

APLIKASI JAVA MOBILE UNTUK KOMPRESI LAYANAN PESAN SINGKAT

Iwan Handoyo Putro¹, Petrus Santoso², Maya Basoeki³

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Universitas Kristen Petra
Siwalankerto 121-131, Surabaya 60236

E-mail : ¹iwanhp@petra.ac.id, ²petrus@petra.ac.id, ³m23404047@john.petra.ac.id

ABSTRAKS

Paper ini membahas pembuatan aplikasi kompresi teks untuk meningkatkan jumlah karakter pada setiap pengiriman pesan singkat (SMS) dengan mengimplementasikan algoritma kompresi aritmatika. Aplikasi ini tidak terintegrasi dengan aplikasi SMS bawaan pada Mobile Phone dan diimplementasikan pada Mobile Phone berbasis Java yang mendukung MIDP 2.0 dan CLDC 1.1.

Tingkat keberhasilan proses kompresi pesan sangat dipengaruhi oleh penggunaan jenis huruf sesuai dengan tingkat probabilitas yang telah ditetapkan. Pengujian memperlihatkan peningkatan sebesar 16 karakter untuk pengiriman 1 SMS, 21 karakter pada pengiriman 2 SMS dan 37 karakter pada pengiriman 3 SMS.

Kata Kunci : kompresi, J2ME, layanan pesan singkat, arithmetic coding

1. Pendahuluan

Penggunaan telepon selular sebagai salah satu media komunikasi dan informasi saat ini menjadi suatu kebutuhan yang cukup penting. Selain digunakan sebagai media komunikasi suara, telepon selular juga digunakan untuk berkomunikasi lewat layanan *Short Message Services* (SMS).

Aplikasi SMS yang tersedia pada *mobile phone*, menetapkan pengiriman teks dengan ukuran kurang atau sama dengan 160 karakter untuk setiap 1 kali pengiriman SMS.. Pengiriman 1 kali SMS adalah pengiriman data teks sebanyak 1.120 bit (160 karakter x 7 bit). Apabila pelanggan mengirimkan lebih dari 1.120 bit atau lebih dari 160 karakter maka biaya yang dibayarkan pelanggan dihitung sebagai kelipatannya. Sebagai contoh, SMS dengan jumlah 161 karakter tetap akan dihitung 2 kali SMS. Padahal karakter lebih hanya 1 karakter dan hal ini menjadi kerugian bagi pelanggan karena harus dikenai biaya 2 kali SMS untuk penambahan karakter lebih tersebut.

Inilah yang melatarbelakangi pembuatan aplikasi kompresi pesan singkat dengan memanfaatkan arithmetic coding. Dengan aplikasi ini diharapkan dapat meningkatkan kapasitas karakter terkirim setiap pengiriman SMS dapat ditingkatkan.

2. Arithmetic Coding

Algoritma *Arithmetic Coding* merupakan salah satu algoritma kompresi yang bersifat *loseless*, yang berarti jika ada *code number* yang hilang maka pesan akan hilang (Langdon, Jr, 1984). Selain itu, *Arithmetic Coding* juga bersifat statik, karena menggunakan tabel probabilitas yang tetap. *Arithmetic Coding* bekerja dengan cara mengubah sebuah *String Message* yang berisi kumpulan

karakter menjadi sebuah *code number* yang bernilai antara 0 sampai 1 (Witten, Neal, Cleary, 1987). Semua karakter yang mungkin muncul dalam sebuah *String Message* ditentukan probabilitas kemunculannya. Dalam setiap proses *Coding* (pengkodean) ada 2 proses penting yaitu proses *Encoding* dan *Decoding*. Berikut merupakan proses *Encoding* dan *Decoding* dari *Arithmetic Coding*.

2.1. Encoding

Proses ini merupakan proses untuk mengubah sebuah *String Message* menjadi sebuah *code number*. Pada tabel probabilitas karakter yang mungkin muncul dalam sebuah *message*, semua karakter memiliki distribusi probabilitas yang berbeda yaitu dari *Low Range – High Range*. Range ini ditentukan berdasarkan probabilitas karakter tersebut (Howard, Vitter, 1992).

Walaupun terdapat karakter yang memiliki probabilitas sama namun *range* probabilitasnya pasti berbeda. Dan *range* semua karakter yang mungkin muncul harus berada dalam range 0 -1.

Prinsip kerja *Encoding* teks pada Algoritma ini adalah sebagai berikut :

1. Nilai *low* dan *high* untuk pertama kali adalah $low = 0$ dan $high = 1$
2. Karakter pertama dicari $current_range = high - low$
3. Cari nilai *high* dan *low* yang baru dengan menggunakan rumus :
 $high = low + current_range \times high_range$ (karakter pertama)
 $low = low + current_range \times low_range$ (karakter pertama)
4. Jika tidak terdapat lagi karakter dalam *Message* maka proses berhenti di sini. Namun, jika masih terdapat karakter dalam *Message* maka proses di ulang dari langkah 2 sampai karakter terakhir. Dengan demikian setelah karakter terakhir maka

akan diperoleh sebuah *range high* dan *low* yang baru.

