# APLIKASI KOMPRESI TEKS SMS PADA MOBILE DEVICE BERBASIS ANDROID DENGAN MENGGUNAKAN ALGORITMA HUFFMAN KANONIK

**Rozzi Kesuma Dinata[1], Muhammad Al hafizh Hasmar [2]**
[1]Program Studi Teknik Informatika Universitas Malikussaleh,
[2]Program Studi Teknik Informatika, Universitas Malikussaleh
e-mail : rozzikesumadinata@gmail.com

**Abstrak**

Kajian ini membahas mengenai implementasi metode *Huffman Kanonik* dalam penggunaan SMS yang masih memiliki kelemahan yaitu keterbatasan penulisan teks SMS untuk karakter 8-bit yang hanya sebanyak 160 karakter untuk satu kali kirim. Oleh sebab itu di perlukan proses kompresi dan dekompresi pada teks SMS dengan membuat aplikasi *Huffman Compression SMS*, sehingga mampu menjadikan ukuran SMS tersebut seminimal mungkin pada saat di kirimkan. Setiap karakter yang di tuliskan akan di ubah menjadi bentuk bit sesuai dengan tabel statis *Huffman*, rangkaian bit tersebut akan di kirimkan ke nomor tujuan dengan menggunakan mode pengiriman *Binary Message*. Aplikasi ini dibuat dengan menggunakan bahasa android yaitu Eclipse Juno dan berjalan pada *mobile phone* berbasis android 4.1 *Jelly Bean* sampai dengan android 6.0 *Marshmallow*. Pengirim dan penerima bisa berkomunikasi apabila aplikasi sudah terinstal dan di jalankan di *mobile phone* keduanya. Aplikasi ini bertujuan untuk menghemat pemakaian SMS dengan meminimalisasikan ukuran data yang akan di kirimkan. Sehingga dapat menghasilkan rasio penghematan kompresi sekitar 32%.

***Kata Kunci* : *Kompresi, Dekompresi, Huffman Kanonik, SMS, Teks, Rasio.***

## 1. PENDAHULUAN

Seiring perkembangan zaman, kebutuhan manusia akan teknologi informasi dan komunikasi juga meningkat. Salah satu teknologi alat komunikasi yaitu telepon genggam atau dikenal sebagai HP (*handphone*). HP menjadi kebutuhan utama bagi orang yang membutuhkan informasi dan komunikasi di seluruh penjuru dunia. Di Indonesia HP sudah umum di gunakan, penggunanya dari segala usia. HP memiliki fasilitas *standart* komunkasi suara yaitu *telephone* dan SMS (*Short Message Services*), fasilitas tersebut digunakan untuk mengirim pesan pendek berupa teks. Dalam kehidupan sehari-hari, fasilitas SMS menjadi pilihan banyak orang untuk berkomunikasi karena relatif murah, mudah, jelas dan cepat.

Berkembangnya teknologi juga memicu keberadaan data digital yang semakin bertambah besar ukurannya, menyebabkan media penyimpanan digital juga membutuhkan kapasitas yang semakin besar, hal tersebut berdampak pula terhadap kebutuhan media transfer digital yang lebih cepat. Kompresi data menjadi solusi untuk menangani atau setidaknya mengimbangi berkembangnya kebutuhan yang tidak terbendung tersebut. Kompresi sangat berguna ketika data terlalu besar tetapi perlu disimpan dalam tempat yang terbatas ataupun data akan dikirim melalui sebuah saluran komunikasi yang memiliki *bandwidth* terbatas.

Pesan yang dikirimkan dari *sender* ke *receiver* umumnya dengan panjang karakter maksimal adalah 160 karakter untuk setiap satu SMS. Untuk mengirimkan pesan yang lebih dari 160 karakter, maka biaya yang dikeluarkan pun berlipat sesuai jumlah karakter SMS.

