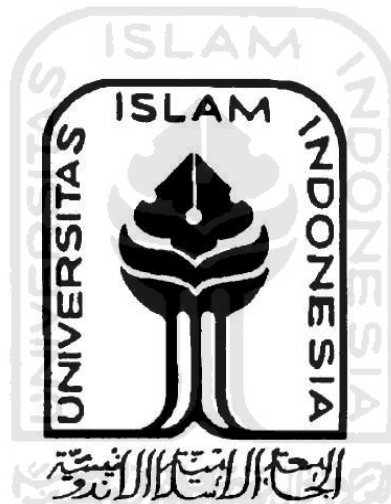


**PENGEMBANGAN APLIKASI KOMPRESI TEKS
MENGUNAKAN ALGORITMA *DYNAMIC MARKOV*
COMPRESSION (DMC) PADA LAYANAN SMS**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana
Jurusan Teknik Informatika



DISUSUN OLEH:

GERIYAN YUNIADI

04 523 364

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2011

LEMBAR PENGESAHAN PEMBIMBING

**PENGEMBANGAN APLIKASI KOMPRESI TEKS
MENGUNAKAN ALGORITMA *DYNAMIC MARKOV
COMPRESSION* (DMC) PADA LAYANAN SMS**

TUGAS AKHIR



DISUSUN OLEH :

GERIYAN YUNIADI

04 523 364

Yogyakarta, 5 November 2011

**Menyetujui,
Dosen Pembimbing Tugas Akhir**

(Yudi Prayudi, S.Si, M.Kom)

LEMBAR PENGESAHAN PENGUJI

**PENGEMBANGAN APLIKASI KOMPRESI TEKS
MENGUNAKAN ALGORITMA *DYNAMIC MARKOV
COMPRESSION* (DMC) PADA LAYANAN SMS**

Disusun Oleh:

GERIYAN YUNIADI

04 523 364

Telah Dipertahankan di Depan Sidang Penguji

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana

Jurusan Teknik Informatika Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta,

Tim Penguji:

Yudi Prayudi, S.Si., M.Kom.

Ketua

Izzati Muhimmah, ST., M.Sc., Ph.D.

Anggota I

Zainuddin Zuhri, ST., MIT.

Anggota II

Mengetahui,

Ketua Jurusan Teknik Informatika

Fakultas Teknologi Industri Universitas Islam Indonesia

Yudi Prayudi, S.Si, M.Kom

HALAMAN PERSEMBAHAN

Allah SWT., kepada-Nya aku menyerahkan

tiap hembusan nafas dan tiap langkah

tempatku mengabdikan dan memohon pertolongan.

Sang Utusan-Nya, kepadanya saya panjatkan Salam dan Salawat

Yang membuat saya mengenal ALLAH sang Raja Alam Semesta

Yang memperkenalkan Jalan Kebenaran kepada hamba-hamba-Nya

Dan dengan izin ALLAH SWT., saya dapat menyelesaikan Tugas Akhir ini.

Kupersembahkan karya kecilku ini untuk kedua orang tuaku

Bapak Djoko Santoso dan Ibu Irianni, SH. tersayang

Kepada Saudara-saudara kandungku tercinta

Sigit, Indah dan Dewo

Yang selalu memberikan dorongan semangat dan materi

Kepada Adek ku tersayang Tertia Permatasari

Yang selalu memberikan bantuan yang berarti, Cinta, Kasih dan semangat

HALAMAN MOTO

﴿ شَرَعَ لَكُمْ مِنَ الدِّينِ مَا وَصَّى بِهِ نُوحًا وَالَّذِي أَوْحَيْنَا إِلَيْكَ وَمَا وَصَّيْنَا بِهِ إِبْرَاهِيمَ وَمُوسَى وَعِيسَى أَنْ أَقِيمُوا الدِّينَ وَلَا تَتَفَرَّقُوا فِيهِ كَبُرَ عَلَى الْمُشْرِكِينَ مَا تَدْعُوهُمْ إِلَيْهِ اللَّهُ يَجْتَبِي إِلَيْهِ مَنْ يَشَاءُ وَيَهْدِي إِلَيْهِ مَنْ يُنِيبُ ﴾

Dia Telah mensyari'atkan bagi kamu tentang Agama yang telah diwasiatkan-Nya kepada Nuh dan apa yang telah kami wahyukan kepadamu dan apa yang telah kami wasiatkan kepada Ibrahim, Musa dan Isa yaitu: Tegakkanlah Agama dan janganlah kamu berpecah belah tentangnya. Amat berat bagi orang-orang musyrik Agama yang kamu seru mereka kepadanya. Allah menarik kepada Agama itu orang yang dikehendaki-Nya dan memberi petunjuk kepada Agama-Nya bagi orang yang kembali kepada-Nya (Qs. As-Syuro: 13)

﴿ قُلْ إِنِّي هَدَىٰ رَبِّي إِلَىٰ صِرَاطٍ مُسْتَقِيمٍ دِينًا قِيمًا مِّلَّةَ إِبْرَاهِيمَ حَنِيفًا وَمَا كَانَ مِنَ الْمُشْرِكِينَ ﴾

Buktikanlah : "Sesungguhnya Aku Telah ditunjuki oleh Robb-ku kepada jalan yang lurus, (yaitu) Agama yang benar, agama Ibrahim yang lurus, dan Ibrahim itu bukanlah termasuk orang-orang musyrik". (QS. Al-An'am : 161)

﴿ قُلْ إِنَّ صَلَاتِي وَنُسُكِي وَمَحْيَايَ وَمَمَاتِي لِلَّهِ رَبِّ الْعَالَمِينَ ﴾

﴿ ١٦٢ ﴾

Buktikanlah : Sesungguhnya sholatku, ibadatku, hidupku dan matiku hanyalah untuk Allah, Robb semesta alam. (QS. Al-An'am : 162)

KATA PENGANTAR



Assalamu'alaikum wr.wb.

Syukur Alhamdulillah penulis panjatkan kehadiran Allah SWT, yang telah memberikan rahmat, hidayah, dan karunia-Nya, sehingga laporan Tugas Akhir dapat penulis selesaikan. Tak lupa shalawat serta salam kami haturkan kepada junjungan kita Nabi Muhammad S.A.W, yang telah memberi uswatun khasanah bagi umat manusia.

Tugas Akhir ini dibuat sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di jurusan Teknik Informatika Universitas Islam Indonesia.

Tugas Akhir yang dilaksanakan penulis adalah Pengembangan Aplikasi Kompresi Teks Menggunakan Algoritma Dynamic Markov Compression pada Layanan SMS.

Dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Ayah, ibu, saudara dan keluarga tercinta, atas dorongan dan doanya.
2. Bapak Yudi Prayudi, S.Si., M.Kom, selaku ketua Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia dan juga sekaligus selaku dosen pembimbing Tugas Akhir, yang telah memberikan pengarahan dan bimbingan selama pelaksanaan Tugas Akhir dan penulisan laporan.