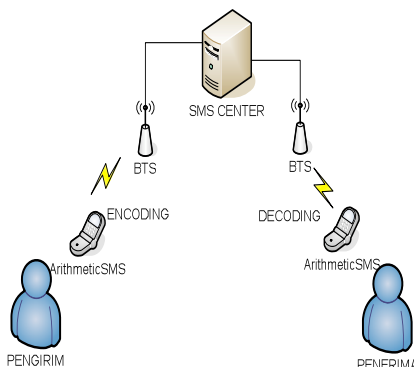
2.2. Decoding

Proses ini merupakan proses untuk mengubah sebuah *Code number* untuk mendapatkan *Message* (Langdon, Jr, 1984). Prinsip kerja algoritma *decoding arithmetic* adalah sebagai berikut:

1. Dari *code number* yang diperoleh, dapat langsung ditentukan karakter pertama dari *Message* yang dikirim. Caranya dengan melihat pada tabel probabilitas, di *range* karakter yang manakah *code number* itu berada.
2. Jika sudah mengetahui karakter pertama yang muncul, maka selanjutnya cari *code_range*-nya dengan menggunakan rumus:
$$\text{code_range} = \text{high_range symbol} - \text{low_range symbol}$$
3. Dapatkan nilai *encoded_number* yang baru dengan menggunakan rumus:
$$\text{encoded_number} = (\text{encoded_number} - \text{low_range symbol}) / \text{code_range}$$
4. Berdasarkan *encoded_number* kita bisa menentukan karakter baru yang muncul sekarang. Jika data belum selesai, kembali ke langkah 2. Jika sudah berakhir, maka proses selesai.

3. Perancangan Aplikasi

Aplikasi ini bertujuan untuk mengirimkan dan menerima pesan terkompresi dengan menggunakan metode kompresi *arithmetic coding*. Diharapkan aplikasi ini mampu meningkatkan jumlah karakter yang dikirimkan dibandingkan dengan jumlah karakter standar untuk setiap kali pengiriman SMS. Aplikasi ini dapat dijalankan pada telepon selular yang mendukung Java MIDP 2.0 dan konfigurasi CLDC 1.1.



Gambar 1. Blok Diagram Sistem

Berdasarkan Gambar 1, maka cara kerja sistem dapat dideskripsikan sebagai berikut:

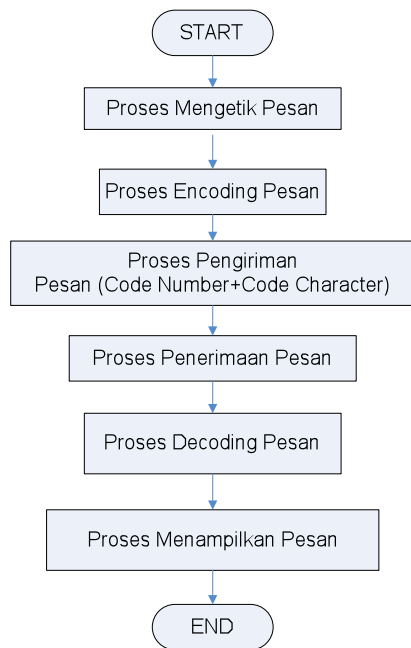
- Pengguna atau pengirim akan mengirimkan SMS kepada penerima melalui telepon selular yang telah ter-*install* aplikasi *ArithmeticSMS*.

- Pengguna menjalankan aplikasi terlebih dahulu dan memilih menu tulis pesan baru. Setelah itu pengirim mengetikkan pesan SMS pada *textbook* yang tersedia dengan menampilkan banyaknya karakter yang sedang diketik pada bagian *title*. Setelah pengguna selesai mengetik pesan maka pengguna dapat melanjutkan proses pengiriman SMS dengan menentukan nomor penerima. Pengguna juga akan diberi informasi mengenai jumlah karakter dan jumlah SMS.
- Ketika pesan tersebut masuk pada telepon selular penerima, aplikasi menampilkan kepada pengguna bahwa terdapat pesan masuk.
- Saat pengguna membaca pesan, sebelumnya dilakukan proses *decoding* pada pesan tersebut dan menampilkan hasil kepada pengguna.

