMS, Encoding, Decoding, Kompresi, Aplikasi

## 1. PENDAHULUAN

Komunikasi merupakan suatu proses penyampaian informasi yang terjadi dari suatu pihak ke pihak lain. Hal ini menjadi dalam kehidupan. Proses komunikasi senatiasa terjadi setiap waktu. Sebagai perangkat komunikasi dikembangkan dan digunakan untuk menunjang tercapainya proses tersebut.

Telepon genggam, salah satu perangkat komunikasi yang saat ini menjadi kebutuhan primer bagi sebagian manusia. Berbagai macam fitur terdapat dalam telepon genggam, salah satunya adalah

layanan SMS (*short message service*). SMS merupakan layanan pengiriman pesan singkat dalam jaringan komunikasi.

Penggunaan SMS ini akan memudahkan para pemakainya untuk saling bertukar informasi. Melalui SMS, seseorang akan dapat mengirimkan pesan singkat dalam waktu cepat dengan harga yang relatif murah, tanpa dibatasi oleh jarak maupun waktu. Sehingga, saat ini SMS menjadi metode komunikasi yang banyak digunakan masyarakat.

Namun, disamping kemudahan-kemudahan yang diberikan, layanan SMS ini membatasi jumlah karakter yang dapat dikirimkan oleh penggunanya. Sebuah pesan yang dikirimkan melalui SMS, memiliki kapasitas maksimal 140 *byte*. Hal ini menyebabkan seseorang yang ingin mengirimkan pesan yang cukup panjang, terdiri dari sejumlah karakter, atau memiliki karakter yang berukuran besar, akan mengalami kesulitan. Walaupun tersampaikan, pesan harus dirangkai menjadi sejumlah SMS sesuai kapasitas maksimalnya.

Beberapa cara dapat dilakukan untuk mengatasi hal tersebut, salah satunya dengan melakukan kompresi. Kompresi merupakan proses *encoding* data menggunakan jumlah bit yang lebih kecil, sehingga bit yang lebih kecil tersebut dapat merepresentasikan informasi yang sama. Kompresi teks atau data akan memperkecil jumlah memori yang digunakan dan mempercepat tercapainya proses informasi.

Kode Huffman, merupakan algoritma yang cukup populer dan banyak digunakan untuk melakukan kompresi teks maupun data. Kode Huffman ini memiliki prinsip berdasarkan metode statistik, yang akan menghitung kekerapan dari karakter yang muncul.

Makalah ini akan membahas penggunaan kode Huffman dan kaitannya dengan kompresi suatu .data atau teks, khususnya dalam layanan SMS. Tujuan pembauatan makalah ini untuk mengetahui implementasi penggunaan kode Huffman dalam pembuatab aplikasi kompresi teks pada layanan SMS.

## 2. KODE HUFFMAN

### 2.1. Sejarah Kode Huffman

Kode Huffman menggunakan tabel dengan variasi kode panjang untuk melakukan *encoding* dari sebuah simbol. Tabel variabel kode panjang tersebut telah dibuat terlebih dahulu secara terpisah berdasarkan nilai kekerapan munculnya suatu simbol. Metode ini ditemukan oleh David A. Huffman ketika ia melakukan studi Ph.D di MIT. Kode ini dipublikasikan [ada tahun 1952 pada tulisannya yang berjudul “A Method for the Constuction of Minimim-Redudancy Codes”][2].

### 2.2. Pembentukan Kode Huffman

Kode Huffman menggunakan metode spesifik untuk merepresentasikan setiap simbol yang menghasilkan suatu kode prefix. Kode prefix ini merupakan sekumpulan kode biner yang pada kode ini tidak mungkin terdapat kode prefix yang menjadi awalan bagi kode biner yang merepresentasikan simbol lain. Hal ini akan mencegah timbulnya keraguan dalam proses *decoding*. Dalam kode Huffman, kode biner untuk simbol dengan kekerapan lebih besar akan memiliki kode yang lebih pendek daripada untuk simbol dengan kekerapan lebih kecil.