3. Seluruh staf pengajar FTI UII, khususnya dosen-dosen jurusan Teknik Informatika yang telah memberikan bekal ilmu.
4. Seluruh teman-teman angkatan 2004 Teknik Informatika yang telah memberikan keceriaan dan semangat tanpa henti kepada penulis.
5. Serta semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah membantu dari awal hingga akhir.

Penulis menyadari bahwa laporan ini masih belum sempurna, karena keterbatasan kemampuan dan pengalaman. Oleh karena itu penulis mengharapkan kritik dan saran membangun untuk membantu penulis di masa yang akan datang.

Akhir kata penulis berharap agar laporan ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum Wr Wb.

Yogyakarta, 5 November 2011

Penulis

SARI

Laporan Tugas Akhir ini membahas perancangan dan hasil pengembangan metode kompresi teks pada aplikasi SMSCompress. SMSCompress adalah sebuah aplikasi kompresi teks pada layanan SMS (Short Message Service) yang diperuntukkan pada perangkat mobile. Pengembangan aplikasi ini berbasiskan teknologi J2ME (Java 2 Micro Edition) yang dirancang khusus untuk pengembangan aplikasi pada perangkat mobile seperti pada telepon selular.

Pengembangan aplikasi *SMSCompress* merupakan implementasi dari metode kompresi yang rumit dari konsep yang telah dikembangkan oleh Cormack dan Horspool yaitu metode *Dynamic Markov Compression* (DMC). Metode ini merupakan metode *Predictive Based* yang menggunakan model *finite-context* atau *finite-state* untuk memprediksi distribusi probabilitas dari simbol-simbol selanjutnya. 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 sehingga dapat menyingkat atau mengurangi kapasitas teks SMS yang akan dikirim. Proses encoding dengan kode *Dynamic Markov Compression* merupakan kode yang dibentuk berdasarkan frekuensi kekerapan yang muncul untuk suatu karakter. Karakter dengan kekerapan lebih besar akan memiliki kode yang lebih pendek dibandingkan karakter dengan kekerapan kecil.

Aplikasi ini mempunyai beberapa pilihan dalam mengirimkan teks SMS. User dapat memilih apabila SMS yang akan dikirimkan berupa teks SMS biasa tanpa kompresi atau dengan menggunakan kompresi. User juga dapat mengetahui perbandingan rasio hasil kompresi.

Kata Kunci: *kompresi, dekompresi, decoding, encoding, SMS, algoritma, Dynamic Markov Compression, Aplikasi J2ME.*

TAKARIR

<i>Kompresi</i>	Proses pemampatan data berupa teks, grafik dll. sehingga memiliki kapasitas data yang lebih kecil.
<i>Dekompresi</i>	Proses meng-ekstrak/mengembalikan data yang telah dikompresi menjadi data aslinya sebelum dilakukan kompresi. (kebalikan kompresi).
<i>Encoding</i>	Proses mengkonversi suatu data menjadi bit-bit data
<i>Decoding</i>	Proses mengkonversi suatu bit-bit data menjadi data aslinya sebelum (kebalikan encoding)
<i>SMS</i>	Layanan pengiriman pesan singkat yang pada awalnya dirancang pada jaringan GSM.
<i>Port</i>	Koneksi yang digunakan oleh perangkat untuk berkomunikasi dengan mesin
<i>UML</i>	Bahasa permodelan untuk memodelkan PBO.
<i>Algoritma</i>	Semua prosedur / urutan langkah yang jelas dan diperlukan untuk menyelesaikan permasalahan.
<i>Enkripsi</i>	Proses penyandian data menjadi bit-bit data yang bertujuan untuk meningkatkan keamanan data
<i>Dekripsi</i>	Kebalikan dari enkripsi, yaitu proses menencoding kembali hasil enkripsi.

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
HALAMAN PERSEMBAHAN.....	iv
HALAMAN MOTTO.....	v
KATA PENGANTAR.....	vi
SARI.....	viii
TAKARIR.....	ix
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Metodologi Penelitian.....	4
1.6.1 Metode Pengumpulan Data.....	4
1.6.2 Metode Pengembangan Sistem.....	5
1.7 Sistematika Penulisan.....	6
BAB II LANDASAN TEORI.....	8
2.1 Tinjauan Pustaka.....	8
2.2 Dasar Teori.....	8
2.2.1 Teknologi Short Message Service (SMS).....	8

2.2.2 Konsep Kompresi Data.....	17
2.2.3 Perangkat Lunak yang digunakan.....	26
2.2.4 Pengujian Hasil Kompresi.....	29
BAB III METODOLOGI.....	31
3.1 Analisis Kebutuhan Perangkat Lunak.....	31
3.1.1 Metode Analisis.....	31
3.1.2 Hasil Analisis	31
3.2 Perancangan Perangkat Lunak.....	33
3.2.1 Metode Perancangan.....	33
3.2.2 Hasil Perancangan.....	34
BAB IV HASIL DAN PEMBAHASAN.....	44
4.1 Implementasi Perangkat Lunak.....	44
4.1.1 Batasan Implementasi.....	44
4.1.2 Implementasi.....	45
4.2 Analisis Kerja Perangkat Lunak.....	56
4.2.1 Pengujian Perangkat Lunak.....	56
BAB V KESIMPULAN DAN SARAN.....	63
5.1 Kesimpulan.....	63
5.2 Saran.....	64
DAFTAR PUSTAKA.....	65

DAFTAR TABEL

Tabel 2.1. <i>SMS Technology Fact Sheets</i>	12
Tabel 2.2. Perkembangan Teknologi <i>Messaging</i>	12
Tabel 2.3. Contoh Tabel Probabilitas	23
Tabel 2.4. Tabel Perbandingan CLDC dan CDC.....	28
Tabel 4.1. Tabel Perbandingan Jumlah Karakter SMS.....	62
Tabel 4.2. Rata-rata Range Jumlah Karakter SMS	62



