

RANCANGAN DAN IMPLEMENTASI KOMPRESI TEKS SMS (*SHORT MESSAGING SERVICE*) DENGAN MENGGUNAKAN ALGORITMA HUFFMAN

Tjut Awaliyah, M.Kom.¹⁾, Arie Qur'ania, S.Kom.¹⁾ & Indra Gunawan²⁾

¹⁾Staff Pengajar Jurusan Ilmu Komputer FMIPA UNPAK

²⁾Mahasiswa Jurusan Ilmu Komputer FMIPA UNPAK

I. PENDAHULUAN

1.1. Latar Belakang

Pelayanan telepon seluler begitu populer, ini disebabkan tingkat kebebasan, mobilitas dan peningkatan produktivitasnya yang mampu dilayani oleh teknologi telepon seluler ini. Dalam waktu tidak terlalu lama, orang-orang telah meninggalkan telepon tetap maupun telepon umum untuk menangani kebutuhan komunikasi telepon mereka. Proses pemanfaatan telepon seluler dianggap lebih menguntungkan secara finansial, karena dalam telepon seluler terdapat fasilitas yang disebut SMS (*Short Messaging Service*) yaitu komunikasi pesan singkat berupa teks.

Short Messaging Service (SMS) merupakan layanan komunikasi pesan singkat yang paling banyak digemari oleh pengguna telepon seluler, layanan ini mampu menyumbangkan pendapatan kepada operator sekitar Rp 10 triliun pertahun (Asosiasi Telekomunikasi Seluler Indonesia, 2006).

Satu kali proses pengiriman SMS akan menampung maksimal 160 karakter, jika ternyata jumlah karakter yang harus dikirimkan telah melebihi 160 karakter maka *user* harus membayar lebih, padahal kelebihan karakternya mungkin hanya beberapa karakter saja. Dalam menghadapi permasalahan tersebut, maka diperlukan aplikasi yang dapat

Menghemat biaya SMS dengan mengkompres karakter-karakter menjadi lebih kecil.

Dalam melakukan kompresi tersebut dapat dilakukan dengan menggunakan metode kompresi yang cukup populer, yaitu *Huffman*. Metode ini banyak sekali variasinya namun cukup sederhana. *Huffman* dapat dikembangkan dengan menggunakan bahasa pemrograman Java yang sudah sangat populer saat ini. Java yang digunakan adalah J2ME (*Java 2 Micro Edition*) karena sesuai dengan keperluan pengembangan aplikasi-aplikasi yang dapat dijalankan di dalam perangkat kecil.

1.2. Tujuan Penelitian

Tujuan diadakannya penelitian ini adalah untuk merancang dan mengimplementasikan Kompresi teks SMS (*Short Messaging Service*) dengan menggunakan Algoritma Huffman.

1.3. Ruang Lingkup Penelitian

Rancangan dan implementasi kompresi teks SMS (*Short Messaging Service*) dengan menggunakan Algoritma Huffman ini, dibatasi pada sistem operasi *Symbian 6.0* (CLDC 1.0 dan MIDP 2.0) yang implementasi aplikasinya menggunakan pemrograman *JAVA 2 Microedition* (J2ME) yang hanya dapat dijalankan di dalam perangkat-perangkat kecil seperti *Handphone* dan *Personal Digital Asistant* (PDA).

II. PEMBA

2.1. Analisis F

Beber
menjadi perti
suatu metode
kecepatan kon
dibutuhkan (n
ukuran file h
redundansi, da
Tidak ada met
efektif untuk se

Alasan

Huffman dalam
Pertama, pe
kemunculan da
input, lalu me
dalam satu wak
lebih sering m
pendek dibanding
jarang muncul
kompresi H
dibandingkan al
file biner, file mu
file hasil kompr
Panggabean, 200

2.2. Perancang

2.2.1. Perancang

Peranca
bertujuan untuk
karakter yang ak
sehingga mengha
baik. Sebagai con
Mom? (terdapat l
masing memiliki
70 bit) Teks akan d
Huffman:

Tabel 1. Simulasi F

Teks	H	
Badan	1000	00
Uahan	1001000	00
Banyak bit	7	

II. PEMBAHASAN

2.1. Analisis Pemikiran

Beberapa faktor yang sering menjadi pertimbangan dalam memilih suatu metode kompresi yang tepat, yaitu kecepatan kompresi, sumber daya yang dibutuhkan (memori, kecepatan PC), ukuran file hasil kompresi, besarnya redundansi, dan kompleksitas algoritma. Tidak ada metode kompresi yang paling efektif untuk semua jenis file.

