

IMPLEMENTASI KODE HUFFMAN DALAM APLIKASI KOMPRESI TEXT PADA LAYANAN SMS

Yuza Reswan¹, Dedy Agung Prabowo²
¹²Universitas Muhammadiyah Bengkulu, Indonesia
dedyagungprabowo@umb.ac.id

| | | |
|-----------------------|--------------------|----------------------|
| Received: 28-05- 2023 | Revised: 03-6-2023 | Approved: 07-06-2023 |
|-----------------------|--------------------|----------------------|

ABSTRAK

Layanan SMS merupakan cara pengiriman pesan singkat dalam waktu cepat dengan relative murah, pesan singkat yang akan dikirimkan dengan kapasitas maksimal 140 byte untuk setiap pesan. Untuk mengatasi kesulitan penyampaian pesan dengan jumlah karakter banyak ataupun karakter dengan bit besar, maka dibuatlah 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 dengan kekerapan kecil. Pembentukan kode Huffman dimulai dengan suatu pohon Huffman. Proses ini dilanjutkan dengan melakukan proses encoding terhadap karakter tersebut sesuai dengan kode Huffman yang ada. Setelah tersampaikan proses decoding dilanjutkan agar pesan dapat di baca kembali.

Kata Kunci: Kode Huffman, SMS.

PENDAHULUAN

Komunikasi merupakan suatu proses penyampaian informasi yang terjadi dari suatu pihak ke pihak lain, hal ini terjadi dalam kehidupan sehari-hari. Berbagai macam perangkat komunikasi dikembangkan dan digunakan untuk menunjang tercapainya proses tersebut. Telepon genggam, salah satu perangkat yang memiliki fitur layanan SMS (*Short Message Service*) merupakan layanan pengiriman pesan singkat dalam jaringan komunikasi.

Short Message Service (SMS) merupakan salah satu layanan yang disediakan oleh sistem telekomunikasi seluler GSM (*Global System of Mobile Telecommunication*) yang pertama kali dikembangkan di Eropa dan sistem telekomunikasi lainnya yang berbasis pada jaringan tanpa kabel. Sesuai dengan namanya *Short Message Service* (SMS) hanya menyediakan layanan pengiriman pesan/informasi hanya dalam jumlah yang sangat terbatas. Seperti yang ditentukan oleh organisasi ETSI (*European Telecommunication Standars Institute*) di dalam dokumen GSM 03.40 dan 03.38, satu paket *Short Message Service* (SMS) hanya dapat membawa pesan sebanyak 160 karakter, 140 karakter atau hanya 70 karakter sesuai dengan sistem pengkodean yang digunakan. Sistem pengkodean 7 bit per karakter adalah sistem pengkodean *default* untuk pesan SMS yaitu berupa alphanumeric Latin yang dikenal luas.

Huffman Coding adalah salah satu pemampatan data algoritma yang menggunakan frekuensi/probabilitas kemunculan dari simbol pada sebuah *string* sebagai input dan menghasilkan output berupa *prefix code* (kode awalan) yang mengkodekan *string* menggunakan bit paling sedikit dari seluruh kemungkinan *binary prefix code*. Algoritma ini dikembangkan oleh David A. Huffman pada paper yang ditulisnya sebagai prasyarat kelulusannya di MIT, prinsip yang di gunakan pada algoritma huffman mirip dengan kode morse yaitu tiap karakter (simbol) dikodekan hanya dengan rangkaian beberapa bit, dimana karakter yang sering muncul dikodekan dengan rangkaian bit yang pendek dan karakter yang jarang muncul dikodekan dengan rangkaian bit yang lebih panjang. *Huffman Coding* salah satu algoritma dasar untuk kompresi data, yang bertujuan untuk mengurangi jumlah bit yang diperlukan untuk merepresentasikan informasi / pesan.