DAFTAR GAMBAR

Gambar 2.1. Jaringan GSM dengan SMSC	14
Gambar 2.2. Skenario SMS MO (Mobile Originating)	15
Gambar 2.3. Skenario MS dari ESME (SMS Terminating)	16
Gambar 2.4. Model Awal DMC	20
Gambar 2.5. Model yang diciptakan DMC.....	21
Gambar 2.6. Struktur Pengembangan Java Application	27
Gambar 3.1 Use Case Diagram.....	35
Gambar 3.2 Class Diagram	37
Gambar 3.3 Sequence Diagram Pesan Baru	38
Gambar 3.4 Sequence Diagram Kompresi SMS.....	38
Gambar 3.5 Sequence Diagram Tampil Inbox.....	39
Gambar 3.6 Sequence Diagram Tampil Draft	39
Gambar 3.7 Activity Diagram.....	40
Gambar 3.8. Rancangan Antarmuka Halaman Utama Aplikasi	40
Gambar 3.9. Rancangan Antarmuka Menu Buat Pesan Baru	41
Gambar 3.10. Rancangan Antarmuka Input Isi pesan	41
Gambar 3.11. Rancangan Antarmuka Menu Inbox	42
Gambar 3.12. Rancangan Antarmuka Menu Draft	42
Gambar 3.13. Rancangan Antarmuka Help	42
Gambar 3.14. Rancangan Antarmuka About	43
Gambar 4.1 Mekanisme Pengiriman dalam berbagai mode pesan.	47
Gambar 4.2 Script Implementasi Tabel Probabilitas.	49
Gambar 4.3 konversi code number dan string ke byte.....	50
Gambar 4.4 Antarmuka Halaman Utama.....	51
Gambar 4.5 Antarmuka Submenu Pesan Baru	52
Gambar 4.6 Perbandingan mode Normal dan mode Terkompresi.....	52
Gambar 4.7 Halaman Laporan Hasil Kompresi.....	53
Gambar 4.8 Form Inputan Nomor Telepon	53
Gambar 4.9 Menu Daftar Inbox	54
Gambar 4.10 Menu Daftar Drafr	54
Gambar 4.11 Halaman Help.....	55
Gambar 4.12 Halaman About	55
Gambar 4.13 Pengujian Pertama Kompresi	57
Gambar 4.14 Pengujian Kedua Kompresi	58
Gambar 4.15 Pengujian Ketiga Kompresi	59
Gambar 4.16 Pengiriman Pesan Berhasil.....	60
Gambar 4.17 Pengiriman Pesan Gagal	60

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Kebutuhan manusia modern yang haus informasi telah memacu semangat para pengembang teknologi informasi untuk berlomba-lomba mencoba mengimbangi tuntutan tersebut. Permasalahan yang penting dalam dunia teknologi informasi adalah bagaimana cara mengolah data dari informasi-informasi yang semakin besar dan kompleks tersebut, sehingga lebih cepat, mudah, aman, dan efisien baik dalam proses penyimpanannya maupun transfer data.

Salah satu cara yang digemari orang saat ini adalah teknologi sms, selain karena tarifnya yang murah cara penggunaannya pun mudah. Dengan mengetikkan pesan kepada orang yang diinginkan dan mengirimkannya, mereka sudah bisa berkomunikasi. Namun penggunaan teknologi ini pun masih memiliki kekurangan. Salah satunya adalah apabila ingin mengirimkan pesan melebihi jumlah karakter yang telah ditentukan oleh ponsel maka harus membayar tarif yang lebih.

Untuk mengatasi permasalahan di atas salah satu solusinya adalah dengan cara mengompres data informasi tersebut sehingga berukuran lebih kecil dari ukuran semula dengan tanpa mengurangi isi dari data tersebut.

Kompresi merupakan proses *encoding* data menggunakan jumlah bit yang lebih kecil, sehingga bit yang lebih kecil tersebut dapat merepresentasikan informasi yang sama. Kompresi teks atau data akan memperkecil jumlah memori yang digunakan dan mempercepat tercapainya proses informasi. Aplikasi kompresi data atau teks dalam layanan SMS dengan menggunakan atau mengimplementasikan Algoritma *Dynamic Markov Compression* (DMC) merupakan salah satu cara yang efektif untuk melakukan kompresi.

Algoritma DMC merupakan teknik pemodelan yang didasarkan pada model *finite-state* (model Markov) yang memiliki rasio kompresi yang sangat baik. Metode ini dimulai dengan memprediksi suatu kode bit dengan menggunakan kode *arithmetic coding*. Proses ini dilanjutkan dengan melakukan proses *encoding* teks SMS terhadap karakter tersebut sesuai dengan tabel yang dibuat. Setelah tersampaikan, proses *decoding* dilakukan agar pesan dapat dibaca kembali.

Pembuatan aplikasi kompresi SMS ini diharapkan dapat membantu masyarakat umum sehingga pengguna teknologi komunikasi dapat menikmati layanan SMS yang lebih murah.

1.2. Rumusan Masalah

Berdasarkan uraian dari latar belakang masalah di atas, dapat dirumuskan sebuah permasalahan yang dapat dijadikan acuan dalam pembuatan program adalah bagaimana menerapkan atau mengimplementasikan algoritma *Dynamic*

Markov Compression (DMC) dalam aplikasi kompresi teks SMS. Sehingga aplikasi ini dapat memudahkan pengguna Layanan SMS pada perangkat *mobile* secara ekonomis dan dapat melakukan kompresi teks dengan maksimal.

1.3. Batasan Masalah

Pembatasan masalah dalam suatu penelitian sangat diperlukan agar penelitian lebih terarah, dan memudahkan dalam pembahasan sehingga tujuan penelitian dapat tercapai. Beberapa batasan yang digunakan dalam penelitian ini adalah :

1. Aplikasi yang akan dibangun adalah aplikasi kompresi teks pada layanan SMS yang dikembangkan dengan *platform* berbasis Java (J2ME).
2. Aplikasi ini merupakan pengembangan metode kompresi dengan algoritma *Dynamic Markov Compression* (DMC) berbasis *predictive*.
3. Kemampuan aplikasi meliputi pengiriman teks SMS dengan 2 pilihan pengiriman yaitu menggunakan teks biasa tanpa dikompresi dan menggunakan kompresi, sedangkan pada proses penerimaan SMS, aplikasi dapat membaca teks SMS terkompresi dan teks SMS biasa, serta juga dapat melakukan penyimpanan teks SMS.
4. Metode ini diujikan untuk mengkompresi dan mendekompresi pesan SMS dengan uji kinerja berdasarkan faktor-faktor yaitu : rasio/perbandingan ukuran file hasil kompresi terhadap file asli (pesan SMS menggunakan aplikasi kompresi dengan SMS biasa tanpa di kompresi) dan faktor kompresi (akurasi file sebelum kompresi dan sesudah dekompresi).

1.4. Tujuan Penelitian

Adapun maksud dan tujuan dari penelitian ini adalah :

1. Membangun aplikasi kompresi teks pada layanan SMS sehingga pengguna aplikasi mendapatkan kemudahan secara ekonomis dalam melakukan komunikasi.
2. Sebagai uji implementasi keberhasilan dan kehandalan pengembangan metode algoritma DMC dalam melakukan kompresi teks SMS.

1.5. Manfaat Penelitian

Manfaat penelitian ini diharapkan dapat memberikan bantuan bagi pengguna telepon seluler dalam mendapatkan layanan SMS yang lebih ekonomis.

1.6. Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini meliputi Pengumpulan Data dan Pengembangan Perangkat Lunak.

1.6.1. Pengumpulan Data

Adalah dengan mengumpulkan data dari buku-buku referensi yang relevan dengan permasalahan yang dihadapi, mempelajari dokumen, artikel, laporan penelitian, situs internet, tabloid ataupun majalah.