Alasan digunakannya *Algoritma Huffman* dalam kompresi teks diantaranya Pertama, penghitungan peluang kemunculan dari tiap simbol dalam file input, lalu mengkodekan satu simbol dalam satu waktu, dimana simbol yang lebih sering muncul diberi kode lebih pendek dibandingkan simbol yang lebih jarang muncul. Kedua adalah hasil kompresi Huffman lebih baik dibandingkan algoritma lain pada kasus file biner, file multimedia, file gambar, dan file hasil kompresi (Linawati, Henry P. Panggabean, 2004).

2.2. Perancangan

2.2.1. Perancangan Program

Perancangan program ini bertujuan untuk menentukan berapa karakter yang akan dilakukan kompresi sehingga menghasilkan hasil yang lebih baik. Sebagai contoh, teks aslinya : **Hallo Mom?** (terdapat 10 karakter yang masing-masing memiliki 7 bit. Sehingga menjadi 70 bit) Teks akan diubah berdasarkan tabel Huffman :

Tabel 1. Simulasi Perancangan Program

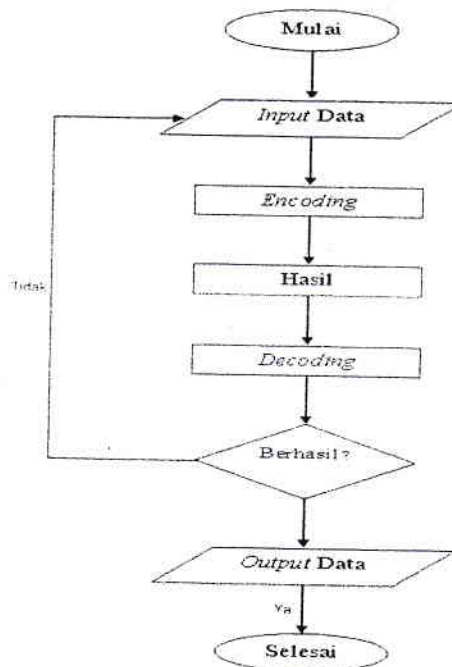
Teks	H	a	l	l	o
Badan	1000	0000	1001	1001	0100
Ubahan	1001000	000000	01001	01001	00100
Banyak bit	7	5	5	5	5

Teks	spasi	M	o	m	?
Badan	1	1010 0	01 00	1010 0	001 0
Ubahan	1011	1001 0100	00 10 0	0101 00	110 001 0
Banyak bit	4	8	5	6	7

Total jumlah bit yang dibutuhkan menjadi 57 bit yang semula 70 bit. Jadi lebih banyak yang bisa diketik dalam satu sms.

2.2.2. Perancangan Sistem

Rancangan sistem mencakup perancangan kompresi teks, proses *encoding*, *decoding*, dan perancangan antarmuka (*User Interface*). Diagram alir dari perancangan program dapat dilihat pada gambar di bawah ini.



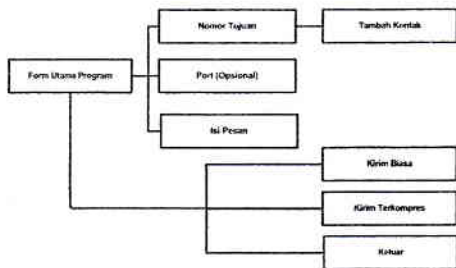
Gambar 1. Diagram Alir (Flowchart) Algoritma Huffman

2.2.3. Perancangan Engine Program

Perancangan *engine* program dimasukkan untuk merancang komponen utama aplikasi "KirimSMS" ini. Komponen utama ini meliputi fungsi kirim sms, kompres karakter, dan terima sms yang akan menjadi input dan output manipulasi jumlah bit.