Dalam menulis pesan SMS, seorang pengguna biasa melakukannya dengan cara menyingkat isi pesan tersebut. Hal ini dilakukan selain karena kesulitan atau malas untuk mengetikkan isi pesan juga untuk menghemat tempat, sehingga terkadang, isi pesan yang ada, harus di edit untuk menyesuaikan agar dapat termuat dalam satu halaman SMS. Maka dari itu perlu di buat

suatu aplikasi dilakukan dengan cara menyusun teks yang ada (mengkodekan informasi) menggunakan informasi lain yang lebih rendah dari pada representasi data yang tidak terkodekan dengan suatu sistem *enkoding* tertentu. Proses *dekoding* merupakan kebalikan dari *enkoding* yang berarti menyusun kembali data dari hasil *enkoding* menjadi sebuah karakter kembali. Proses *enkoding* dilakukan terhadap data SMS yang akan dikirimkan ke *handphone* tujuan sedangkan *dekoding* dilakukan terhadap data SMS yang diterima oleh *handphone* penerima.

Dengan adanya proses kompresi terhadap teks SMS maka akan terjadi pemampatan terhadap data SMS sehingga dapat menghemat biaya pengiriman SMS (pulsa). Selain itu dengan adanya untuk menambah pemuatan karakter SMS yang akan dikirimkan dalam satu halaman, dengan cara mengkompresi isi pesan atau teks SMS tersebut melalui proses *enkoding* dan *dekoding* terhadap teks SMS dapat memproteksi isi pesan SMS ketika terjadi penyadapan. Hal ini terjadi karena isi pesan akan sulit di baca jika pesan tersebut belum di lakukan proses *dekoding*.

Saat ini terdapat berbagai tipe algoritma kompresi, antara lain: *Huffman, Huffman Kanonik LZ77* dan variannya *(LZ78, LZW, GZIP), Dynamic Markov Compression (DMC), Run-Length, Shannon-Fano, Arithmetic, Block Sorting* dan lain-lain. Kompresi data menjadi sangat penting karena dapat memperkecil kebutuhan penyimpanan data, mempercepat pengiriman data, dan memperkecil kebutuhan *bandwidth.* (Dessyanto, 2013).

## 2. METODE PENELITIAN

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 di lakukan di dalam program. Hal ini di lakukan karena penambahan informasi mengenai *Huffman tree* akan memerlukan tempat tersendiri sehingga proses kompresi menjadi kurang efektif. Hal ini di perkuat dengan hasil penelitian yang menyimpulkan bahwa jika ukuran file yang di kompresi kecil maka file hasil kompresi bisa jadi bukannya semakin kecil tetapi malah semakin besar karena harus menyimpan *Huffman tree* yang di hasilkan.

Untuk menggantikan informasi mengenai *Huffman tree* tersebut maka di buat suatu tabel *Huffman* yang terbentuk dari *tree* itu sendiri dalam bentuk biner, yang berisi kode-kode *Huffman* dari karakter-karakter SMS *default* yang akan di gunakan. Tabel ini bersifat statis dan akan di gunakan oleh aplikasi, baik aplikasi pengirim maupun penerima, sebagai acuan untuk melakukan proses kompresi atau dekompresi terhadap teks SMS. *http://download.portalgaruda.org/ article. php ? article = 59116 & val* di akses 15 Agustus 2015.

## 3. HASIL DAN PEMBAHASAN

### Pengujian kompresi

Sampel pengujian kompresi teks sms yang di gunakan di dalam penelitian ini terdiri dari sepuluh contoh kompresi sms yang akan di kirim kepada penerima. Dalam pengirimannya berupa karakter atau pun berbentuk kalimat yang sering di gunakan di kalangan masyarakat.