1.6.2. Pengembangan Perangkat Lunak

Pada penelitian ini, untuk mencapai hasil yang baik dalam merancang program, maka metodologi yang digunakan adalah:

a. Analisis Kebutuhan Perangkat Lunak

Yaitu melakukan analisis terhadap permasalahan yang ada untuk lebih mendapatkan gambaran yang luas mengenai sistem yang dibutuhkan dalam mengatasi permasalahan tersebut. Sistem yang dibutuhkan tersebut lebih terarah ke perangkat lunak yang melibatkan *user*.

b. Perancangan Perangkat Lunak

Tahap ini merupakan tahap penerjemahan kebutuhan atau data yang telah dianalisis, ke dalam bentuk yang mudah dimengerti oleh *user*. Pada perancangan sistem ini, peneliti melakukan observasi terhadap data-data yang diperlukan berdasarkan sumber-sumber yang berkaitan dengan penelitian. Dalam hal ini, peneliti menentukan perancangan proses, perancangan *input* dan *output*, serta *interface*.

c. Implementasi Perangkat Lunak

Pada tahap ini, aplikasi dibuat menggunakan *platform Java (J2ME)* untuk perancangan dan pembuatan elemen-elemen inti maupun tambahan.

d. Pengujian Perangkat Lunak

Setelah aplikasi selesai dibuat, maka pada tahap ini merupakan tahap uji coba terhadap aplikasi tersebut. Pengujian ini dapat dilakukan dengan menggunakan kondisi-kondisi berbeda untuk menciptakan suatu aplikasi

yang interaktif sesuai, dengan kebutuhan pengguna dan spesifikasi yang ditentukan sebelumnya.

1.7. Sistematika Penulisan

Dalam penyusunan laporan Tugas Akhir, disusun per bab dan berurutan untuk mempermudah pembahasannya. Secara garis besar, sistematika penulisan dari pokok permasalahan terdiri atas lima bab, yaitu:

BAB I PENDAHULUAN

Diawali dengan penjelasan mengenai latar belakang masalah, kemudian dilanjutkan dengan rumusan masalah, batasan masalah, maksud dan tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Membahas lebih dalam mengenai landasan teori yang dapat membantu para pembaca dalam memahami implementasi yang akan dilakukan. Landasan teori tersebut mencakup tentang *SMS (Short Message Service)*, *Algoritma Dynamic Markov Compression (DMC)* dan prinsip dasar aplikasi yang dibuat.

BAB III METODOLOGI

Bagian ini terdiri dari analisis kebutuhan perangkat lunak dan perancangan sistem. Analisis kebutuhan perangkat lunak membahas tentang metode-metode analisis yang dipakai pada analisis kebutuhan perangkat lunak dan pemilihan kebutuhan dalam pembuatan antarmuka (*interface*).

Sedangkan perancangan sistem membahas tentang metode-metode yang dipakai dalam perancangan perangkat lunak berupa perancangan berorientasi objek dan hasil perancangan merupakan terjemahan kebutuhan perangkat lunak yang berupa arsitektur atau gambaran kasar mengenai perangkat lunak yang akan dibangun dan kebutuhan antarmuka.

BAB IV HASIL DAN PEMBAHASAN

Berisi hasil implementasi dan analisa kinerja. Hasil implementasi menjelaskan tentang implementasi perangkat lunak yang dibangun pada sistem nyata, dan diimplementasikan sesuai kebutuhan pengguna.

Sedangkan analisis kinerja berisi dokumentasi hasil pengujian terhadap perangkat lunak yang dibandingkan kebenaran dan kesesuaiannya dengan kebutuhan pengguna untuk kemudian dianalisa.

BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari hasil implementasi dan analisa kinerja yang telah dibangun. Serta saran-saran yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan dari perangkat lunak yang telah dibangun.

Bagian akhir

Bagian akhir terdiri dari daftar pustaka.

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian ini merupakan bentuk lain dari berbagai aplikasi kompresi teks SMS dengan mengimplementasikan salah satu metode algoritma, yang bertujuan untuk menguji dan menemukan metode yang paling tepat untuk efektifitas dan efisiensi teknologi SMS.

Penelitian ini juga merupakan pengembangan dari penelitian sebelumnya dengan judul “Implementasi Metode *Lempel Ziv Welch* (LZW) dalam aplikasi kompresi teks pada layanan sms” yang disusun oleh Ibnu hajar S.Komp dengan melakukan kompresi terhadap SMS untuk penghematan biaya SMS dan Karakter. Dalam aplikasi ini digunakan metode kompresi yang khusus. Teknik kompresinya disusun oleh Jacob Ziv, Abraham Lempel dan Terry Welch yang dipublikasikan melalui jurnal bergengsi *IEEE Transactions on Information Theory* berjudul “*A Universal Algorithm for Sequential Data Compression*” tahun 1977 dan “*A Technique for High-Performance Data Compression*” tahun 1984.

2.2. Dasar Teori

2.2.1. Teknologi *Short Message Service* (SMS)

Layanan pesan singkat atau *Short Message Service* (SMS) adalah sebuah layanan yang digunakan pada suatu telepon genggam (*handphone*) untuk

mengirim atau menerima pesan-pesan pendek. Awalnya, SMS dirancang sebagai bagian daripada jaringan GSM, sekarang hal ini telah berkembang ke jaringan bergerak lainnya termasuk jaringan UMTS [SHA06]. SMS dapat dikirimkan ke perangkat Stasiun Seluler Digital lainnya hanya dalam beberapa detik selama berada pada jangkauan pelayanan GSM. SMS berkembang menjadi salah satu layanan yang cukup diminati oleh penggunanya.

2.2.1.1. Deskripsi Umum

Short Message Service (SMS) merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel (*wireless*), memungkinkan dilakukannya pengiriman pesan dalam bentuk *alphanumeric* antara terminal pelanggan atau antar terminal pelanggan dengan sistem eksternal seperti *e-mail*, *paging*, *voice mail*, dan lain-lain. Isu SMS pertama kali muncul di belahan Eropa pada sekitar tahun 1991 bersamaan dengan sebuah teknologi wireless yang saat ini cukup banyak penggunanya, yaitu *Global System for Mobile Communication* (GSM)

Layanan SMS dikembangkan dan distandardisasi oleh suatu badan yang bernama ETSI (*European Telecommunication Standards Institute*). Teknologi ini sebenarnya dikembangkan sebagai bagian dari pengembangan GSM *Phase 2*, yang terdapat pada dokumentasi GSM 03.40 dan GSM 03.38. Fitur SMS ini memungkinkan perangkat Stasiun Seluler Digital (*Digital Cellular Terminal*, seperti ponsel) dapat mengirim dan menerima pesan-pesan melalui jaringan GSM

melalui SMS *Centre* (SMSC) yang akan mengontrol lalu lintas pesan singkat tersebut [HEN05].