Untuk membentuk suatu kode Huffman dimulai dengan membuat pohon biner. Pohon ini akan disimpan pada suatu tabel, dengan ukuran yang bergantung pada jumlah dari simbol. 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, bisa juga mengandung *link* ke simpul orang tua yang akan memudahkan pembacaan kode. Pada simbol dalam terdapat bobot *link* ke dua simpul anak dan bisa ke simpul orang tua. 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.

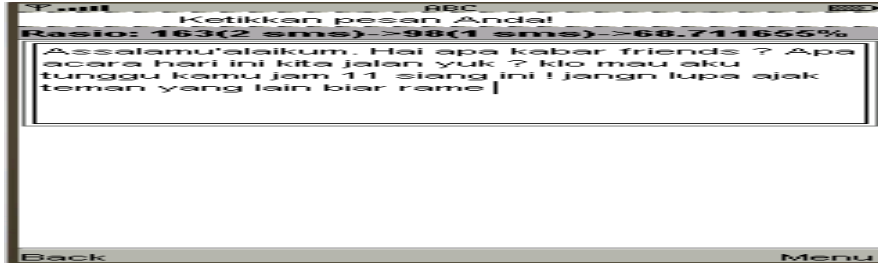
J2ME merupakan *Java Runtime Environment (JRE)* lengkap, yang terdiri dari *configuration* yang menentukan JVM yang dipakai, dan profile yang mendefinisikan aplikasi yang didukung yaitu dengan menambahkan *domain-specific class*. (M.shalahudin dan Rosa A.S, 2008 :3) *Configuration* mendefinisikan bagian yang berisi JVM dan JRE yang digunakan oleh J2ME sehingga aplikasi yang dibuat dengan menggunakan J2ME dapat berjalan pada peralatan dengan tipe yang telah ditentukan.

HASIL PENELITIAN

PDU SMS yang digunakan untuk membawa pesan SMS terdiri dari delapan bagian *Header*. Dimana bagian - bagian *Header* tersebut telah dijelaskan pada pembahasan mengenai *Short Message Service (SMS)*. Dari kedelapan *Header* tersebut hanya sebuah *Header* yang digunakan untuk menampung pesan *Short Message Service (SMS)* yang dikirimkan oleh *user* yang menggunakan layanan pengiriman *Short Message Service (SMS)*.

Semakin pendek sistem pengkodean yang digunakan maka jumlah simbol ASCII didalam pesan SMS yang dapat ditampung akan semakin banyak. Pada aplikasi yang menggunakan J2ME, *Header* isi SMS selain digunakan untuk menyimpan pesan SMS juga digunakan untuk menyimpan informasi nomor port yang digunakan untuk mengirimkan dan menerima pesan SMS yang dikirimkan dengan menggunakan aplikasi pesan yang dibuat menggunakan J2ME. Jumlah data pada *Header* tersebut yang digunakan untuk menyimpan informasi nomor port ini adalah sebesar 7 byte. Misalkan hasil kompresi pesan SMS hanya 98 byte dengan ukuran sebesar ini, dalam kondisi

optimal jumlah symbol ASCII yang dapat dikirimkan dengan menggunakan satu PDU SMS adalah:



Gambar 1. Pesan SMS

$$\frac{98 \times 8}{5} = \frac{784}{5} = 156,8 \rightarrow 156 \text{ simbol ACSII}$$

Kapasitas maksimum ini dapat dicapai, dengan beberapa kondisi di bawah ini:

- 1) Menggunakan simbol yang panjang kodenya hanya lima bit per simbol.
- 2) Menggunakan simbol yang panjangnya lima bit per simbol sebanyak 92 simbol ditambah empat simbol dengan panjang 6 bit per simbol.