2.2.4. Perancangan Navigasi Program

Perancangan Navigasi Sistem dilakukan untuk memberikan suatu gambaran mengenai informasi yang akan ditampilkan untuk selanjutnya dijadikan sebagai kerangka kerja dalam pembuatan aplikasi. Navigasi aplikasi ini dapat dilihat pada gambar.



Gambar 2. Navigasi Aplikasi KirimSMS

Form Utama Program memiliki tiga form, Nomor Tujuan, Port, dan Isi Pesan yang akan dikirimkan. Pada Form Nomor Tujuan, *user* dapat mengakses nomor yang akan menjadi tujuan pada dalam buku telepon yang tersimpan dalam ponsel. Form Utama Program juga memiliki Form Pilihan yang terdiri dari Kirim Biasa, Kirim Terkompres dan Keluar.

2.2.5. Perancangan User Interface

Perancangan antar muka pengguna merupakan salah satu bagian yang perlu diperhatikan untuk

perancangan *user interface*. Perancangan ini dipaparkan melalui gambar desain aplikasi. Antar muka dari aplikasi ini dibagi menjadi 3 bagian yaitu Nomor Tujuan, Port (Opsional), dan Isi.

Nomor Tujuan

Port (Opsional)

Isi :

Pilihan

Gambar 2. Rancangan User Interface

2.3. Implementasi Sistem

Implementasi dibuat ke dalam dalam *class-class* pada sistem aplikasi ini. Pada aplikasi ini terdiri dari empat *class* yaitu :

1. Class KirimSMS

Class ini berfungsi memulai semua perintah dalam program, dengan membuat *FormMain* dan *TreadTimer* untuk mulai berjalan. Pada *class* ini juga diatur *port* sms yang akan dikirimkan serta proses penerimaan sms tanpa dikompres. Validasi ini dapat dilihat pada sintaks.

```

{
    TextMessage tm =
    (TextMessage)con.newMess
    age("text", st);

    tm.setPayloadText(formMai
    n.isiPesan.getString());

    m = tm;
}
  
```

Untuk terkompres sintaks

```

if(cmd == form
{
    BinaryMes
    BinaryMessage
    st);
    bm.setPayload
    Main.isiPesan.g
    m
}
  
```

2. Class From Class ini Class Kirim untuk men dan membe Validasi ini Void NDS throw

```

{
    setTitik
    append
    no
    Tujuan");
    nomor
    nomor
    -1);
    append
    noPort
    (opsional)");
    noPort
    append
    isiP
    114);
    isiPesan
    isiPesan
    isiPesan
    AddCo
    addComm
    noPort
    noPort
  
```

Untuk pengiriman pesan yang terkompres maka dapat dilihat pada sintaks

```
if(cmd == formMain.tmbkKirimKompres)
{
    BinaryMessage bm =
    BinaryMessage(con.newMessage("binary",
    st));

    bm.setPayloadData(Kompres.kompres(form
    Main.isiPesan.getString()));
    m = bm;
}
```

2. Class FromMain

Class ini merupakan lanjutan dari Class KirimSMS yang berfungsi untuk menampilkan *User Interface* dan membuat inisialisasi *Exception*. Validasi ini dapat dilihat pada sintaks.

```
VoidNDSInit()
    throws Exception
{
    setTitle("Kirim SMS");
    append(nomorTujuan);
    nomorTujuan.setLabel("Nomor
    Tujuan");
    nomorTujuan.setConstraints(3);
    nomorTujuan.setPreferredSize(-1,
    -1);
    append(noPort);
    noPort.setConstraints(2);
    noPort.setLabel("Port
    (opsional)");
    noPort.setPreferredSize(-1, -1);
    append(isiPesan);
    isiPesan.setPreferredSize(176,
    114);
    isiPesan.setMaxSize(900);
    isiPesan.setLayout(2099);
    isiPesan.setLabel("Isi");
    AddCommand(tmbkKirim);
    addCommand(tmbkKirimKompres);
    noPort.setMaxSize(5);
    noPort.setString("7575");
}
```