Sebuah pesan SMS memiliki kapasitas maksimal 140 byte, untuk karakter 8-bit sebuah pesan bisa memuat 140 karakter, untuk karakter 7-bit sebanyak 160 karakter, dan untuk karakter 16-bit seperti bahasa Jepang, Bahasa Mandarin dan Bahasa Korea yang memakai Hanzi (Aksara Kanji / Hanja) dapat dimuat 70 karakter. Selain kapasitas 140 byte terdapat pula data-data lain. Untuk pesan yang lebih dari 140 byte, seorang pengguna akan dihitung mengirim pesan lebih dari satu. Hal ini menyebabkan seseorang yang ingin mengirimkan pesan yang cukup panjang, terdiri dari sejumlah karakter, atau memiliki karakter yang berukuran besar, akan mengalami kesulitan. Walaupun tersampaikan, pesan harus dirangkai menjadi sejumlah SMS sesuai kapasitas maksimalnya.

Lebih dari sekadar pengiriman pesan biasa, layanan SMS memberikan garansi SMS akan sampai pada tujuan meskipun perangkat yang dituju sedang tidak aktif yang dapat disebabkan karena sedang dalam kondisi mati atau berada di luar jangkauan layanan GSM. Jaringan SMS akan menyimpan sementara pesan yang belum terkirim, dan akan segera mengirimkan ke perangkat yang dituju setelah adanya tanda kehadiran dari perangkat di jaringan tersebut.

Dalam proses pengiriman atau penerimaan pesan pendek (SMS), data yang dikirim maupun diterima oleh stasiun bergerak menggunakan salah satu dari dua mode yang ada, yaitu mode teks atau mode PDU (*Protocol Data Unit*). Dalam mode PDU, pesan yang dikirim berupa informasi dalam bentuk data dengan

beberapa kepala informasi [SUL06]. Hal ini akan memberikan kemudahan jika dalam pengiriman akan dilakukan kompresi data, atau akan dibentuk sistem penyandian data dari karakter dalam bentuk untaian bit-bit biner. Senarai PDU tidak hanya berisi pesan teks saja, tetapi terdapat beberapa *meta*-informasi lainnya, seperti nomor pengirim, nomor SMS *Center*, waktu pengiriman, dan sebagainya. Semua informasi yang terdapat dalam PDU, dituliskan dalam bentuk pasangan-pasangan bilangan heksadesimal yang disebut dengan pasangan *octet*.

Dengan fakta bahwa layanan SMS (melalui jaringan GSM) mendukung jangkauan atau jelajah nasional dan internasional dengan waktu keterlambatan yang sangat kecil, memungkinkan layanan SMS cocok untuk dikembangkan ke dalam berbagai aplikasi. Misalnya saja aplikasi *pager*, *e-mail*, dan notifikasi *voice mail*, serta layanan pesan banyak pemakai (*multiple users*). Namun, pengembangan aplikasi tersebut masih bergantung pada tingkat layanan yang disediakan oleh operator jaringan.

Berbagai keunggulan yang mendukung hal tersebut, yaitu :

- Memiliki harga yang relatif murah, bahkan sebagian perusahaan secara gratis memberikan layanan ini untuk menarik lebih banyak pengguna.
- *Deliver Oriented Service*, pesan akan selalu diusahakan untuk dikirimkan ke tujuan. Jika suatu tujuan sedang tidak aktif, pesan akan disimpan di SMSC (*short message service centre*) sebagai server dan akan dikirim sesegera setelah nomor tujuan aktif. Pesan juga akan tetap terkirim walaupun pengguna

sedang melakukan telepon karena transmisi SMS menggunakan kanal *signaling* bukan kanal suara.

- Dapat dikirim ke berbagai tujuan / penerima pada saat bersamaan.
- Dapat dikirim ke berbagai jenis penerima, seperti *e-mail*, IP, maupun berbagai aplikasi lain.

Memiliki beberapa macam kegunaan bila dilakukan integrasi dengan berbagai aplikasi seperti *chatting*, *voting*, kuis, reservasi, informasi tertentu, dan sebagainya. Fakta mengenai teknologi SMS diperlihatkan pada tabel 2.1.

Tabel 2.1. SMS technology fact sheets

Standard :	GSM (ETS 03.40)
Transport technology:	GSM Signaling Path, GPRS
Transport protocol:	Short Message Protocol
Addressing scheme:	MSISDN
Message description language:	SMS PDU
User data length:	140 byte
Basic character set:	7-bit SMS, 8-bit SMS, UCS2

2.2.1.2. Perkembangan Teknologi Messaging

SMS mulai dikenal pada era teknologi wireless generasi kedua (2G), saat komunikasi data pada telekomunikasi wireless dapat dilakukan. Di Eropa, SMS diperkenalkan pada tahun 1991 saat GSM mulai digunakan. Perkembangan teknologi *messaging* akan ditunjukkan pada Tabel 2.2.

Tabel 2.2. Perkembangan Teknologi Messaging

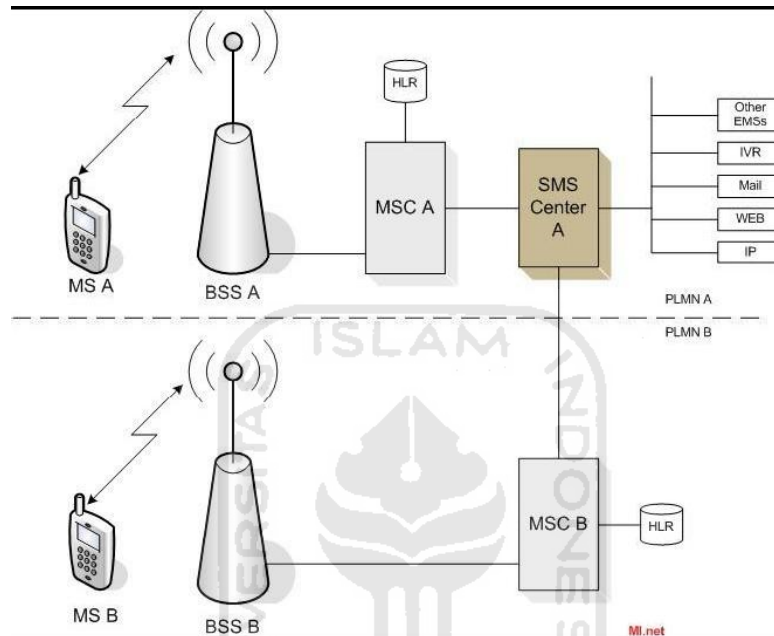
Generasi	Teknologi	Service	Keterangan
1 G	AMPS	✓ <i>Circuit-switched wireless</i>	✓ Tidak bisa

		<i>analog voice.</i> ✓ <i>Limited system capacity and capability.</i> ✓ <i>No data</i>	SMS
2 G	TDMA CDMA GSM	✓ <i>Circuit-switched wireless digital voice data</i> ✓ <i>Security lebih baik</i> ✓ <i>Kapasitas lebih besar</i> ✓ <i>Support komunikasi data</i>	✓ Bisa SMS
2.5 G	GPRS CDMA200-1x EDGE	✓ <i>circuit-switched wireless digital voice</i> ✓ <i>diperkenalkannya packet-switched data services</i> ✓ <i>Kecepatan dan kapasitas lebih baik</i>	✓ Migrasi ke 3G ✓ Bisa SMS, EMS, MMS
3 G	WCDMA CDMA200-MX UMTS	✓ <i>packet-switched wireless</i> ✓ <i>voice dan data</i> ✓ <i>encryptsi, high-speed multimedia</i>	✓ Bisa SMS, EMS, MMS ✓ 3G platform

SMS berkembang menjadi EMS (*Enhanced Message Service*), dengan jumlah karakter yang bisa dikirimkan menjadi lebih banyak serta dapat digunakan untuk mengirim pesan berupa nonkarakter (gambar sederhana) dan rekaman suara. Saat teknologi *packet switch* seperti GPRS mulai digunakan, servis pengiriman pesan berkembang menjadi tidak hanya sebatas teks melainkan mulai digunakan dalam bentuk gambar, suara maupun video (multimedia), layanan ini dikenal dengan MMS (*multimedia message service*) [WIC02].