Untuk kondisi lainnya kapasitas maksimum paket SMS sebesar 156 simbol tidak dapat dicapai. Dari pesan SMS diatas dapat dilihat bahwa SMS kompresi dapat mengirimkan sebanyak 156 simbol ASCII dengan menggunakan satu PDU SMS. Beberapa hal yang perlu diperhatikan berkaitan dengan kapasitas satu PDU SMS yang dapat dicapai aplikasi SMS kompresi pada contoh pesan SMS yang telah disebutkan di atas adalah :

1. Kapasitas satu PDU SMS untuk mengirimkan pesan SMS tidak konstan, tetapi berubah sesuai dengan jenis simbol ASCII yang digunakan. Semakin banyak simbol ASCII dengan kode pendek digunakan dalam pesan SMS, maka kapasitas PDU SMS semakin banyak. Dengan menggunakan asumsi panjang kode simbol rata-rata selalu lebih kecil dari pada 4,87 bit per simbol, maka SMS kompresi masih mampu menambah kapasitas PDU SMS paling tidak satu simbol lebih banyak. Nilai batas atas dari panjang rata-rata simbol ASCII tersebut diperoleh dengan cara kapasitas PDU standar adalah 160 simbol. Maka agar SMS kompresi masih memberikan tambahan simbol dalam satu PDU SMS minimal satu simbol lebih banyak dari pada kapasitas PDU SMS standar, panjang bit maksimumnya adalah:

$$\frac{98 \times 8}{161} = 4,87 \text{ bit tiap simbol}$$

2. Jika terjadi kasus khusus yaitu adanya penggunaan simbol ASCII yang sangat jarang digunakan melebihi batas maksimum yang diizinkan dalam pesan SMS maka SMS kompresi tidak dapat melakukan proses kompresi pesan SMS sama sekali. Kondisi yang menyebabkan kasus khusus tersebut adalah :
 - a. Jumlah simbol dengan panjang 11 bit melebihi 43 buah.
 - b. Jumlah simbol dengan panjang 10 bit melebihi 51 buah.
 - c. Jumlah simbol dengan panjang 8 bit melebihi 87 buah.

Pesan SMS tersebut akan dikodekan dengan menggunakan pohon Huffman tetap yang telah didefinisikan dan pohon Huffman yang dibangun pada saat proses pengkodean SMS.

Kode pesan SMS dengan pohon Huffman yang tetap.

- Simbol , panjang kode : 5, Kode : 00000
- Simbol ' , panjang kode : 10, Kode : 011100101
- Simbol - , panjang kode : 10, Kode : 001101010

Data byte inilah yang akan dikirimkan dengan menggunakan PDU SMS. Panjang data tersebut adalah 98 byte, yang merupakan ukuran data maksimum yang dapat dikirimkan dengan menggunakan satu PDU SMS.

Kode pesan SMS dengan pohon Huffman yang berubah.

- Penghitungan frekuensi kemunculan karakter :
- fregtable[32] atau alphabel frekuensinya adalah : 31
- fregtable[39] atau alphabet ' frekuensinya adalah : 1
- fregtable[45] atau alphabet - frekuensinya adalah : 1
- fregtable[46] alau alphabet . frekuensinya adalah : 7

Perbandingan hasil kompresi kedua metode tersebut.

Dari data diatas dapat dilihat bahwa data byte hasil kompresi yang menggunakan pohon Huffman yang variabel jauh lebih kecil jika dibandingkan dengan data byte hasil kompresi yang menggunakan tabel tetap yang telah didefinisikan sebelumnya. Perbandingan tersebut dapat dilihat di bawah ini :

- a) Panjang data byte hasil kompresi dengan pohon Huffman tetap: 163 byte.
- b) Panjang data byte hasil kompresi dengan pohon Huffman berubah: 107 byte.
- c) Panjang pesan asli sebelum dikodekan dengan kode Huffman : 208 byte.

Tabel 5.
 Perbandingan Kompresi Pesan SMS dengan Pohon Huffman Tetap

| | Tabel Tetap | Tabel Berubah |
|-------------------------------|--------------------------------------------------|--------------------------------------------------|
| Panjang rata-rata tiap symbol | $\frac{163}{208} = 0,783$ byte | $\frac{107}{208} = 0,514$ byte |
| Compression rate | $\frac{(208 - 163)}{208} \times 100\% = 21,63\%$ | $\frac{(208 - 107)}{208} \times 100\% = 48,56\%$ |

Dari tabel diatas terlihat bahwa hasil kompresi pesan SMS dengan menggunakan pohon Huffman yang berubah jauh lebih baik dari pada hasil kompresi dengan pohon Huffman yang tetap.

