

# KODE HUFFMAN UNTUK KOMPRESI SMS

Dini Lestari Tresnani (13508096)

Jurusan Teknik Informatika ITB  
Jalan Ganesha 10, Bandung  
e-mail: [neng.kucing@gmail.com](mailto:neng.kucing@gmail.com); [if18096@students.if.itb.ac.id](mailto:if18096@students.if.itb.ac.id)

## ABSTRAK

Teknologi informasi yang canggih pada era modern seperti ini selalu dipandang dari 2 sisi dasar. Yaitu kecepatan akses, serta kapasitas memori yang dibutuhkan. Semakin cepat akses untuk suatu teknologi, dan semakin kecil kapasitas memori yang dibutuhkan, maka akan semakin canggih teknologi tersebut. Hal ini dikarenakan jika semakin besar suatu memori yang dibutuhkan, maka akses yang dibutuhkan menurun.

Dalam masalah SMS (*Short Message Service*), kecepatan akses dan kapasitas memori adalah sesuatu yang bersifat statik. Hal ini berhubungan dengan media perantara yang digunakan untuk mengirim SMS (BTS, ponsel, dll.) tidak dapat diganti dan di-upgrade begitu saja. Karena masalah statik ini, dibutuhkan suatu cara agar SMS yang dikirim tidak memiliki ukuran yang besar sehingga dapat diterima dengan cepat. Salah satu caranya adalah dengan kompresi menggunakan kode huffman.

**Kata kunci:** kecepatan akses, kapasitas memori, SMS, kode huffman.

## 1. PENDAHULUAN

Kemajuan teknologi yang begitu pesat memicu kebutuhan informasi yang sangat besar. Hal ini menyebabkan dunia membutuhkan suatu metode yang dapat digunakan untuk menyebarkan informasi secara cepat, dan tidak memakai memori terlalu besar.

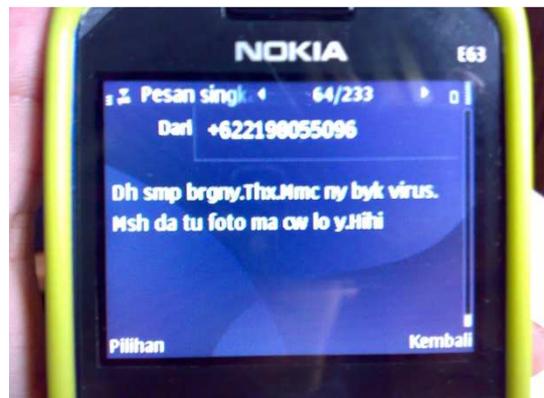
Salah satu metode yang diciptakan adalah SMS (*Short Message Service*) yaitu suatu layanan untuk mengirim ataupun menerima pesan-pesan pendek yang berbasis tulisan. Dalam penerimaan dan pengiriman pesan pendek ini dibutuhkan sebuah media perantara yaitu telepon genggam, yang pada era modern seperti ini hampir setiap orang memilikinya.

Metode SMS ini merupakan terobosan yang sangat inovatif, karena tidak dibutuhkan waktu yang lama (bahkan hanya sekitar 2-3 detik sampai pesan terkirim dan diterima oleh penerima) dan harga untuk SMS relatif

murah (sekitar Rp 15,- hingga Rp 350,-) dibandingkan metode pengiriman pesan yang lain.

Namun kelemahan dari metode SMS adalah sebuah pesan SMS maksimal terdiri dari 140 bytes, dengan kata lain sebuah pesan bisa memuat 140 karakter 8-bit, 160 karakter 7-bit atau 70 karakter 16-bit untuk bahasa Jepang, Bahasa Mandarin dan Bahasa Korea yang memakai *Hanzi* (Aksara *Kanji/Hanja*). Dalam melakukan pengiriman pesan SMS seorang pengguna dapat mengirim pesan lebih dari 140 byte, tetapi untuk itu seorang pengguna harus membayar lebih dari sekali. Hal ini terjadi karena pesan yang dikirimkan terdiri lebih dari satu halaman sehingga proses pengiriman pesan akan dilakukan sebanyak jumlah halaman yang ada, jumlah halaman sesuai dengan isi SMS yang diketikkan.