2.2.1.3. Jaringan SMS

Salah satu contoh arsitektur jaringan GSM dengan *SMS center* (SMSC) di dalamnya akan ditunjukkan pada Gambar 2.1.



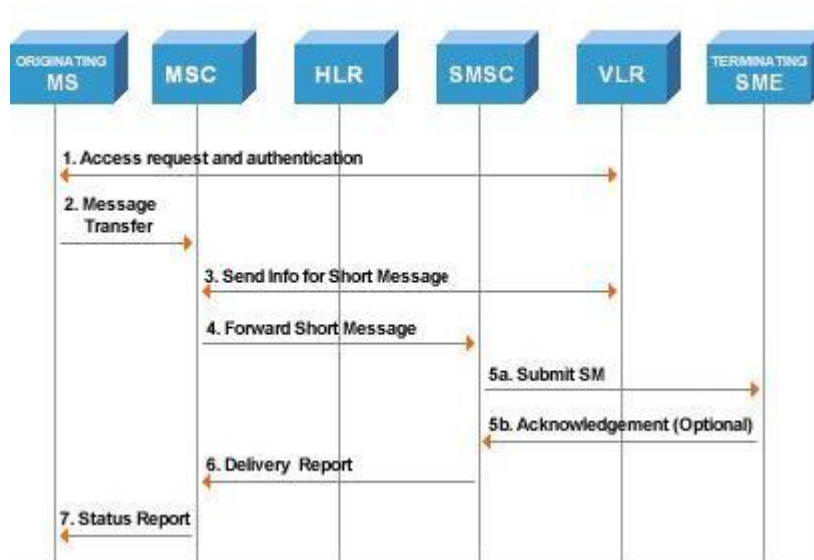
(*ESME = External Short Message Entities)

Gambar 2.1. Jaringan GSM dengan SMSC

Sumber : [HEN05]

SMS, dapat mentransmisikan pesan singkat dari dan ke *Mobile Subscriber* (MS). Pengiriman pesan singkat (SMS) ini dimungkinkan dengan adanya sebuah SMSC. Secara umum SMSC berfungsi menerima SMS yang dikirim, menyimpannya untuk sementara, dan mem-forward (mengirimkan) SMS tersebut ke *mobile subscriber* (MS) ataupun ESME tujuan.

Gambar 2.2 akan menunjukkan alur skenario pengiriman SMS MO (*Mobile Originating*) dari MS ke ESME (*SMS Originating*).

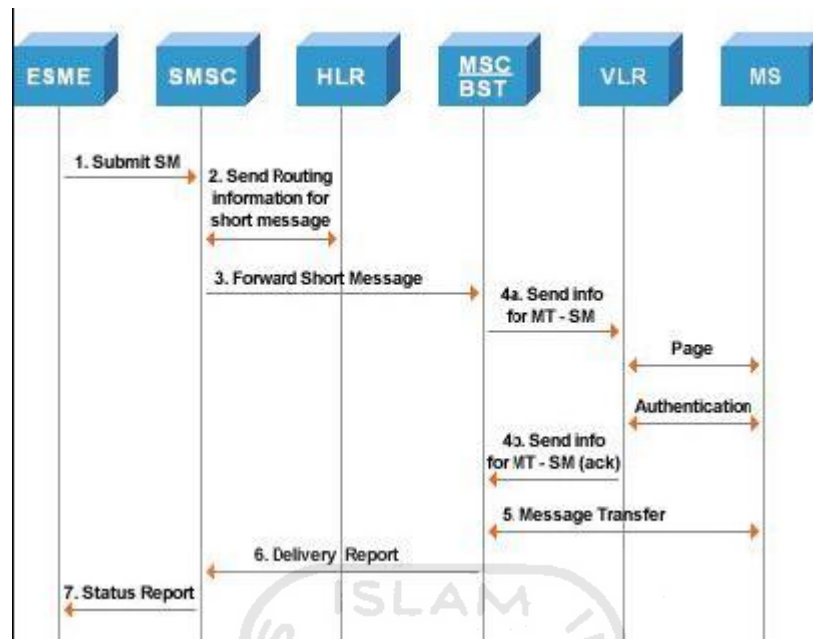


Gambar 2.2. Skenario SMS MO (Mobile Originating)
Sumber : [HEN05]

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 MS tidak sedang dalam keadaan diblok untuk mengirimkan SMS.
4. MSC mengirimkan SMS ke SMSC dengan menggunakan operasi *forward ShortMessage*.
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.

Alur skenario SMS yang diterima MS dari ESME (SMS Terminating) ditunjukkan pada Gambar 2.3.



Gambar 2.3. Skenario MS dari ESME (SMS Terminating)

Sumber : [HEN05]

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.

Transmisi pesan singkat antara SMSC dan *handset* menggunakan *Mobile Application Part* (MAP) dari *protocol SS7*. Pesan dikirimkan dengan MAP *mo* dan *mt-Forward operations*. Jaringan inilah yang memiliki batas *signaling protocol* 140 byte [WIC02].

2.2.2. Konsep Kompresi Data

Kompresi ialah proses pengubahan sekumpulan data / informasi menjadi suatu bentuk kode untuk menghemat kebutuhan tempat penyimpanan dan waktu untuk transmisi data [SUL06]. Proses mengkodekan informasi menggunakan bit atau *information-bearing* unit yang lain yang lebih rendah daripada representasi data yang tidak terkodekan dengan suatu sistem *encoding* tertentu. Pengiriman data hasil kompresi dapat dilakukan jika pihak pengirim atau yang melakukan kompresi dan pihak penerima memiliki aturan yang sama dalam hal kompresi data. Saat ini terdapat berbagai tipe algoritma kompresi antara lain: Huffman, LIFO, LZHUF, LZ77 dan variannya (LZ78, LZW, GZIP), Dynamic Markov Compression (DMC), Block-Sorting Lossless, Run-Length, Shannon-Fano, Arithmetic, PPM (Prediction by Partial Matching), Burrows-Wheeler Block Sorting, dan Half Byte.