Oleh karena itu, jika penulis hanya melihat hasil ini saja tentunya akan menggunakan pohon Huffman yang berubah untuk melakukan proses kompresi pesan SMS. Namun yang terjadi adalah sebaliknya, penulis menggunakan pohon Huffman yang tetap untuk melakukan proses kompresi pesan SMS. Hal ini dapat dijelaskan sebagai berikut :

- 1) Pada pengirim dan penerima pesan SMS yang menggunakan aplikasi SMS kompresi diperlukan pohon Huffman yang sama untuk melakukan proses kompresi dan dekompresi pesan SMS.
- 2) Pohon Huffman yang sama pada pengirim dan penerima dapat diperoleh dengan menggunakan pohon Huffman tetap yang telah didefinisikan terlebih dahulu atau dengan mengirimkan pohon Huffman bersama dengan data byte hasil kompresi pesan SMS.
- 3) Pada aplikasi SMS kompresi yang menggunakan pohon Huffman tetap yang telah didefinisikan terlebih dahulu pohon Huffman tidak perlu dikirimkan bersama data byte hasil kompresi pesan SMS. Sehingga pada pesan SMS yang disebutkan pada bagian ini, data yang dikirimkan sebesar 98 byte atau hanya memerlukan sebuah PDU SMS.
- 4) Oleh karena itulah aplikasi SMS kompresi menggunakan pohon Huffman tetap didefinisikan terlebih dahulu. Hal ini disebabkan karena untuk ukuran pesan SMS yang sama jumlah PDU SMS yang diperlukan untuk mengirimkan pesan SMS yang dikompresi dengan pohon Huffman tetap lebih sedikit dari pada yang menggunakan pohon Huffman yang berubah.

Tabel 6.

Perbandingan Kompresi Pesan SMS Setelah Melibatkan Pohon Huffman

| | Pesan SMS dikirim dengan pohon Huffman tetap | Pesan SMS dikirim dengan pohon Huffman berubah |
|---------------------------------------|-----------------------------------------------------|-------------------------------------------------------|
| Ukuran pesan dikirim | 98 byte | 157 byte |
| PDU SMS diperlukan | 1 buah | 2 buah |
| <i>Compression rate</i> pesan dikirim | $\frac{(208 - 98)}{208} \times 100\%$ = 52,88% | $\frac{(208 - 157)}{208} \times 100\% =$ 24,52% |

Dapat dilihat pada tabel di atas bahwa jumlah PDU SMS yang diperlukan untuk mengirimkan pesan SMS yang dikompresi dengan pohon Huffman tetap lebih sedikit.