Salah satu solusi dari permasalahan tersebut adalah dengan melakukan kompresi pada pesan yang dikirimkan. Kompresi ini bisa berarti pengirim mempersingkat panjang kata dengan singkatan-singkatan yang lazim digunakan (Lihat Gambar 1). Solusi lain yang bisa digunakan adalah dengan melakukan proses *encoding* terhadap pesan singkat yang dikirim menggunakan jumlah bit yang lebih sedikit namun memiliki informasi yang sama, kemudian pesan yang telah di-*encoding* itu dikirimkan dan dibaca oleh ponsel penerima dengan menggunakan proses *decoding* sehingga tampilan pesan akan sama dengan yang dimaksud.



Gambar 1. Kompresi dengan cara menyingkat kata yang hendak dikirimkan

## 2. SMS (Short Message Service)

SMS (*Short Message Service*) secara umum dapat diartikan sebagai sebuah service yang memungkinkan ditransmisikannya pesan text pendek dari dan ke telepon genggam, fax, mesin, ataupun IP address. Disebut pesan teks pendek karena pesan yang dikirimkan hanya berupa karakter.

Teknologi yang mendukung SMS antara lain adalah GSM, TDMA dan CDMA. Dengan didukung oleh ketiga teknologi ini, SMS telah menjadi layanan data bergerak yang bersifat universal.

Kegunggulan dari teknologi SMS ini adalah:

- Harganya murah.
- Merupakan "deliver oriented service", artinya pesan akan selalu diusahakan untuk dikirimkan ke tujuan. Jika suatu saat nomor tujuan sedang tidak aktif atau di luar servis area, maka pesan akan disimpan di SMSC server dan akan dikirimkan segera setelah nomor tujuan aktif kembali. Pesan juga akan tetap terkirim ke tujuan walaupun nomor tujuan sedang melakukan layanan suara.
- Dapat dikirim ke banyak penerima sekaligus pada saat yang bersamaan.
- Pesan dapat dikirimkan ke berbagai jenis tujuan, seperti e-mail, IP ataupun aplikasi lain.
- Kegunaannya banyak. SMS dapat digunakan untuk berbagai macam keperluan seperti kuis, voting, chatting, reservasi, request informasi, sensus/survey, dan lain-lain. [2]

### 2.1 Perkembangan SMS

Layanan SMS ini tidak langsung ada pada saat telepon genggam pertama hadir di masyarakat. Hal ini dikarenakan pada saat telepon genggam pertama hadir, jaringan yang digunakan masih menggunakan teknologi AMPS.

Teknologi AMPS merupakan sistem komunikasi seluler pertama kali atau sering disebut generasi pertama (1G) diperkenalkan pada tahun 1983 yang hanya mentransmisikan voice dengan frekuensi 800-900 MHz dan 400-500 MHz serta masih menggunakan modulasi analog. Lebar channel 30 kHz dengan kecepatan data (data rate) 5-10 kbps. Dasar sistem analog ini menggunakan FDMA (*Frekuensi Divison Multiple Access*) dimana perbedaan antara satu saluran dengan saluran yang lain dilakukan dengan perbedaan frekuensi. Keterbatasan AMPS seperti kapasitas komunikasi yang kecil, keterbatasan spektrum, tidak ada ruang untuk perkembangan spektrum, komunikasi data yang buruk, minimal privacy dan tidak dapat memproteksi kecurangan/penipuan membuat AMPS ditinggalkan. [6]

Seiring dengan perkembangannya, dikembangkan suatu teknologi yang lebih maju dari AMPS sehingga dapat mengirimkan tidak hanya suara tetapi sebuah pesan

singkat juga. Teknologi ini disebut sebagai teknologi 2G (generasi kedua) dan pada saat teknologi muncul, layanan SMS pertama kalinya ada.