Membentuk suatu kode Huffman dimulai dengan membuat suatu pohon biner yang disebut pohon Huffman. Pohon ini akan disimpan pada suatu tabel, dengan ukuran yang bergantung pada jumlah dari simbol tersebut. Suatu simpul pada pohon biner dapat berupa simpul daun (simpul yang memiliki jumlah anak nol) ataupun simpul dalam (simpul yang mempunyai anak). Pada awalnya, semua simpul merupakan simpul daun, yang mengandung simbol itu sendiri serta bobotnya (frekuensi kekerapan) dari simbol tersebut dan bisa juga mengandung *link* ke simpul orangtua yang akan memudahkan pembacaan kode (secara terbalik) dimulai dari simpul daun. Pada simpul dalam terdapat bobot dan *link* ke dua simpul anak dan bisa ke simpul orangtua. Sebagai perjanjian, bit ‘0’ akan merepresentasikan anak kiri dan bit ‘1’ akan merepresentasikan anak kanan. Pohon yang telah selesai akan memiliki  $n$  buah simpul daun dan  $n-1$  buah simpul dalam [2].

Sutu pohon Huffman dapat dibentuk dengan cara sebagai berikut :

1. Membuat simpul daun sebanyak sejumlah simbol
2. Memilih dua simbol dengan peluang terkecil dan dikombinasikan sebagai suatu simpul orangtua
3. Membuat simpul yang merupakan simpul orangtua dari dua simpul dengan peluang terkecil
4. Memilih sebuah simpul berikutnya (termasuk

5. Melakukan prosedur yang sama pada dua simbol berikutnya yang memiliki peluang terkecil

Tabel 1 adalah contoh kode ASCII untuk beberapa karakter.

Tabel 1. Kode ASCII

Simbol	Kode ASCII
A	01000001
B	01000010
C	01000011
D	01000100

Maka, rangkaian bit untuk string ‘ABACCDCA’ :  
01000001010000100100000101000011010000110100010001000001

Berdasarkan metode pengkodean ASCII, setiap huruf direpresentasikan dalam 8-bit sehingga untuk merepresentasikan 7 huruf akan membutuhkan  $7 \times 8 = 56$ -bit.

Berikut ini merupakan contoh pengkodean Huffman untuk melakukan kompresi *string*. Tabel 2 akan memperlihatkan tabel kekerapan dan kode Huffman untuk *string* ‘ABACCDCA’.

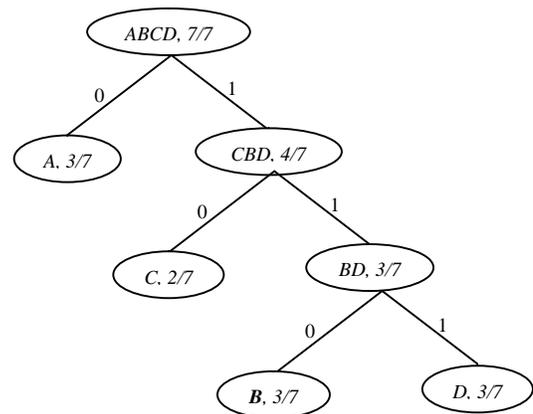
Tabel 2. kekerapan dan kode Huffman *string* ‘ABACCDCA’

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

Sehingga, dengan menggunakan kode Huffman rangkaian bit untuk string ‘ABACCDCA’ :  
0110010101110

Rangkaian ini direpresentasikan dalam 13 bit.

Gambar 1 berikut ini akan menggambarkan Pohon Huffman untuk ‘ABACCDCA’.



Gambar 1. Pohon Huffman untuk *string* ‘ABACCDCA’

### 2.3. Encoding

Proses untuk melakukan pembentukan kode dari suatu data tertentu disebut *encoding*. Dalam hal ini, kode Huffman akan terbentuk sebagai suatu kode biner. Kode Huffman didapatkan dengan membaca setiap kode dari daun simbol tersebut hingga ke akarnya. Ketika suatu kode Huffman telah dibentuk, suatu data dapat akan mudah di *encode* dengan mengganti setiap simbol menggunakan kode yang telah dibentuk. Contoh tabel kode Huffman yang telah di-*encoding* dapat dilihat pada Tabel 2.

### 2.4. Decoding

*Decoding* merupakan proses yang mengembalikan suatu data dari suatu kode tertentu. Proses *decoding* ini merupakan kebalikan dari proses *encoding*.