3. Class Kompres

Class ini berfungsi untuk meng-convert semua jenis karakter yang di print menggunakan sintaks *PrintStream* ke dalam bentuk *byte*. Dalam *class* ini terjadi dua kali proses, yang pertama *convert* dari bentuk *string* ke *byte*

```
int lenAsli = st.length();
st = st + "11111111";
byte hs[] = new byte[(lenAsli + 7)
/ 8];
for(int i = 0; i < lenAsli; i += 8)
{
    String sebyte = st.substring(i, i
    + 8);
    byte b =
    (byte)Integer.parseInt(sebyte, 2);
    hs[i/8] = b;
}
```

Selanjutnya yang kedua adalah *convert* dari bentuk *byte* kedalam bentuk *string* untuk proses *decompressing*.

```
String hs = "";
for(int i = 0; i < ba.length; i++)
{
    int b = ba[i];
    if(b < 0)
        b += 256;
    String tp = Integer.toBinaryString(b);
    tp = "00000000".substring(0, 8 -
    tp.length()) + tp;
    hs = hs + tp;
}
```

4. Class TulisPanjangIsi

Class ini mengontrol timer dan koneksi yang terdapat didalam aplikasi program. Selain itu juga menghitung panjang karakter yang dituliskan dalam form isi. Berikut potongan kode dari class ini.

```
Form.isiPesan.setLabel("Isi: " + len
+ "(" + seg + ") ->" + len2 + "(" + seg2 + ")", "
+ (100 * len2 * 8) / len / 7 + "%");
```


2.4. Uji Coba Sistem

Uji coba sistem dilakukan setelah pembuatan *class-class* sistem selesai dengan percobaan pada *Emulator Sun Java Wireless toolkit*. Dengan melakukan uji coba ini dapat diketahui kekurangan dari sistem yang telah dibuat sebelumnya, diantaranya apakah sistem berjalan dengan baik, dan apakah sistem yang dibuat sesuai dengan perancangan sistem yang dirancang. Berikut ini adalah beberapa pengujian yang dilakukan terhadap sistem yang dibuat.

2.4.1. Pengujian Struktural

Tahapan ini dilakukan dengan tujuan untuk mengetahui apakah sistem yang sudah dibuat sesuai dengan rancangan yang sudah ada. Jika terjadi kesalahan atau tidak berfungsi, maka proses akan kembali ke tahap implementasi. Pengujian dilakukan dengan cara menjalankan program, lalu menguji struktur dari program, diantaranya:

1. Sambungan dari masing-masing *class* pada program yang di representasikan oleh objek telah berjalan dengan baik
2. Menu menu pada program telah sesuai dengan rancangan pembuatan menu.

2.4.2. Pengujian Fungsional

Setelah melakukan uji coba struktural selanjutnya dilakukan uji coba fungsional. Pada tahap ini dilakukan pengujian yang bertujuan untuk mengetahui apakah uji coba yang dilakukan sudah berjalan dengan baik dan sesuai dengan sistem yang ada. pengujian dilakukan tahap demi tahap sesuai dengan tahapan proses yang dilakukan oleh program. Jika terjadi kesalahan atau

kesalahan atau tidak berfungsi, maka proses akan kembali ke tahap implementasi. Pengujian ini dibagi menjadi dua, yaitu pengujian *User Interface* dan pengujian kompresin

1. Pengujian *User Interface*

Tahapan dari proses pengujian ini adalah menselaraskan antara program dengan *interface* yang berhasil ditampilkan dalam ponsel, sesuai dengan *source code* yang dibuat.

Pilihan

Gambar 3. *User Interface*

```
void NDSinit()
    throws Exception
{
    setTitle("Kirim SMS");
    append(nomorTujuan);
    nomorTujuan.setLabel("Nomor
    Tujuan");
    nomorTujuan.setConstraints(3);
    nomorTujuan.setPreferredSize(-
    1, -1);
}
```

```

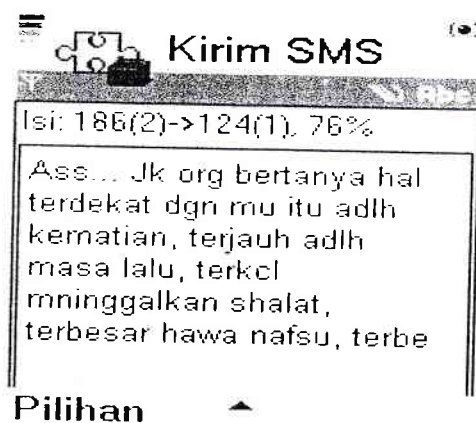
append(noPort);
noPort.setConstraints(2);
noPort.setLabel("Port
(opsional)");
noPort.setPreferredSize(-1, -
1);
append(isiPesan);
isiPesan.setPreferredSize(176,
114);
isiPesan.setMaxSize(900);
isiPesan.setLayout(2099);
isiPesan.setLabel("Isi");
addCommand(tmbkKirim);

addCommand(tmbkKirimKompres);
noPort.setMaxSize(5);
noPort.setString("7575");
}
    
```