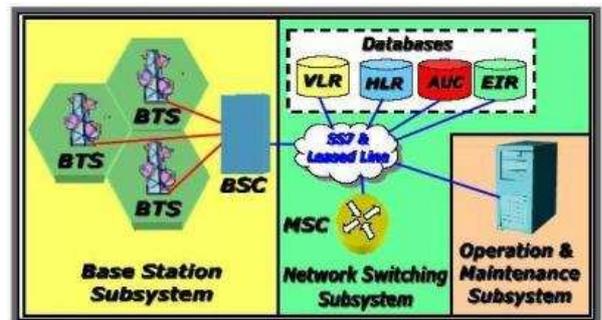
Sistem telekomunikasi seluler pada generasi kedua menggunakan teknologi digital. Sistem telekomunikasi seluler pada generasi kedua menggunakan basis teknologi TDMA dan CDMA. Sistem yang menggunakan TDMA adalah IS-136 dan GSM. Rancangan utama dari sistem ini adalah untuk mendukung aliran suara berbentuk circuit-switched, pada perkembangannya sistem ini mampu pula mendukung paket data circuit-switched dan layanan pesan dengan menggunakan Short Message Service (SMS). Teknologi lainnya pada 2G adalah IS-95 atau Narrowband CDMA dan CDMA.

SMS pertama kali dikirimkan lewat jaringan GSM Vodafone di Inggris pada bulan Desember tahun 1992. Bunyi tulisan yang dikirimkan adalah "Merry Christmas".

Setelah teknologi 2G ini, berkembang lagi teknologi-teknologi lainnya yaitu 2.5G, 3G (saat ini banyak digunakan di Indonesia), dan 4G (baru diadakan di Jepang).

### 2.2 Mekanisme Kerja SMS

Skema jaringan GSM yang kini banyak beredar di Indonesia sebagai teknologi yang digunakan untuk SMS dapat dilihat pada Gambar 2.



Gambar 2. Skema jaringan GSM

Elemen yang terdapat dalam sebuah jaringan operator seluler yang berfungsi besar dalam penyampaian SMS ada dua elemen, yaitu SMC dan MSC.

Short Message Center (SMC) adalah sebuah entitas yang bertugas untuk menerima dan meneruskan pesan dari dan ke ponsel seluler. SMC dibangun oleh beberapa Short Message Entity (SME) yang dapat diletakkan dalam sebuah jaringan atau telepon selular.

Mobile Switching Center (MSC) adalah sebuah entitas dalam sebuah jaringan operator seluler yang bertugas mengendalikan koneksi antar ponsel atau antara ponsel dengan jaringan operator selular. Gateway Mobile Switching Center (GMSC) adalah sebuah gerbang MSC yang juga dapat menerima pesan dan berupa sebuah sistem kontak yang berhubungan dengan jaringan lain. Dalam menerima pesan dari SMC, GMSC menggunakan

jaringan SS7 dalam sistem Home Location Register (HLR).

HLR adalah database utama dalam sebuah jaringan operator selular. Sistem ini memegang kendali atas informasi nomor-nomor ponsel dan juga tentang alur informasi dari setiap nomor ponsel, misalnya informasi atas wilayah jangkauan. Visitor Location Register (VLR) berkorespondensi terhadap setiap MSC. VLR berisi informasi tentang identitas ponsel. Dengan bantuan VLR, MSC dapat meneruskan informasi pesan pendek kepada Base Station System (BSS), dimana kemudian BSS akan meneruskannya ke ponsel penerima. [4]

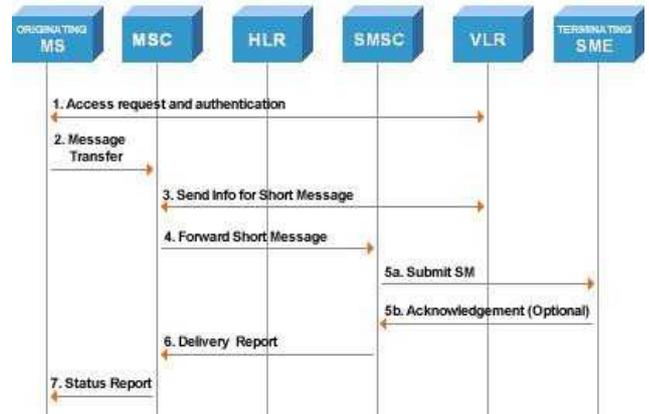
*External Short Message Entities (ESME)* adalah device selain MS yang dapat berfungsi untuk menerima atau mengirim SMS. Pada umumnya ESME dipakai untuk menciptakan layanan yang lebih beragam kepada pelanggan ataupun untuk meningkatkan performance jaringan telekomunikasi dari operator telekomunikasi *wireless* yang bersangkutan. ESME dapat berupa antara lain:

- VMS (*Voice Mail Service*). VMS berfungsi untuk menerima, menyimpan dan memainkan/memperdengarkan *voicemail* (pesan suara) yang ditujukan kepada subscriber. Pesan suara ini direkam ketika ada orang yang hendak menghubungi subscriber tertentu, tapi subscriber tersebut dalam keadaan tidak aktif, sibuk, ataupun di luar coverage, sehingga si pemanggil tidak dapat tersambung dengannya. Pada saat ini, pemanggil dapat meninggalkan pesan berupa suara dan akan disimpan di VMS. Pada suatu saat nanti, apabila subscriber yang akan dipanggil tadi sudah aktif kembali atau sudah *idle*, dia akan menerima notifikasi bahwa ada pesan suara untuknya, dan dia dapat mendengarkan pesan suara tersebut dengan merequest VMS untuk memperdengarkannya di handsetnya.
- Web. Dengan teknologi internet yang berkembang pesat, MS dapat mengirimkan SMS dan langsung ditampilkan dalam suatu halaman web.
- E-mail. MS dapat juga mengirimkan sms ke suatu alamat e-mail dan akan diterima sebagai sebuah e-mail.
- Aplikasi content lainnya. Dengan perkembangan teknologi IT, terutama di bidang software, maka SMS dapat digunakan untuk berbagai macam tujuan, seperti: voting, reservasi tiket, registrasi anggota suatu komunitas, games/kuis, survey, bahkan memungkinkan juga untuk digunakan sebagai sarana pemilu (pemilihan umum). Semua itu dimungkinkan karena adanya aplikasi-aplikasi content yang mendukung.

Bila sebuah SMS dikirimkan dari MS A ke MS B, maka SMS itu akan diteruskan oleh BSS ke MSC dan kemudian ke SMSC. SMSC berfungsi mengirim SMS tersebut ke

MS B. Untuk keperluan ini, smsc harus tahu bagaimana status subscriber (aktif/tidak aktif), dimana lokasi MS B berada. Informasi-informasi mengenai MS B ini didapat dari HLR.

Skenario pengiriman SMS dapat dilihat di Gambar 3.

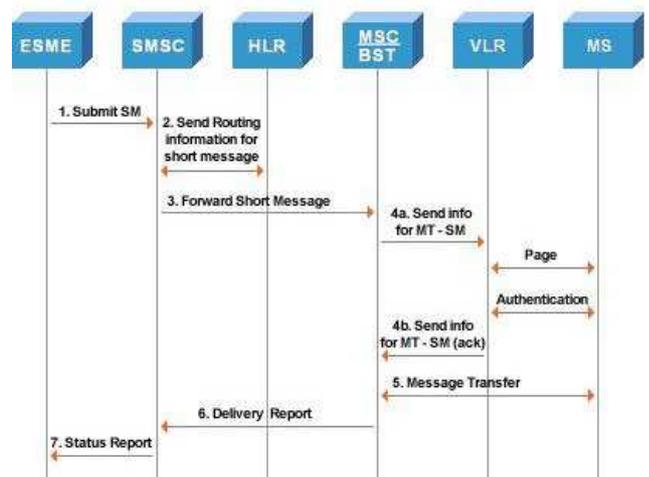


Gambar 3. Skenario SMS MO (*Mobile Originating*)

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. SMSC memberitahukan MSC bahwa SMS telah dikirimkan ke SME.

Sedangkan skenario penerimaan SMS dapat dilihat 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. [2]

### 3. Kode Huffman

Dalam ilmu komputer dan teori informasi, kode Huffman adalah sebuah pengkodean entropi algoritma yang berfungsi untuk kompresi sebuah data. Dasar dari kode Huffman ini adalah frekuensi-sorted binary tree.

#### 3.1 Sejarah Kode Huffman

Pada tahun 1951, David A. Huffman dan teman-temannya di mata kuliah "Teori Informasi" di MIT diberikan pilihan untuk menulis paper untuk ujian akhir. Sang profesor, Robert M. Fano, menugaskan untuk membuat makalah tentang kode biner yang paling efisien. Huffman, tidak dapat membuktikan kode manapun yang paling efisien. Kemudian saat ia menyerah tiba-tiba ia mendapatkan ide untuk menggunakan *frequency-sorted binary tree* dan akhirnya membuktikan bahwa metodenya yang paling efisien.