Terdapat dua cara yang cukup cepat untuk melakukan *decoding* simbol :

#### 1. Membaca dari pohon Huffman

Hal ini dapat dilakukan dengan membaca sebuah bit dari kode binernya dan menelusuri hingga sampai pada simpul daun yang mengandung simbol tersebut untuk setiap bitnya. Ketika suatu bit sampai pada daun suatu pohon, suatu simbol yang terkandung dalam daun tersebut ditulis untuk *decoded* data tersebut dan mengulanginya kembali dari akar pohon tersebut.

#### 2. Menggunakan tabel kode Huffman

*Decoding* cara ini dilakukan dengan menyimpan setiap kode pada suatu tabel yang terurut berdasarkan panjang kode dan mencari kesamaan dari setiap bit yang dibaca.

Metode membaca dari pohon Huffman lebih cepat dilakukan untuk menangani kasus terburuk *encoding*, yaitu ketika seluruh simbolnya memiliki panjang 8-bit. 8-bit kode akan menunjuk pada simbol dengan kedalaman 8, tetapi pencarian kode biner untuk 256 simbol adalah  $O(\log_2(256))$  atau sekitar 16 langkah[3].

### 2.4. Kompleksitas Algoritma Huffman

Dalam melakukan satu kali proses iterasi untuk menggabungkan dua buah pohon dengan frekuensi terkecil ada sebuah akar, waktu yang dibutuhkan adalah  $O(\log n)$ . Proses tersebut akan dilakukan sebanyak  $n$  kali hingga terbentuk sebuah pohon Huffman. Sehingga, waktu untuk kompleksitas waktu algoritma Huffman adalah  $O(n \log n)$ .

## 3. SHORT MESSAGE SERVICE (SMS)

### 3.1. Deskripsi Umum

Layanan pesan singkat atau *short message service* adalah sebuah layanan yang digunakan pada suatu telepon genggam untuk mengirim atau menerima pesan-pesan pendek. Awalnya, SMS dirancang

sebagai bagian daripada jaringan GSM, sekarang hal ini telah berkembang ke jaringan bergerak lainnya termasuk jaringan UMTS.

Sebuah pesan SMS memiliki kapasitas maksimal 140 *byte*, untuk karakter 8-bit sebuah pesan bisa memuat 140 karakter, untuk karakter 7-bit sebanyak 160 karakter, dan untuk karakter 16-bit seperti bahasa Jepang, Bahasa Mandarin dan Bahasa Korea yang memakai Hanzi (Aksara Kanji / Hanja) dapat dimuat 70 karakter. Selain kapasitas 140 *byte* terdapat pula data-data lain. Untuk pesan yang lebih dari 140 *byte*, seorang pengguna akan dihitung mengirim pesan lebih dari satu [4].

SMS berkembang menjadi salah satu layanan yang cukup diminati oleh penggunanya. Berbagai keunggulan yang mendukung hal tersebut, yaitu :

- Memiliki harga yang relatif murah, bahkan sebagian perusahaan secara cuma-cuma memberikan layanan ini untuk menarik lebih banyak pengguna
- *Deliver Oriented Service*, pesan akan selalu diusahakan untuk dikirimkan ke tujuan. Jika suatu tujuan sedang tidak aktif, pesan akan disimpan di SMSC (*short message service centre*) sebagai server dan akan dikirim sesegera setelah nomor tujuan aktif. Pesan juga akan tetap terkirim walaupun pengguna sedang melakukan telepon karena transmisi SMS menggunakan kanal *signaling* bukan kanal suara.
- Dapat dikirim ke berbagai tujuan / penerima pada saat bersamaan
- Dapat dikirim ke berbagai jenis penerima, seperti email, IP, maupun berbagai aplikasi lain.
- Memiliki beberapa macam kegunaan bila dilakukan integrasi dengan berbagai aplikasi, seperti *chatting*, *voting*, kuis, reservasi, informasi tertentu, dan sebagainya.

Tabel 3 akan memperlihatkan beberapa fakta mengenai teknologi SMS.