2. Pengujian Kompresing

Tahapan ini menguji proses kompresi yang dilakukan sistem ketika karakter mulai diketikan dalam sistem aplikasi. Kompresing akan menghitung jumlah karakter asli sebanyak 7 bit dan karakter terkompres sebanyak 8 bit.

Contoh:
$$\left(\frac{124 \times 8 \times 100}{186} \right) = 76 \%$$



Gambar 4. Hasil Kompresi

Setelah dilakukan pengetikan maka nampak lebih jelas hasil yang diperoleh dari sistem kompresi ini. Isi : 186 menunjukkan jumlah karakter yang sudah diketikan, (2) merupakan tanda bahwa karakter yang diketikan telah melebihi satu kali pengiriman yang hanya 160 karakter. Sedangkan 124(1) merupakan hasil kompresi dari karakter sebelumnya yang mencapai 2 kali pengiriman. Hasil dari kompresi karakter pertama menjadi karakter kedua dilakukan sebanyak 76%.

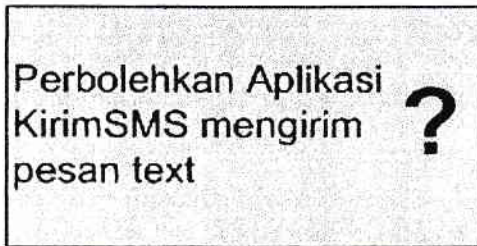
2.4.3. Pengujian Validasi

Tahap ini dilakukan dengan tujuan untuk mengetahui sistem yang dibuat sudah bekerja dengan benar atau tidak. Sama seperti pengujian fungsional pengujian ini dilakukan dengan menguji satu persatu komponen pada aplikasi apakah berfungsi dengan benar sesuai tahapan jalannya program, hanya saja pengujian dilakukan dengan lebih mendalam.

Tahapan ini menguji sejauh mana aplikasi dapat berjalan pada sebuah ponsel, baik ponsel tersebut memiliki aplikasi ataupun tidak. Karena walaupun aplikasi dapat mengirim pesan tanpa *error* dan berjalan, belum tentu bekerja dengan benar. Beruntung *java* menyediakan suatu fitur yang bernama *Handle Exception* atau penanganan error. Fitur ini amat berguna untuk mengatur agar aplikasi tidak *crash* atau mati ketika *error* terjadi. Pengujian ini di bagi menjadi dua, yaitu :

1. Pengujian Pengiriman

Pertama pengujian ini dilakukan dengan memasukan nomor tujuan yang menjadi tujuan pengiriman. Apabila benar maka akan muncul peringatan

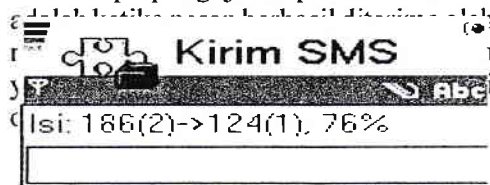


Gambar 5. Peringatan Pengiriman

dan selanjutnya dapat dikirim dengan Kirim Biasa atau Kirim Terkompresi. Kedua, seperti dijelaskan pada bab sebelumnya jika menggunakan port maka pesan yang dikirimkan akan langsung dieksekusi oleh program aplikasi, tetapi jika dikosongkan maka pesan yang sampai akan ditangani oleh aplikasi asli ponselnya.