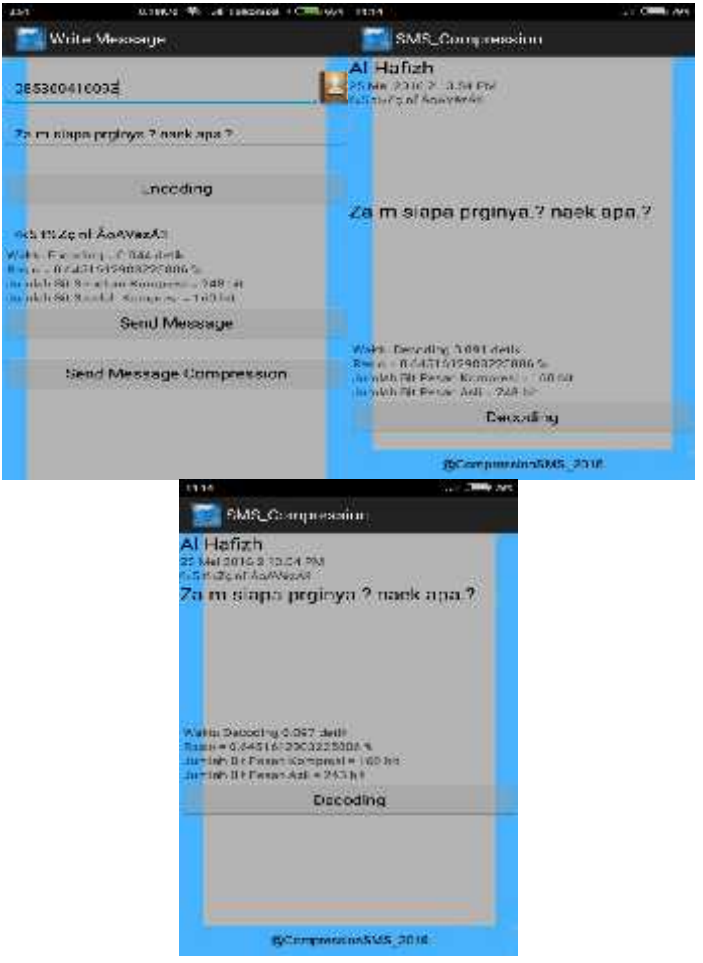
1. Berikut ini merupakan gambar dari pengujian pertama yang di awali dari tulis pesan, kotak masuk dan pesan terkirim yang berisikan kalimat “Muhammad Al hafizh Hasmar”


**Gambar 4.1** Pengujian pertama

2. Sampel pengujian ke dua
   Pesan ini berisikan pesan *default* "Dimana posisi.?"


**Gambar 4.2** Pengujian ke dua

3. Sampel pengujian ke tiga
   Pesan berisikan pesan *default* "Za m siapa prginya.? naek apa.?"




**Gambar 4.3** Pengujian ke tiga

**Pengujian rasio**

Agar dapat memastikan rasio yang sesuai dengan contoh yang telah di tampilkan di atas, maka selanjutnya akan di coba menghitung dengan perhitungan manual.

1. Pengujian pertama

   Teks : Muhammad Al hafizh Hasmar (25 karakter)

Jumlah bit sebelum di kompres : 25 karakter x 8 bit = 200 bit
Jumlah bit setelah di kompres : 136 bit (dari kode *ASCII* dan *Huffman*)
Waktu *encode* : 0,033 detik
Rasio : 136/200 = 0,68 %
= 0,32 %

2. Pengujian ke dua
Teks : Dimana posisi.? (15 karakter)
Jumlah bit sebelum di kompres : 15 karakter x 8 bit = 120 bit
Jumlah bit setelah di kompres : 80 bit (dari kode *ASCII* dan *Huffman*)
Waktu *encode* : 0, 022 detik
Rasio : 80/120 = 0,66 %
= 0,34 %

3. Pengujian ke tiga
Teks : Za m siapa prginya.? naek apa.? (31 karakter)
Jumlah bit sebelum di kompres : 31 karakter x 8 bit = 248 bit
Jumlah bit setelah di kompres : 160 bit (dari kode *ASCII* dan *Huffman*)
Wakktu *encode* : 0,044 detik
Rasio : 160/248 = 0,64 %
= 0,36 %

**Tabel pengujian sistem**

Tabel ini di buat untuk lebih *detail* dalam menganalisa keseluruhan sempel.