Tabel 3. SMS *technology fact sheets*

<i>Standard :</i>	GSM (ETS 03.40)
<i>Transport technology:</i>	GSM Signaling Path, GPRS
<i>Transport protocol:</i>	Short Message Protocol
<i>Addressing scheme:</i>	MSISDN
<i>Message description language:</i>	SMS PDU
<i>User data length:</i>	140 <i>byte</i>
<i>Basic character set:</i>	7-bit SMS, 8-bit SMS, UCS2

### 3.2. Perkembangan Teknologi Messaging

SMS mulai dikenal pada era teknologi *wireless* generasi kedua (2G), saat komunikasi data pada

telekomunikasi wireless dapat dilakukan. Di Eropa, SMS diperkenalkan pada tahun 1991 saat GSM mulai digunakan. Perkembangan teknologi *messaging* akan ditunjukkan pada Tabel 4.

Tabel 4. Perkembangan Teknologi *Messaging*

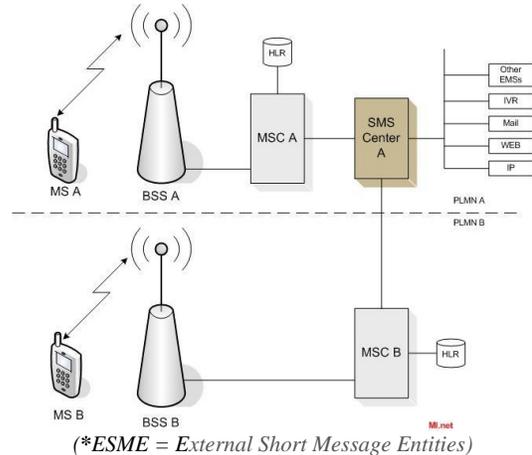
Gen	Teknologi	Service	Ket
1	AMPS	<ul style="list-style-type: none"> <li>• <i>Circuit-switched wireless analog voice</i></li> <li>• <i>Limited system capacity &amp; capability</i></li> <li>• <i>No data</i></li> </ul>	tidak bisa SMS
2	TDMA CDMA GSM	<ul style="list-style-type: none"> <li>• <i>Circuit-switched wireless digital voice data</i></li> <li>• Security lebih baik</li> <li>• Kapasitas lebih besar</li> <li>• Support komunikasi data</li> </ul>	bisa SMS
2.5	GPRS CDMA200-1x EDGE	<ul style="list-style-type: none"> <li>• <i>circuit-switched wireless digital voice</i></li> <li>• diperkenalkannya <i>packet-switched data services</i></li> <li>• Kecepatan &amp; Kapasitas lebih baik</li> </ul>	Migrasi ke 3G bisa SMS, EMS, MMS
3	WCDMA CDMA200-MX UMTS	<ul style="list-style-type: none"> <li>• <i>packet-switched wireless</i></li> <li>• voice dan data</li> <li>• <i>encrypts, high-speed multi-media</i></li> </ul>	bisa SMS, EMS, MMS 3G platform

SMS berkembang menjadi EMS (*Enhanced Message Service*), dengan jumlah karakter yang bisa dikirimkan menjadi lebih banyak serta dapat digunakan untuk mengirim pesan berupa non-karakter (gambar sederhana) dan rekaman suara.

Saat teknologi, *packet switch* seperti GPRS mulai digunakan, *service* pengiriman pesan berkembang menjadi tidak hanya sebatas teks melainkan mulai digunakan dalam bentuk gambar, suara maupun video (multimedia), layanan ini dikenal dengan MMS (*multimedia message service*) [5].

### 3.3. Jaringan SMS

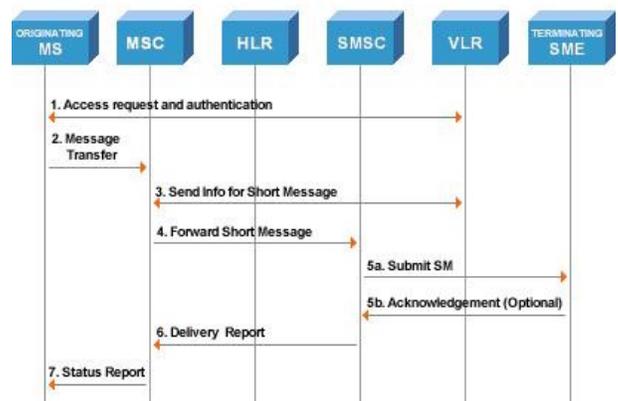
Salah satu contoh arsitektur jaringan GSM dengan SMS center (SMSC) di dalamnya akan ditunjukkan pada Gambar 2.