Berawal dari kejadian itu Kode Huffman diciptakan.

#### 3.2 Teori Kode Huffman

Pengkodean Huffman menggunakan metode tertentu untuk memilih perwakilan untuk setiap simbol, dan menghasilkan sebuah kode awalan (kadang-kadang disebut "awalan kode bebas") (yaitu bit string yang mewakili simbol tertentu tidak pernah merupakan awalan dari string bit yang mewakili setiap simbol lain) yang menyatakan karakter yang paling umum menggunakan string bit yang lebih pendek daripada yang digunakan untuk simbol sumber kurang umum.

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.

Suatu kode Huffman didapatkan dengan cara berikut.

Pertama-tama buat tabel frekuensi dari semua simbol atau karakter yang muncul dalam suatu text. Kemudian diurutkan mulai dari simbol dengan frekuensi paling sedikit sampai simbol dengan frekuensi paling banyak. Pembentukan huffman tree dimulai dari dua simbol paling depan untuk dijadikan anak kiri dan anak kanan dari tree yang terbentuk dan frekuensinya dijumlahkan. Setelah itu diurutkan kembali berdasarkan frekuensi yang baru. Demikian dilakukan terus menerus sampai semua simbol terbentuk menjadi tree. Untuk dapat memahaminya diberikan contoh berikut ini.

Kode yang sering digunakan adalah kode ASCII. Dengan kode ASCII setiap karakter dikodekan dalam 8 bit biner. Tabel 1 berikut adalah contoh kode ASCII.

Tabel 1. Tabel kode ASCII

Simbol	Kode ASCII
<b>A</b>	<b>01000001</b>
<b>B</b>	<b>01000010</b>
<b>C</b>	<b>01000011</b>
<b>D</b>	<b>01000100</b>

Dengan mengikuti ketentuan pengkodean pada Tabel 1, string 'ABACCD A' direpresentasikan menjadi rangkaian bit:

01000001010000100100000101000011010000110100010001000001

Berdasarkan metode pengkodean ASCII, representasi 7 huruf membutuhkan  $7 \times 8 = 56$  bit (7 byte). Untuk meminimumkan jumlah bit yang dibutuhkan, panjang kode untuk setiap karakter sedapat mungkin diperpendek, terutama karakter yang kekerapan (*frequency*) kemunculannya besar. Karena itu digunakan metode Huffman. Pertama-tama untuk menjadikannya kode Huffman, buat dulu tabel kekerapan dan kode Huffman untuk string 'ABACCD A' (lihat Tabel 2).

**Tabel 2. Tabel kekerapan dan kode Huffman untuk string 'ABACCCA'**

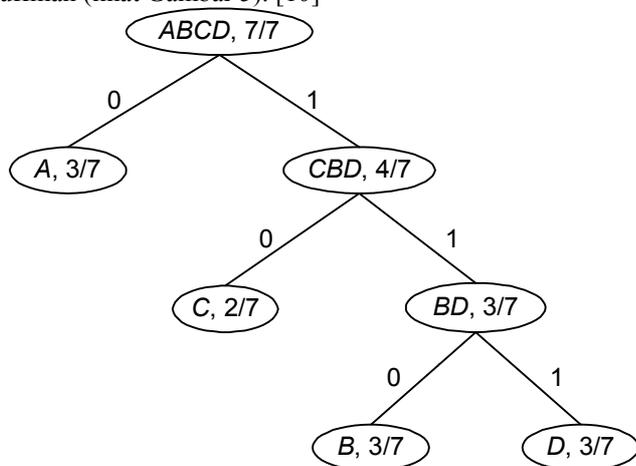
Simbol	Kekerapan	Peluang	Kode Huffman
<b>A</b>	<b>3</b>	<b>3/7</b>	<b>0</b>
<b>B</b>	<b>1</b>	<b>1/7</b>	<b>110</b>
<b>C</b>	<b>2</b>	<b>2/7</b>	<b>10</b>
<b>D</b>	<b>1</b>	<b>1/7</b>	<b>111</b>

Dengan menggunakan kode Huffman, pesan 'ABACCCA' ini direpresentasikan menjadi rangkaian bit:

0110010101110

Jadi dengan kode Huffman, bit yang dibutuhkan untuk string 'ABACCCA' menjadi 13 bit dari 56 bit.