2. Pengujian Penerimaan

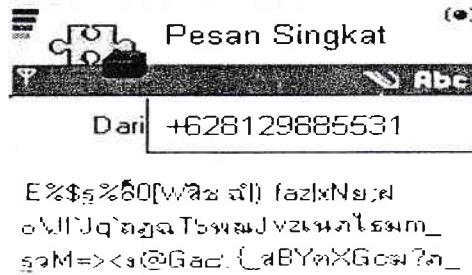
Tahap pengujian penerimaan ini



Pesan dari:
 sms://+6285216282093:7575
 Ass... Jk org bertanya hal
 terdekat dgn mu itu adlh
 kematian, terjauh adlh
 masa lalu, terkcl
 mninggalkan shalat,
 terbesar hawa nafsu. terbe

Pilihan

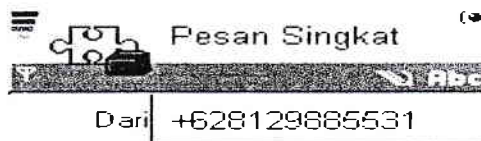
Apabila dalam proses pengiriman menggunakan perintah Kirim Terkompresi, dan ponsel penerima pesan tidak memiliki aplikasi maka pesan yang disampaikan tidak akan sesuai dengan apa yang telah di tulis mengirim.



Pilihan Kembali

Gambar 7. Penerimaan Pesan Tidak Menggunakan Aplikasi KirimSMS

Selain itu, pesan juga tidak dapat ditampilkan pada ponsel penerima yang memakai sistem operasi dibawah *sybian.6.0* karena hanya memiliki karakter-karakter ASCII antara 32 sampai dengan 127.



Pesan tidak dapat ditampilkan

Pilihan Kembali

Gambar 8. Pesan Tidak dapat

Setela pengujian pen dan jumlah p Seperti pada ta

Tabel. 2. Hasil kali Pengiriman

No	Panjang Karakter	SMS Ke-	K
1	15	1	
2	41	1	
3	113	1	
4	151	1	
5	186	2	
6	234	2	
7	299	3	
8	347	3	
9	381	3	
10	458	4	

Setelah melakuk pengiriman, maka persentase pengru 22%.

25. Pengguna Untuk per yang di masukkan yakni : memasukan port, dan pesan ya Apabila pesan yang satu kali pengirim metode pengiriman Namun apabila ha

Setelah melakukan beberapa kali pengujian pengiriman, maka di dapat hasil dan jumlah persentasi yang bervariasi. Seperti pada tabel 2 dibawah ini :

Tabel. 2. Hasil Pengujian dari beberapa kali Pengiriman

No	Panjang Karakter	SMS Ke-	Hasil Kompresi	SMS Ke-	Persentase Kompresi (%)	Pengurangan Memori (%)
1	15	1	11	1	83	17
2	41	1	29	1	80	20
3	113	1	80	1	81	19
4	151	1	103	1	77	23
5	186	2	124	1	76	24
6	234	2	161	2	78	22
7	299	3	204	2	77	23
8	347	3	236	2	77	23
9	381	3	258	3	77	23
10	458	4	298	3	74	26

Setelah melakukan 10 kali percobaan pengiriman, maka dapat diambil rata-rata persentase pengurangan memori sebesar 22%.

2.5. Penggunaan Sistem

Untuk penggunaan sistem, *input* yang di masukkan terdiri dari 3 bagian yakni : memasukan nomor tujuan, nomor port, dan pesan yang akan disampaikan. Apabila pesan yang disampaikan melebihi satu kali pengiriman maka dapat dipilih metode pengiriman Kirim Terkompres. Namun apabila hanya beberapa karakter

saja yang disampaikan maka dapat menggunakan metode pengiriman Kirim Biasa.

III. KESIMPULAN

3.1. Kesimpulan

Rancangan dan implementasi kompresi teks SMS (*Short Messaging Service*) dengan menggunakan algoritma huffman dapat memperkecil jumlah karakter pengiriman SMS yang semula 160 karakter menjadi lebih kecil. Pembuatan aplikasi Kirim SMS ini menggunakan bahasa pemrograman *java*, sehingga dapat dijalankan di hampir semua sistem operasi yang sudah terinstall *Java Runtime Environment*.