Gambar 2. Jaringan GSM dengan SMSC

SMS, dapat mentransmisikan pesan singkat dari dan ke *Mobile Subscriber* (MS). Pengiriman pesan singkat (SMS) ini dimungkinkan dengan adanya sebuah SMSC. Secara umum SMSC berfungsi menerima SMS yang dikirim, menyimpannya untuk sementara, dan mem-forward (mengirimkan) SMS tersebut ke *mobile subscriber* (MS) ataupun ESME tujuan.

Gambar 3 akan menunjukkan alur skenario pengiriman SMS MO dari MS ke ESME (SMS *Originating*).



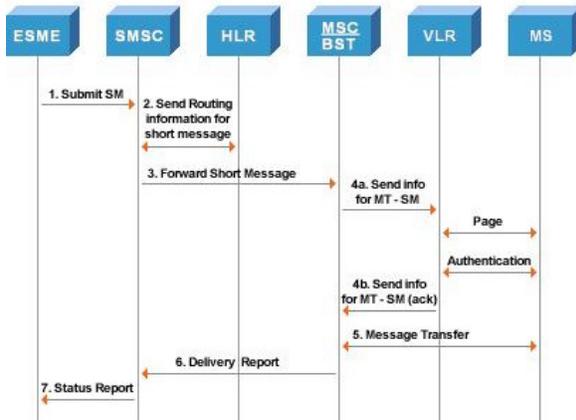
Gambar 3. Skenario SMS MO (*Mobile Originating*)

Tahapan pengiriman tersebut :

1. MS diaktifkan dan ter-register ke *network*-nya.
2. MS mengirimkan SMS ke MSC.
3. MSC berkomunikasi dengan VLR untuk memverifikasi bahwa pesan yang dikirimkan sesuai dengan *supplementary service* yang ada dan tidak MS tidak sedang dalam keadaan diblok untuk mengirimkan SMS.
4. MSC mengirimkan SMS ke SMSC dengan menggunakan operasi *forwardShortMessage*.

5. SMSC meneruskan SMS ke SME. Secara optional, SMSC dapat juga menerima *acknowledgment* bahwa SMS telah diterima SME.
6. MSC memberitahukan MSC bahwa SMS telah dikirimkan ke SME.

Alur sekenario SMS yang diterima MS dari ESME (SMS *Terminating*) ditunjukkan pada Gambar 4.



Gambar 4. Skenario MS dari ESME (SMS *Terminating*)

Tahapan penerimaan tersebut :

1. ESME mengirimkan SMS ke SMSC.
2. Setelah menerima SMS, SMSC akan berkomunikasi dengan HLR mengetahui status dan lokasi MS.
3. SMSC meneruskan SMS ke MSC.
4. MSC akan menghubungi VLR untuk mengetahui informasi dari MS. Dalam tahap ini termasuk juga proses autentikasi MS.
5. Jika MS dalam keadaan aktif dan tidak diblock, MSC mentransfer SMS ke MS.
6. MSC akan mengirimkan informasi delivery message ke SMSC
7. Jika diminta oleh ESME, SMSC akan mengirimkan status report ke ESME.

Transmisi pesan singkat antara SMSC dan *handset* menggunakan *Mobile Application Part* (MAP) dari protocol SS7. Pesan dikirimkan dengan MAP modan *mt-Forward operations*. Jaringan inilah yang memiliki batas *signaling protocol 140 byte* [5].

#### 4. KOMPRESI SMS DENGAN KODE HUFFMAN

Terdapat beberapa aplikasi kompresi SMS yang telah dikembangkan untuk melakukan kompresi SMS, diantaranya *SMS Teks Compression* dan *Costfix*.