Untuk mendapatkan kode Huffman, digunakan Pohon Huffman (lihat Gambar 5). [10]



**Gambar 5. Pohon Huffman untuk pesan 'ABACCCA'**

### 3.3 Kompresi dan Dekompresi

Seperti pada contoh di atas, proses memampatkan pesan string 'ABACCCA' dari 56 bit menjadi 13 bit disebut sebagai kompresi. Namun kompresi ini masih berupa bentuk dalam kode Huffman, sedangkan sudah jelas, kode Huffman untuk setiap pesan itu tidak sama. Karena itu diperlukan suatu konversi dari kode Huffman kembali ke kode ASCII. Proses pengonversian ini disebut dekompresi.

Dekompresi dapat dilakukan dengan membaca dari Pohon Huffman, ataupun dari Tabel Huffman. Perlu diperhatikan bahwa setiap kode Huffman memiliki kode awalan yang unik.

### 4. Kode Huffman dan Kompresi SMS

Salah satu contoh aplikasi yang dapat dilihat kompresinya adalah Java Microedition 2.0. Pada aplikasi ini kode Huffman yang digunakan dapat dilihat pada Tabel 3.

**Tabel 3. Kode Huffman pada Aplikasi Java Microedition 2.0**

	Awalan	Badan	Ubahan
a		0000	00000
b		00101	000101
c		1101	01101
d		01011	001011
e		11001	011001
f		10101	010101
g		1000	01000
h		0001	00001
i		110001001	0110001001
j		1100011	01100011
k		1001	01001
l		10100	010100
m		0110	00110
n		0100	00100
o		10110	010110
p		1100010000	01100010000
q		0111	00111
r		0011	00011
s		111	0111
t		10111	010111
u		0101001	00101001
v		0101000	00101000
w		11000101	011000101
x		00100	000100
y		1100010001	01100010001
z		0000	1000000
A		00101	10000101
B		1101	1001101
C		01011	10001011
D		11001	10011001
E		10101	10010101
F		1000	1001000
G		0001	1000001
H		110001001	100110001001
I		1100011	1001100011
J		1001	1001001
K		10100	10010100
L		0110	1000110
M		0100	1000100
N		10110	10010110
O		1100010000	1001100010000
P		0111	1000111
Q		0011	1000011
R		111	100111
S		10111	10010111
T		0101001	1000101001
U		0101000	1000101000
V		11000101	10011000101
W		00100	10000100
X		1100010001	1001100010001
Y		00100	10000100
Z		1100010001	1001100010001

Bandingkan kode Huffman diatas dengan kode ASCII yang sering digunakan dalam pengkodean.

Tabel 4. Kode ASCII untuk Character

Letter	ASCII Code	Binary	Letter	ASCII Code	Binary
a	097	01100001	A	065	01000001
b	098	01100010	B	066	01000010
c	099	01100011	C	067	01000011
d	100	01100100	D	068	01000100
e	101	01100101	E	069	01000101
f	102	01100110	F	070	01000110
g	103	01100111	G	071	01000111
h	104	01101000	H	072	01001000
i	105	01101001	I	073	01001001
j	106	01101010	J	074	01001010
k	107	01101011	K	075	01001011
l	108	01101100	L	076	01001100
m	109	01101101	M	077	01001101
n	110	01101110	N	078	01001110
o	111	01101111	O	079	01001111
p	112	01110000	P	080	01010000
q	113	01110001	Q	081	01010001
r	114	01110010	R	082	01010010
s	115	01110011	S	083	01010011
t	116	01110100	T	084	01010100
u	117	01110101	U	085	01010101
v	118	01110110	V	086	01010110
w	119	01110111	W	087	01010111
x	120	01111000	X	088	01011000
y	121	01111001	Y	089	01011001
z	122	01111010	Z	090	01011010

Contoh dari pengaplikasian kode Huffman pada Tabel 3 adalah jika kita hendak mengirimkan sms yang bertuliskan 'sayacantik'.

Perbandingan dalam menggunakan kode ASCII dengan kode Huffman untuk kata 'sayacantik' dapat dilihat dalam Tabel 5.