Proses kompresi akan mengubah sebuah pesan menjadi sebuah kode tunggal. Kode tunggal ini merupakan deretan angka yang didapat dari hasil perhitungan berdasarkan probabilitas kemunculan setiap huruf yang sudah ditentukan sebelumnya. Pesan yang diketik pada *TextBox* akan diambil dan disimpan dalam sebuah *String* pesan dan akan diolah untuk mendapatkan kode tunggal sebelum dikirim ke penerima. Proses *encoding* atau kompresi ini menggunakan tabel statik yang sudah ditentukan. Dalam tabel tersebut terdapat 90 karakter yang mungkin muncul dalam sebuah pesan beserta dengan probabilitasnya. Karakter-karakter tersebut adalah karakter A-Z, a-z, 0-9, dan 28 karakter tambahan seperti @, &, * dan lain sebagainya.

String pesan yang diketik pada *textbox* akan dihitung mulai dari karakter pertama sampai karakter terakhir dengan menggunakan nilai probabilitasnya. Nilai probabilitasnya berkisar antara angka 0 sampai 1 dengan *range* yang berbeda untuk tiap karakter. Pada perhitungan sampai pada karakter terakhir maka akan muncul sebuah kode tunggal. *Code Number* dalam tipe data *double* inilah yang akan dikirim ke penerima sebagai *String Message*.

Proses dekompresi dilakukan dengan cara mengambil *String Message* yang dikirimkan dan memisahkan kode pertama yang menunjukkan banyaknya karakter dalam pesan dan kode kedua yang merupakan kode yang akan menunjukkan karakter-karakter pesan yang dikirim. Proses dekompresi ini juga memanfaatkan tabel yang telah ditetapkan. *Flowchart* dari proses yang dilakukan oleh sistem diperlihatkan pada Gambar 2.



Gambar 2. Flowchart Sistem

4. Pengujian dan Analisa

Pengujian dilakukan dengan mengirimkan pesan singkat dengan jumlah karakter diatas 160 sebanyak 20 variasi pengiriman. Pengujian ini dilakukan menggunakan *Mobile Phone* Nokia 6681 serta Emulator Sun Java Wireless Toolkit 2.5.1 dan dilanjutkan dengan pengujian pada telepon selular secara langsung.

Dari pengujian ini diharapkan akan diketahui batas maksimum jumlah karakter yang dapat dikirimkan untuk setiap kali pengiriman SMS. Seperti diketahui batas pengiriman 1 SMS adalah 160 karakter, 2 SMS sebanyak 320 dan 480 karakter untuk 3 SMS.

Untuk setiap pengujian dilakukan pengecekan pulsa sebelum dan sesudah mengirimkan pesan. Hal ini dimaksudkan sebagai verifikasi untuk melihat berapa biaya yang harus dibayarkan untuk SMS yang terkirim. Hasil pengujian diperlihatkan pada Tabel 1.

Tabel 1 Pengujian Pengiriman SMS

PENGUJIAN	JUMLAH KARAKTER YANG DIKIRIM	SMS BIASA	HASIL APLIKASI
1	162	2	1
2	163	2	1
3	165	2	1
4	169	2	1
5	172	2	1
6	176	2	1
7	179	2	2
8	180	2	2
9	190	2	2
10	197	2	2
11	312	2	2
12	321	3	2
13	330	3	2
14	341	3	2
15	352	3	3
16	370	3	3
17	498	4	3
18	500	4	3
19	517	4	3
20	580	4	4

Berdasarkan Tabel 1, terlihat bahwa aplikasi ini mampu meningkatkan kapasitas pengiriman karakter sebanyak maksimum 16 karakter pada pengiriman 1 SMS. Jika semula jumlah karakter yang dapat terkirim pada 1 pengiriman SMS biasa adalah 160 karakter maka aplikasi ini sanggup mengirim sampai dengan 176 karakter.

Pada pengiriman sejumlah 2 SMS, aplikasi ini mampu meningkatkan kapasitas karakter terkirim sebanyak 341 karakter, meningkat dari kapasitas standar sebanyak 320 karakter. Sedangkan pada pengiriman 3 SMS, tercatat mampu dikirimkan sebanyak 517 karakter dibandingkan dengan kapasitas standar sebesar 480 karakter.

5. Kesimpulan

1. Aplikasi yang dibangun mampu dijalankan pada telepon selular dengan dukungan JavaMIDP 2.0.
2. Aplikasi yang dibangun berhasil meningkatkan jumlah karakter pesan singkat (SMS) sebanyak 16 karakter untuk pengiriman 1 SMS, 21 karakter untuk pengiriman 2 SMS dan 37 karakter untuk pengiriman 3 SMS masing-masing dibandingkan dengan konfigurasi normal pengiriman SMS.

PUSTAKA

Howard, P.G., Vitter, J.S., Analysis of Arithmetic Coding for Data Compression, Information Processing and Management, Vol 28, No 6, November 1992

Langdon, Jr., G.G., An Introduction to Arithmetic Coding, IBM J. RES. DEVELOP, Vol 28, No 2, March 1984

Witten, H.I., Neal, R.M., Cleary, J.G., Arithmetic Coding for Data Compression, Communications of the ACM, Vol 3, No 6, June 198