#### 4.1. Kompresi SMS dengan Program *SMS Teks Compression*

##### 4.1.1. Prinsip Dasar

Aplikasi ini dibuat dengan menggunakan Java Microedition 2.0. Menggunakan daftar kode Huffman yang telah ada dan menambahkan awalan diikuti dengan badannya. Huruf kecil, akan lebih sering muncul, diberi awalan 0. Huruf besar yang memiliki kekerapan lebih kecil, diberi awalan 100. Angka dan beberapa simbol 111. Tanda baca yang umum digunakan memiliki kode awalan 110, spasi memiliki kode 1011 dan khusus untuk karakter ‘e’ menjadi 1010. Karakter sisanya (simbol langka dan karakter *Unicode*) diberi awalan 1111110 yang diikuti kode karakter aslinya 16-bit. Awalan dan badan digabungkan akan menghasilkan kode ubahan. Tabel 5 berikut akan memperlihatkan beberapa kode Huffman untuk aplikasi ini [6].

Tabel 5. Kode Huffman

Karakter	Awalan	Badan	Ubahan	
a	0	0000	00000	
b		00101	000101	
c		1101	01101	
d		01011	001011	
f		11001	011001	
g		10101	010101	
A		100	0000	1000000
B	00101		10000101	
C	1101		1001101	
D	01011		10001011	
F	11001		10011001	
G	10101		100101010	
<space>	101		1	1011
e		0	1010	
!	110	0011	1100011	
“		0110	1100110	
‘		0100	1100100	
,		0001	1100001	
-		0101	1100101	
.		0000	1100000	
?		0010	1100010	
E		0111	1100111	
0		111	0000	1110000
1			0001	1110001
2	0010		1110010	
3	0011		1110011	
7	0111		1110111	
8	1000		1111000	
9	1001		1111001	
(	11010		11111010	
)	11011		11111011	
/	1011		1111011	
:	1100		1111100	
@	1010		1111010	
sis	1111110		1111110	

Sebagai perbandingan, berikut Tabel 6 yang menunjukkan kode ASCII standar.

Tabel 6. Tabel ASCII

Karakter	Kode	Karakter	Kode
0	0110000	@	1000000
1	0110001	?	0111111
2	0110010	A	1000001
a	1100001	B	1000010
b	1100010	C	1000011

#### 4.1.2. Penggunaan Aplikasi

Pada karakter 7-bit, satu buah SMS dapat mempunyai maksimal 160 karakter dengan sehingga total memori yang dipakai adalah 1120-bit. Dengan aplikasi kompresi ini, jumlah bit yang dipakai untuk suatu pesan lebih pendek daripada karakter ASCII biasa. Karakter yang memiliki ubahan kurang dari 7-bit yang akan memberikan keuntungan kompresi [6].

Sebagai contoh, penulisan teks *Hello* :

Tabel 7. Karakter *Hello* dan ubahannya

Karakter	Ubanan
H	1001000
e	1010
l	01001
l	01001
o	00100
<b>Jumlah bit</b>	<b>24</b>

Tabel 7 memeperlihatkan jumlah bit ubahan pada karakter yang membentuk kata *Hello* adalah 24-bit, sementara jika tanpa melakukan kompresi jumlah bit yang digunakan akan sebanyak  $5 \times 7\text{-bit} = 35\text{-bit}$ . Sehingga dengan kompresi ini untuk penulisan *Hello* akan menghemat sekitar 68,57% memori.

Penggunaan aplikasi kompresi SMS ini ditampilkan oleh beberapa contoh gambar berikut.

- a. Tampilan aplikasi  
Gambar 5 akan menampilkan tampilan aplikasi Kompresi SMS ini.



Gambar 5. Tampilan Aplikasi

- Kolom 1 : Nomor Tujuan  
Diisi dengan nomor penerima SMS.
- Kolom 2 : *Port*  
Seperti internet, SMS memiliki port tertentu. Kolom *port* ini dapat diisi atau tidak. Pada contoh diatas *port* ini diisi oleh nomor 7575, SMS akan dikirim ke `://+62810121234:7575`, jika kosong SMS akan dikirim ke `://+62810121234`. Dengan *port*, program yang sama dapat dipasang pada ponsel penerima dan program ini akan menangkap seluruh SMS yang diterima melalui *port* tersebut. Namun, dengan penggunaan *port* ini akan mengurangi 4 *byte* kapasitas SMS menjadi 136 *byte*.
- Kolom 3 : Isi  
Berisi teks yang akan dikirimkan.

- b. Tampilan isi (kolom 3)  
Gambar 6 berikut akan menampilkan isi SMS yang akan dikirimkan.