Berdasarkan tipe peta kode yang digunakan untuk mengubah pesan awal (isi file input) menjadi sekumpulan *codeword*, metode kompresi terbagi menjadi dua kelompok, yaitu :

- (a) Metode statik : menggunakan peta kode yang selalu sama. Metode ini membutuhkan dua fase (*two-pass*): fase pertama untuk menghitung

probabilitas kemunculan tiap symbol / karakter dan menentukan peta kodenya, dan fase kedua untuk mengubah pesan menjadi kumpulan kode yang akan ditransmisikan. Contoh: algoritma Huffman statik.

- (b) Metode dinamik (adaptif) : menggunakan peta kode yang dapat berubah dari waktu ke waktu. Metode ini disebut adaptif karena peta kode mampu beradaptasi terhadap perubahan karakteristik isi file selama proses kompresi berlangsung. Metode ini bersifat *onepass*, karena hanya diperlukan satu kali pembacaan terhadap isi file. Contoh: algoritma LZW dan DMC.

Berdasarkan teknik pengkodean/pengubahan simbol yang digunakan, metode kompresi dapat dibagi ke dalam tiga kategori, yaitu :

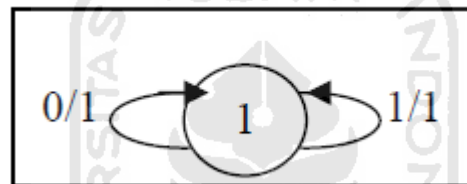
- (a) Metode *symbolwise* : menghitung peluang kemunculan dari tiap symbol dalam file input, lalu mengkodekan satu simbol dalam satu waktu, dimana simbol yang lebih sering muncul diberi kode lebih pendek dibandingkan simbol yang lebih jarang muncul, contoh: algoritma Huffman.
- (b) Metode *dictionary* : menggantikan karakter / *fragmen* dalam file input dengan indeks lokasi dari karakter / *fragmen* tersebut dalam sebuah kamus (*dictionary*), contoh: algoritma LZW.
- (c) Metode *predictive* : menggunakan model *finite-context* atau *finite-state* untuk memprediksi distribusi probabilitas dari simbol-simbol selanjutnya; contoh: algoritma DMC.

Ada 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. Namun, pada kenyataannya tidak ada metode kompresi yang paling efektif untuk semua jenis file. Dalam penelitian ini, diimplementasikan sebuah metode kompresi untuk membuat aplikasi kompresi teks SMS yaitu algoritma DMC. Metode ini diujikan untuk mengkompresi dan mendekompresi pesan teks SMS. Lalu dilakukan analisis statistik untuk membandingkan kinerja metode berdasarkan dua faktor, yaitu rasio perbandingan ukuran file hasil kompresi terhadap file asli dan kecepatan kompresinya.

2.2.2.1. Algoritma Dynamic Markov Compression (DMC)

Dynamic Markov Compression adalah algoritma kompresi data yang dikembangkan oleh Gordon Cormack dan Nigel Horspool. Algoritma DMC merupakan teknik pemodelan yang didasarkan pada model *finite-state* (model Markov). DMC memiliki algoritma kompresi yang baik, namun memerlukan memori yang lebih dan implementasinya belum dapat diperluas secara maksimal sampai saat ini [HOR86]. Beberapa implementasi terbaru yakni program kompresi *Hook* oleh Nania Francesco Antonio, program *Ocamyd* oleh Frank Schwellinger, dan model sub program *PAQ8L* oleh Matt Mahoney. Semua program tersebut didasari implementasi tahun 1993 dalam bahasa C oleh Gordon Cormack.

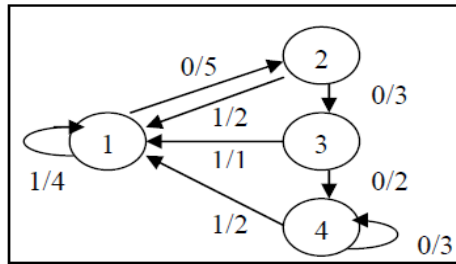
Algoritma DMC ini merupakan teknik pemodelan yang didasarkan pada model *finite-state* (model Markov). DMC membuat probabilitas dari karakter *biner* berikutnya dengan menggunakan pemodelan *finite-state*, dengan model awal berupa mesin FSA (*Finite State Automata*) dengan transisi 0/1 dan 1/1. *DMC* merupakan teknik kompresi yang adaptif, karena struktur mesin *finite-state* berubah seiring dengan pemrosesan file. Kemampuan kompresinya tergolong amat baik, meskipun waktu komputasi yang dibutuhkan lebih besar dibandingkan metode lain.



Gambar 2.4. Model awal DMC

Sumber : [HOR86]

Pada DMC, simbol alfabet input diproses per *bit*, bukan per *byte*. Setiap output transisi menandakan berapa banyak simbol tersebut muncul. Penghitungan tersebut dipakai untuk memperkirakan probabilitas dari transisi. Contoh: pada Gambar 2.5, transisi yang keluar dari *state* 1 diberi label 0/5, artinya bit 0 di *state* 1 terjadi sebanyak 5 kali. *Decoder* membangun model yang identik menggunakan simbol yang didekodekan dan mencatat di *state* mana *encoder* sedang berada.



Gambar 2.5. Sebuah model yang diciptakan oleh metode DMC

Sumber : [COM86]

Secara umum, transisi ditandai dengan $0/p$ atau $1/q$ dimana p dan q menunjukkan jumlah transisi dari *state* dengan *input* 0 atau 1. Nilai probabilitas bahwa *input* selanjutnya bernilai 0 adalah $p/(p+q)$ dan *input* selanjutnya bernilai 1 adalah $q/(p+q)$. Lalu bila *bit* sesudahnya ternyata bernilai 0, jumlah bit 0 di transisi sekarang ditambah satu menjadi $p+1$. Begitu pula bila *bit* sesudahnya ternyata bernilai 1, jumlah bit 1 di transisi sekarang ditambah satu menjadi $q+1$. Contoh : jika suatu *encoder* yang menggunakan model dari gambar 2.5 berada di *state* 2 dan *input* selanjutnya adalah *bit* nol, maka *input* dikodekan dengan probabilitas $3/(2+3) = 0.6$. Jumlah *bit* 0 di transisi sekarang ditambah satu menjadi 4, dan transisi berlanjut ke *state* 3. *Input* selanjutnya dikodekan dari *state* baru tersebut.

Algoritma ini merupakan merupakan salah satu algoritma kompresi yang bersifat *loseless* yang berarti jika ada code number yang hilang maka *message* pun akan hilang. Selain itu DMC bersifat statik karena menggunakan tabel probabilitas yang tetap. DMC bekerja dengan cara mengubah sebuah *String Message* yang berisi kumpulan karakter menjadi sebuah *code number* yang

bernilai antara 0 dan 1. Semua karakter yang mungkin muncul dalam sebuah *String Message* ditentukan probabilitas kemunculannya.