Pengujian menggunakan metode uji coba struktural, fungsional, validasi sistem yang dirancang untuk memastikan kinerja sistem sesuai dengan yang diinginkan. Sebagaimana yang telah dijelaskan, jumlah karakter yang sudah diketikkan sebanyak 186 karakter dan mencapai 2 kali pengiriman sms, dapat di kompres sebanyak 124 karakter dan satu kali pengiriman. Hasil dari kompresi karakter pertama menjadi karakter kedua dilakukan sebanyak 76%. Setelah melakukan 10 kali percobaan pengiriman, maka dapat diambil rata-rata persentase pengurangan memori sebesar 22%.

Hasil yang diterima dari proses pengiriman masih menjadi kendala dalam penyampaian pesan, ini dikarenakan pihak penerima harus menggunakan aplikasi yang sama dengan pengirim. Sehingga data yang diterima dapat dibaca/di-dekode kembali dengan benar. Proses yang terjadi dalam aplikasi KirimSMS tidak melibatkan SMSC (*Short Messaging Service Center*) atau Operator.

Selain itu program ini masih dapat dikembangkan lagi dengan menggunakan pesan pengiriman berbasis *Multimedia Message Service* (MMS). Teknik kompresi MMS dapat menggunakan algoritma *RUN-LENGTH* atau algoritma *HALF-BYTE*. Sehingga dapat mengakomodasi data teks/biner, gambar (JPEG, PNG, TIFF), audio (MP3, AAC, RMA, WMA), dan video (MPEG, H261, H263).

DAFTAR PUSTAKA

Aryani, N. 2006. SMS dan Dinamika Komunikasi Masa, <http://www.trendigital.com/03012006/Techno/Techno.htm>

Hartawan, I. Pemampatan Data dengan Menggunakan Algoritma Huffman Kanonik, <http://www.stttelkom.ac.id>

Hartono, A. A. 2007. Aplikasi Penghemat Biaya SMS, <http://kagakribet.com/tutorial>

<http://lecturer.ukdw.ac.id/anton/download/multimedia7.pdf>

<http://www.prepressure/huffmancompression.com/huffman.htm>

<http://www.rajapasar.blogspot.com/2005/06/sejarah-sms.html>

Howe, D. 1993. "Free On-line Dictionary of Computing", <http://www.foldoc.org/>

Setelah pengujian peng... dan jumlah p... Seperti pada tab

Tabel. 2. Hasil kali Pengiriman

No	Panjang Karakter	SMS Ke-	K...
1	15	1	
2	41	1	
3	115	1	
4	151	1	
5	186	2	
6	234	2	
7	299	3	
8	347	3	
9	381	3	
10	458	4	

Setelah melaku... pengiriman, mak... persentase peng... 22%.

2.5. Pengun...

Untuk pe... yang di masukka... yaitu : memasuka... per, dan pesan y... Apabila pesan yan... satu kali pengirim... merde pengirim... Namun apabila h...

Setelah melakukan beberapa kali pengujian pengiriman, maka di dapat hasil dan jumlah persentasi yang bervariasi. Seperti pada tabel 2 dibawah ini :

Tabel. 2. Hasil Pengujian dari beberapa kali Pengiriman

No	Panjang Karakter	SMS Ke-	Hasil Kompresi	SMS Ke-	Persentase Kompresi (%)	Pengurangan Memori (%)
1	15	1	11	1	83	17
2	41	1	29	1	80	20
3	113	1	80	1	81	19
4	151	1	103	1	77	23
5	186	2	124	1	76	24
6	234	2	161	2	78	22
7	299	3	204	2	77	23
8	347	3	236	2	77	23
9	381	3	258	3	77	23
10	458	4	298	3	74	26

Setelah melakukan 10 kali percobaan pengiriman, maka dapat diambil rata-rata persentase pengurangan memori sebesar 22%.

2.5. Penggunaan Sistem

Untuk penggunaan sistem, *input* yang di masukkan terdiri dari 3 bagian yakni : memasukan nomor tujuan, nomor port, dan pesan yang akan disampaikan. Apabila pesan yang disampaikan melebihi satu kali pengiriman maka dapat dipilih metode pengiriman Kirim Terkompres. Namun apabila hanya beberapa karakter

saja yang disampaikan maka dapat menggunakan metode pengiriman Kirim Biasa.