Gambar 6. Kolom Isi

Pada bagian sebelah kanan isi terlihat angka 186 yang merupakan banyaknya karakter 7-bit dan (2) merupakan jumlah SMS ketika tidak dilakukan kompresi. Setelah kompresi memori yang digunakan menjadi 124 *byte* dan (1) jumlah SMS setelah dikompresi. 76% adalah ukuran kompresi dibandingkan dengan aslinya.

- c. SMS diterima  
Gambar 7 menampilkan SMS yang diterima setelah dikompresi (tahap percobaan).



- **character sum** : jumlah karakter yang akan dikirimkan.
- **Compression ratio** : persentase rasio kompresi SMS, dapat dihitung dengan rumus :  
(bit hasil kompresi/bit tidakdikompresi) x 100%
- **increase percentage**: persentase peningkatan karakter / banyaknya karakter lebih yang dapat dikirimkan. Hal ini dapat dihitung dengan rumus:  
jumlah karakter lebih/(160\*jumlah pesan) x 100%

Untuk mengirim pesan, pengguna menekan tombol *send*, sesaat kemudian akan ditampilkan info pesan terkirim (gambar kanan) dan secara otomatis akan kembali ke menu utama.

Pada contoh tersebut diperlihatkan pengiriman SMS 201 karakter dengan satu kali SMS. Jika tidak menggunakan aplikasi ini, untuk mengirimkan 201 karakter pada aplikasi telepon genggam biasa akan menjadikan 2 SMS. Hasil kompresi pada setiap pesan akan berbeda-beda tergantung kepada pesan yang diketikkan. Untuk mendapatkan hasil kompresi yang maksimal sebaiknya lebih banyak menggunakan huruf kecil [7].

## 5. KESIMPULAN

Kompresi data atau teks dalam layanan SMS menggunakan kode Huffman merupakan salah satu cara yang efektif untuk melakukan pengiriman pesan singkat yang memiliki cukup banyak karakter atau menggunakan karakter dengan bit besar.

Aplikasi kompresi SMS yang dibuat dengan menerapkan algoritma Huffman dapat melakukan kompresi pesan dua hingga tiga kali lebih kecil. Proses ini diawali dengan pembuatan pohon Huffman untuk karakter yang terdapat dalam pesan. Selanjutnya, karakter-karakter pada pesan tersebut akan di-*encoding* menjadi kode Huffman. Pada saat penerimaan, pesan akan di-*decoding* kembali.

## DAFTAR REFERENSI

- [1] Munir, Rinaldi. Diktat Kuliah IF2153 Matematika Diskrit. Program Studi Teknik Informatika, Institut Teknologi Bandung. 2006.
- [2] Wikipedia, The Free Encyclopedia, *Huffman Coding*. 2007.  
[http://en.wikipedia.org/wiki/Huffman\\_coding](http://en.wikipedia.org/wiki/Huffman_coding).  
Waktu akses : Rabu, 26 Desember 2007 pukul 23.00 WIB.
- [3] Diapperstein, Michael. *Huffman Code Discussion and Implementation*. 2007.  
<http://michael.dipperstein.com/huffman/index.html>.  
Waktu akses : Rabu, 26 Desember 2007 pukul

23.00 WIB.

- [4] Wikipedia, The Free Encyclopedia, *Short Message Service*. 2007.  
[http://en.wikipedia.org/wiki/Short\\_message\\_service](http://en.wikipedia.org/wiki/Short_message_service).  
. Waktu akses : Rabu, 26 Desember 2007 pukul 23.00 WIB.

- [5] Riswan. *Mengenal SMS*. Mobile Indonesia [dot] net. 2006.  
<http://mobileindonesia.net/2006/08/01/short-message-service-sms>. Waktu akses : Rabu, 26 Desember 2007 pukul 23.00 WIB

- [6] Yuku. *Kompresi Teks SMS*. Kejut. 2007.  
<http://www.kejut.com/kompresisms>. Waktu akses :  
Rabu, 26 Desember 2007 pukul 23.00 WIB

- [7] Farid. *SMS Murah untuk Semua Merk HP..* 2007.  
<http://www.costfix.net/2007/12/19/costfix/sms-murah-untuk-semua-merk-hp-dan-operator-selular/>. Waktu akses : Rabu, 26 Desember 2007 pukul 23.00 WIB