Tabel 5. Perbandingan kode ASCII dengan kode Huffman

Karakter	Kode ASCII	Kode Huffman
s	01110011	00011
a	01100001	00000
y	01111001	000100
a	01100001	00000
c	01100011	01101
a	01100001	00000
n	01101110	00110
t	01110100	0111
i	01101001	00001
k	01101011	01100011

Dari tabel di atas, dapat dilihat bahwa dengan menggunakan kode ASCII, bit yang dibutuhkan sebanyak  $10 \times 8 = 80$  bit. Sedangkan dengan kode Huffman, dibutuhkan sebanyak 53 bit saja. Karena itu dapat disimpulkan dengan menggunakan kode Huffman, dapat menghemat 34% bit yang digunakan daripada dengan menggunakan kode ASCII.

Gambar 6 akan menampilkan contoh nyata penggunaan aplikasi kompresi SMS.



Gambar 6. (a) Format pengiriman aplikasi (b) SMS yang hendak dikirim (c) Verifikasi pengiriman SMS (d) Tampilan jika penerima tidak memiliki aplikasi yang sama

## IV. KESIMPULAN

Kode Huffman adalah salah satu metode kompresi file yang banyak digunakan. Konsep dari kode Huffman adalah *frequency-sorted binary tree*.

Kode Huffman adalah salah satu solusi dalam masalah pengiriman pesan singkat dengan karakter yang membutuhkan bit banyak. Dengan kode Huffman ini, bit yang dibutuhkan dapat terkompresi sekitar 30-36% dari ukuran aslinya. Dengan kompresi ini, biaya yang dibutuhkan akan semakin kecil sehingga bisa lebih hemat.

## REFERENSI

- [1] Wikipedia, Ensiklopedi Bebas. *Layanan Pesan Singkat*. 2009. <http://id.wikipedia.org/wiki/Sms>. Waktu akses : Jumat, 18 Desember 2009 pukul 20.00 WIB.
- [2] Firman, Hengki. *Tentang SMS*. 2009. <http://hengky-style.blogspot.com/2009/07/sms-short-message-service-secara-umum.html>. Waktu akses : Jumat, 18 Desember 2009 pukul 20.00 WIB.
- [3] Edukasi, Yayasan Total Sarana. *Pengertian SMS*. 2005. <http://www.total.or.id/info.php?kk=Short%20Message%20ervice>. Waktu akses : Jumat, 18 Desember 2009 pukul 20.00 WIB.
- [4] Tondowijoyo, Benny. *Perkembangan Teknologi Informasi*. 2008. <http://one.indoskripsi.com/judul-skripsi-tugas-makalah/teknik-elektro/perkembangan-teknologi-komunikasi>. Waktu akses : Jumat, 18 Desember 2009 pukul 20.00 WIB.
- [5] Agastiya, Bayu. *Teknologi AMPS dan GSM*. 2009. <http://lalaboy.wordpress.com/2009/04/29/teknologi-amps-gsm/>. Waktu akses : Jumat, 18 Desember 2009 pukul 20.00 WIB.
- [6] Hendriadi, Ade Andri. *Perkembangan Teknologi Seluler*. 2007. <http://hendriadi.wordpress.com/2007/07/21/perkembangan-teknologi-seluler/>. Waktu akses : Jumat, 18 Desember 2009 pukul 20.00 WIB.
- [7] Wikipedia, The Free Encyclopedia. *Huffman Coding*. 2009. [http://en.wikipedia.org/wiki/Huffman\\_coding](http://en.wikipedia.org/wiki/Huffman_coding). Waktu akses : Jumat, 18 Desember 2009 pukul 20.00 WIB.
- [8] Yuku. *Kompresi Teks SMS*. Kejut. 2007. <http://www.kejut.com/kompresisms>. Waktu akses : Jumat, 18 Desember 2009 pukul 20.00 WIB.
- [9] *ASCII-Binary Character Table*. 2009. <http://sticksandstones.kstrom.com/appen.html>. Waktu akses : Minggu, 20 Desember 2009 pukul 07.00 WIB.
- [10] Munir, Rinaldi. *Diktat Kuliah IF2091 Struktur Diskrit*. Program Studi Teknik Informatika. 2008.
- [11] Lipesik, Victor Julian. *Pembuatan Perangkat Lunak untuk Kompresi File Text dengan Huffman Tree*. Universitas Kristen Petra. 2009.