Tampilan Awal Aplikasi SMSEnkripsi

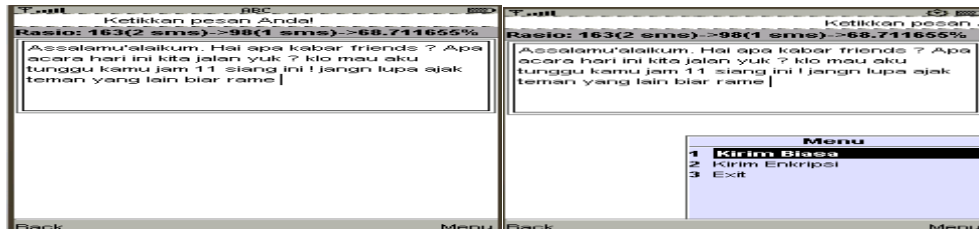


Gambar 2. Default Phone Emulator

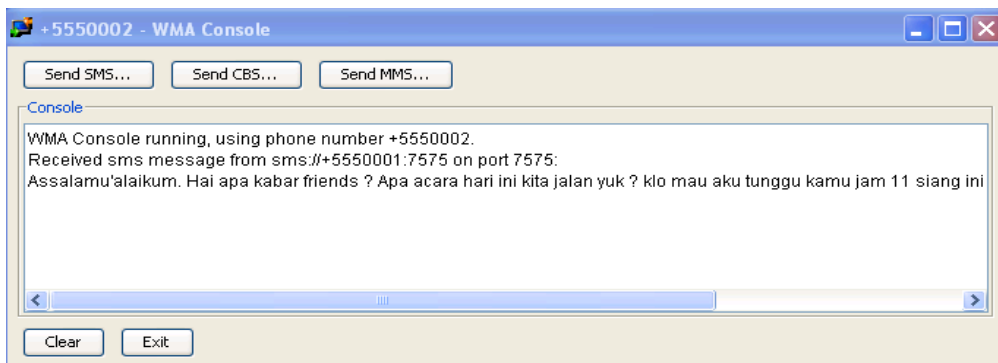
Kode merupakan deklarasi yang menyatakan kelas First Font yang diperlukan sebagai sebuah objek MIDlet. Bagian source display mengembalikan objek display dari MIDlet untuk menampilkan displayable canvas. Metode exitMIDlet dipergunakan untuk fungsi keluar pada MIDlet. Klik tombol Lunch untuk menjalankan MIDlet ke menu password.

Pengujian Pengiriman Dan Penerimaan Pesan SMS

Data dibawah ini adalah data yang di output-kan pada emulator dijalankan untuk mengirimkan SMS.



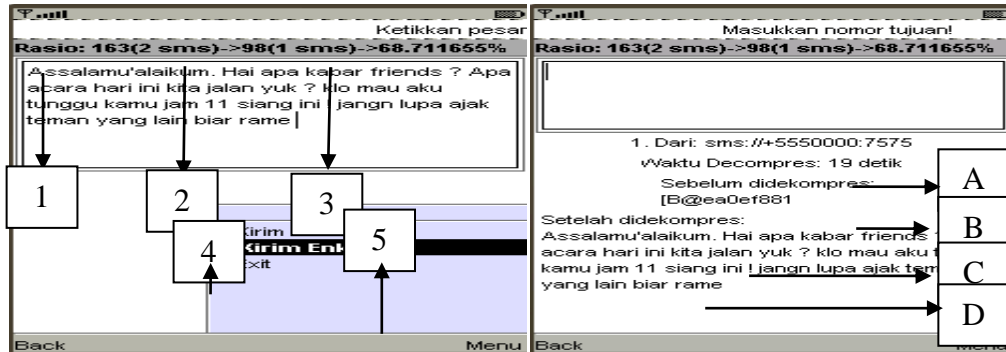
Gambar 3. proses pengiriman SMS biasa



Gambar 4. proses penerimaan SMS biasa menggunakan WMA Console

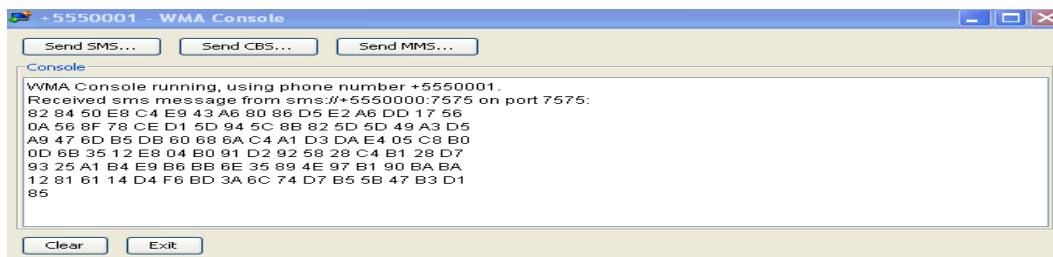
Dari proses diatas dapat dilihat bahwa pesan yang dikirimkan oleh pengirim sama dengan pesan yang diterima oleh penerima. Hal ini dikarenakan tidak ada proses enkripsi / deskripsi dalam pengiriman maupun penerimaan. Berbeda dengan pesan yang

dikirim melalui proses enkripsi oleh pengirim dan didekompresi oleh penerima. Maka pesan yang didapat akan berbeda.