III. KESIMPULAN

3.1. Kesimpulan

Rancangan dan implementasi kompresi teks SMS (*Short Messaging Service*) dengan menggunakan algoritma huffman dapat memperkecil jumlah karakter pengiriman SMS yang semula 160 karakter menjadi lebih kecil. Pembuatan aplikasi Kirim SMS ini menggunakan bahasa pemrograman *java*, sehingga dapat dijalankan di hampir semua sistem operasi yang sudah terinstall *Java Runtime Environment*.

Pengujian menggunakan metode uji coba struktural, fungsional, validasi sistem yang dirancang untuk memastikan kinerja sistem sesuai dengan yang diinginkan. Sebagaimana yang telah dijelaskan, jumlah karekter yang sudah diketikan sebanyak 186 karakter dan mencapai 2 kali pengiriman sms, dapat di kompres sebanyak 124 karakter dan satu kali pengiriman. Hasil dari kompresi karakter pertama menjadi karakter kedua dilakukan sebanyak 76%. Setelah melakukan 10 kali percobaan pengiriman, maka dapat diambil rata-rata persentase pengurangan memori sebesar 22%.

Hasil yang diterima dari proses pengiriman masih menjadi kendala dalam penyampaian pesan, ini dikarenakan pihak penerima harus menggunakan aplikasi yang sama dengan pengirim. Sehingga data yang diterima dapat dibaca/di-dekode kembali dengan benar. Proses yang terjadi dalam aplikasi KirimSMS tidak melibatkan SMSC (*Short Messaging Service Center*) atau Operator.

Selain itu program ini masih dapat dikembangkan lagi dengan menggunakan pesan pengiriman berbasis *Multimedia Message Service* (MMS). Teknik kompresi MMS dapat menggunakan algoritma *RUN-LENGTH* atau algoritma *HALF-BYTE*. Sehingga dapat mengakomodasi data teks/biner, gambar (JPEG, PNG, TIFF), audio (MP3, AAC, RMA, WMA), dan video (MPEG, H261, H263).

DAFTAR PUSTAKA

- Aryani, N. 2006. SMS dan Dinamika Komunikasi Masa, <http://www.trendigital.com/03012006/Techno/Techno.htm>
- Hartawan, I. Pemampatan Data dengan Menggunakan Algoritma Huffman Kanonik, <http://www.stttelkom.ac.id>
- Hartono, A. A. 2007. Aplikasi Penghemat Biaya SMS, <http://kagakribet.com/tutorial>
- <http://lecturer.ukdw.ac.id/anton/download/multimedia7.pdf>
- <http://www.prepressure/huffmancompression.com/huffman.htm>
- <http://www.rajapasar.blogspot.com/2005/06/sejarah-sms.html>
- Howe, D. 1993. "Free On-line Dictionary of Computing", <http://www.foldoc.org/>

KRISTA

Bagian
Kelompok

L PENDAHULUAN

Kristal fotonik adalah suatu struktur periodik dari bahan dielektrik yang memiliki indeks bias yang berbeda-beda. Sebuah kristal fotonik dapat diibaratkan sebagai kristal dengan elektron digantikan oleh foton. Kristal fotonik memiliki celah fotonik yang dapat mencegah foton dengan energi tertentu untuk bergerak melalui kristal. Konsep ini pertama kali diusulkan oleh Lord Rayleigh pada tahun 1890-an. Kemudian, pada tahun 1980-an, John Joannopoulos dan Eli Yablonovitch melakukan penelitian mendalam tentang kristal fotonik. Mereka menunjukkan bahwa kristal fotonik dapat memiliki celah fotonik yang lebar, yang dapat mencegah foton dengan energi tertentu untuk bergerak melalui kristal. Penelitian ini membuka jalan bagi pengembangan berbagai perangkat fotonik, seperti filter, pemisah, dan pemroses sinyal. Saat ini, kristal fotonik telah menjadi salah satu bidang penelitian yang sangat aktif dalam teknologi fotonik. Dengan integrasi kristal fotonik dengan teknologi lain, seperti serat optik dan perangkat elektronik, diharapkan akan tercipta berbagai aplikasi yang revolusioner di masa depan.