Tabel. 3.1 Tabel pengujian

| No Teks | Teks | Jumlah bit sebelum di kompres | Jumlah bit setelah di kompres | Waktu *encode* | Rasio % |
|---|---|---|---|---|---|
| **1.** | **25 karakter** | **200 bit** | **136 bit** | **0,033 detik** | **0,68%** |
| **2.** | **15 karakter** | **120 bit** | **80 bit** | **0,022 detik** | **0,66%** |

| | | | | | |
|---|---|---|---|---|---|
| **3.** | **31 karakter** | **248 bit** | **160 bit** | **0,044 detik** | **0,64%** |
| **4.** | **15 karakter** | **210 bit** | **80 bit** | **0,018 detik** | **0,66%** |
| **5.** | **19 karakter** | **152 bit** | **104 bit** | **0,033 detik** | **0,68%** |
| **6.** | **76 karakter** | **608 bit** | **376 bit** | **0,12 detik** | **0,61%** |
| **7.** | **101 karakter** | **808 bit** | **504 bit** | **0,137 detik** | **0,62%** |
| **8.** | **88 karakter** | **704 bit** | **424 bit** | **0,121 detik** | **0,60%** |
| **9.** | **143 karakter** | **1144 bit** | **728 bit** | **0,189 detik** | **0,67%** |
| **10.** | **145 karakter** | **1160 bit** | **712 bit** | **0,159 detik** | **0,61%** |

## 4. KESIMPULAN

1. Dengan adanya aplikasi kompresi teks sms ini dapat membantu *user* untuk lebih mempermudah dalam penghematan memori dan mempercepat dalam proses pengiriman.
2. Pada aplikasi ini memiliki tiga tahap agar mendapatkan rasio sms yaitu, pohon *huffman kanonik, encoding* dan *decoding.*
3. Semakin banyak karakter yang muncul pada teks sms yang di tulis oleh *user,* maka akan sedikit pula bit yang muncul.
4. Dengan menggunakan tabel statis *Huffman* tabel 3.1 rasio pesan teks yang di tulis dapat menghasilkan pesentase hingga 32% dan terhadap contoh *default* yang di buat, maka rasio pesan *teks* dapat menghasilkan persentase hingga 60%.
5. Proses algoritm*a Huffman Kanonik* pada *smartphone* menunjukkan bahwa susunan soal yang di tampilkan oleh aplikasi seimbang. Tidak ada pengujian yang tidak sesuai dengan rasio yang di hasilkan melalui perhitungan manual.
6. Lemahnya sistem pada saat proses *decoding* di kotak masuk dan di pesan terkirim, mengakibatkan karakter yang muncul tidak dapat terbaca sekaligus di karenakan

karakter yang tertulis melebihi 160 karakter. Dan perlunya menekan tombol *decoding* secara terus menerus agar pesan dapat di baca dengan sempurna.

7. Hasil perhitungan waktu yang telah di terapkan pada aplikasi ini dapat menunjukkan kinerja *smartphone,* atau pun dapat mengetahui dalam pemrorosesan *encoding* dan *decoding* yang apabila semakin banyaknya karakter yang melebihi 160 karakter, maka hasil waktu yang di keluarkan pun semakin lamban. Terlebih lagi apabila aplikasi-aplikasi lain sedang berjalan di *smartphone* tersebut.

## DAFTAR PUSTAKA


Ir. Yuniar Supardi, 2015. *Belajar Coding Android Bagi Pemula.* Penerbit PT Elex Media Komputindo, Jakarta.

Komputer, W, 2012. *Membuat Aplikasi Android Untuk Tablet dan Handphone.* Penerbit PT Elex Media Komputindo, Jakarta.

Komputer, W, 2013. *Android Programming With Eclipse.* Penerbit Andi, Yogyakarta.

Kadir, A, 2014. *From Zero to a Pro Pemograman Aplikasi Android.* Penerbit Andi, Yogyakarta.

Muhammad Sadeli, 2014. *Toko Buku Online Dengan Android.* Penerbit Maxikom. Palembang.

Tim HLIndo Android, 2015. *Aplikasi Android SMS Smart Blast.* Penerbit PT Elex Media Komputindo, Jakarta.