Gambar 5. proses pengiriman dan penerimaan SMS Enkripsi

Data hasil simulasi kompresi adalah data yang dikirimkan oleh pengirim dan diterima oleh penerima.



Gambar 6. proses penerimaan sms enkripsi menggunakan WMA Console

Dari hasil yang didapat terlihat bahwa pesan yang diterima oleh penerima berupa bilangan heksadesimal. Karena kedua belah pihak tidak menggunakan program yang sama, maka pesan yang diterima berbeda dengan pesan yang dikirimkan, hal ini terjadi karena tidak adanya dekomposisi oleh penerima. Ketika proses pengiriman dengan enkripsi metode Huffman, maka pada toolkit akan terlihat sebagai berikut :

```

Kirim ke: sms://+5550001:7575
SMS dari com.sun.midp.io.j2me.sms.BinaryObject@f828ed68
1000001010000100010100001110100011000100111010010100001110100110100000
0010000110110101011110001010100110110111010001011101010110000010100101
0110100011110111100011001110110100010101110110010100010111
    
```

Data tersebut adalah data yang dihasilkan oleh program setelah melalui proses perpindahan dari huruf kode standar ke *alphanumeric* biner.

KESIMPULAN

Implementasi *Huffman Coding* dengan menggunakan aplikasi *SMSEnkripsi* yang berbasis pada J2ME menunjukkan kesimpulan sebagai berikut : 1) Dengan menggunakan aplikasi SMS Kompresi untuk mengirim sms, biaya yang dikeluarkan dapat ditekan menjadi lebih murah, pesan yang terkirim akan lebih aman karena pesan

yang dikirim maupun yang diterima terbungkus secara enkripsi yang memungkinkan terhindarnya penyadapan yang dilakukan dari pihak luar. 2) *Huffman Coding* dapat digunakan untuk melakukan proses kompresi pesan SMS dengan *compression rate* mencapai 52,8% dalam kondisi optimum dan mencapai *compression rate* lebih dari 24,5% dengan mudah dalam kondisi standar.

DAFTAR PUSTAKA

- Anonym, *Pengiriman Pesan Waktu* akses : 26 mei 2009. Pukul 17:00 wib. http://id.wikipedia.org/wiki/pengiriman_pesan_instan.com.
- Budi Raharjo, Imam Heryanto & Arif Haryono. 2010. *Tuntunan Pemrograman Java Untuk Handphone Dan Alat Telekomunikasi Mobile Lainnya*. Bandung : Informatika
- Farid. *SMS Murah Untuk Semua Merk Hp*. Waktu akses : 16 Mei 2009 pukul 13:00 wib. <http://www.costfix.net/2007/12/19/costfix/smsmurah-untuk-semua-merk-hp-dan-operatorselular>.
- Munir, Renaldi. 2007. *Matematika diskrit*. Bandung : informatika
- Shalahudin. M & Rosa A. S. 2008. *Pemrograman J2ME Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile*. Bandung : Informatika
- Sujaini, Herry dan Yessi Mulyani. 2008. *Algoritma Run-Length half-Byte dan Huffman untuk pemampatan file*. Bandung : Teknik Elektro Institut Tehnik Bandung.
- Wikipedia, *The Free Encyclopedia Huffman Coding*. Waktu akses : 16 Mei 2009 pukul 12:05 wib. <http://en.wikipedia.org/wiki/Huffmancoding>.
- Wardoyo, Irwan. 2008. *Kompresi Teks Dengan Menggunakan Algoritma Huffman*. Sekolah Tinggi Teknologi Telkom : Bandung.