2.2.2.2. *Encoding*

Pada dasarnya, dalam teknik *predictive DMC* dilakukan proses kompresi atau *encoding* dengan menyatakan suatu state dengan jumlah berhingga dan dapat berpindah dari satu state ke state yang lain. Masalah tidak terdapatnya kemunculan suatu bit pada state dapat diatasi dengan menginisialisasi model awal state dengan satu. Probabilitas dihitung menggunakan frekuensi relatif dari dua transisi yang keluar dari state yang baru. Pada tabel probabilitas karakter yang mungkin muncul dalam sebuah *message*, semua karakter memiliki distribusi probabilitas yang berbeda yaitu *Low Range – High Range*. Range ini ditentukan berdasarkan probabilitas karakter tersebut.

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

Algoritma kompresi DMC diberikan 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 disini. Namun jika masih terdapat karakter dalam *message* maka proses diulangi dari langkah dua sampai karakter terakhir. Dengan demikian setelah karakter terakhir maka akan diperoleh sebuah *range high* dan *low* yang baru.

Misalnya telah didefinisikan probabilitas karakter yang mungkin muncul dalam sebuah pesan adalah sebagai berikut :

Tabel 2.3. Contoh Tabel Probabilitas

Karakter	Probabilitas	Range (Low - High)
a	0.2	0.0 - 0.2
b	0.2	0.2 - 0.4
c	0.2	0.4 - 0.6
d	0.2	0.6 - 0.8
e	0.2	0.8 - 1.0

Sebagai contoh agar dapat memahami algoritma ini, misalnya kita akan mengkodekan sebuah *string message* "abe". Maka proses encoding akan berlangsung seperti dibawah ini :

Encode karakter "a"

$$\text{Current range} = \text{high} - \text{low} = 1 - 0 = 1$$

$$\text{High} = 0 + (1 \times 0,2) = 0,2$$

$$\text{Low} = 0 + (1 \times 0) = 0$$

Encode karakter "b"

$$\text{Current range} = \text{high} - \text{low} = 0,2 - 0 = 0,2$$

$$\text{High} = 0 + (0,2 \times 0,4) = 0,08$$

$$\text{Low} = 0 + (0,2 \times 0,2) = 0,04$$

Encode karakter “e”

$$\text{Current range} = \text{high} - \text{low} = 0,08 - 0,04 = 0,04$$

$$\text{High} = 0,04 + (0,04 \times 1) = 0,08$$

$$\text{Low} = 0,04 + (0,04 \times 0,8) = 0,072$$

Dengan demikian range *high* – *low* yang diperoleh adalah 0,072 – 0,08.

Hal ini berarti *code number* untuk *string message* “abe” berada dalam range 0,072 – 0,08.

2.2.2.3. Decoding

Proses ini merupakan proses untuk mengubah sebuah *code number* untuk mendapatkan *message*.

Algoritma *decoding* DMC diberikan 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 manakah *code number* itu berada.
2. Jika sudah mengetahui karakter pertama yang muncul maka selanjutnya mencari *code range* nya dengan menggunakan rumus :

$$\text{code_range} = \text{high_range simbol} - \text{low_range simbol}$$

3. Dapatkan nilai *encoded number* yang baru dengan menggunakan rumus :

$$\text{Encoded number} = (\text{encoded number} - \text{low range simbol}) / \text{code range}$$

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.

Misalkan dalam contoh encoding diatas *range code number* yang diperoleh adalah 0,072 – 0,08, maka *code number* yang dapat dipakai untuk meng *decode* pesan “abe” haruslah angka yang berada pada *range* tersebut. Dipilih angka yang tepat diu tengah *range* yaitu angka 0,076. selanjutnya prosesnya adalah melihat pada tabel probabilitas pada *range* karakter manakah 0.076 berada. Berdasarkan tabel maka karakter pertama yang diperoleh adalah karakter ‘a’. selanjutnya akan dihitung *code range* dan *encoded number* dengan rumus di atas.

$$\begin{aligned} \text{Code range} &= \text{high range karakter 'a'} - \text{low range karakter 'a'} \\ &= 0,2 - 0,0 = 0,2 \end{aligned}$$

$$\text{Encoded number} = (0,076 - 0,0) / 0,2 = 0,38$$

Berdasarkan tabel, 0.38 berada pada range probabilitas karakter ‘b’. selanjutnya perhitungan *code number* dan *encoded number* yang baru adalah :

$$\begin{aligned} \text{Code range} &= \text{high range karakter 'b'} - \text{low range karakter 'b'} \\ &= 0,4 - 0,2 = 0,2 \end{aligned}$$

$$\text{Encoded number} = (0,38 - 0,2) / 0,2 = 0,9$$

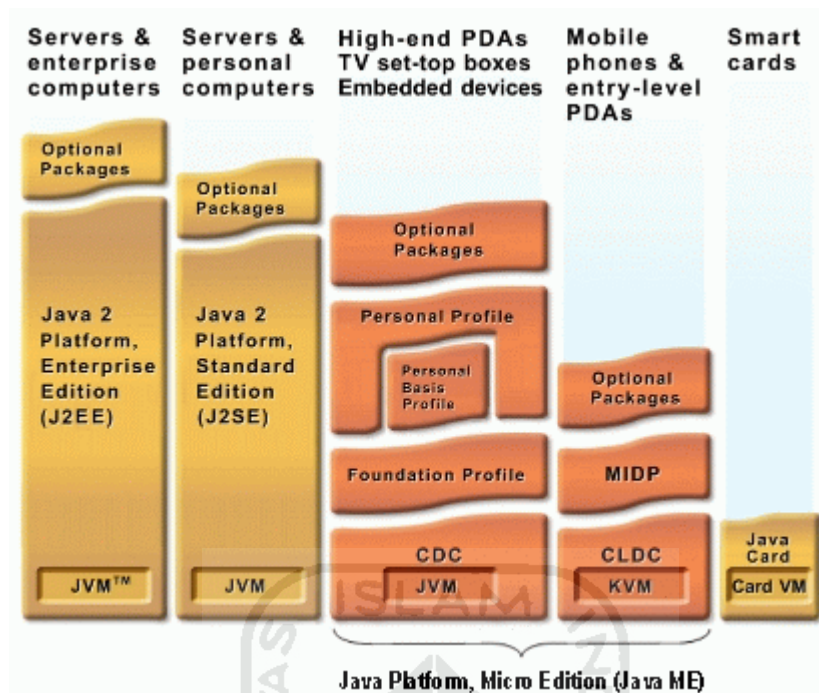
Berdasarkan tabel, 0.9 berada pada range probabilitas karakter ‘e’. dengan demikian pesan yang diperoleh adalah “abe”.

2.2.3. Perangkat Lunak yang digunakan

Untuk mengimplementasikan penelitian ini dikembangkan suatu aplikasi kompresi data dengan menggunakan metode kompresi atau Algoritma *Dynamic Markov Compression* (DMC). Pengembangan aplikasi ini menggunakan platform Java yaitu *Java 2 Micro Edition* (J2ME) sehingga perlu dijelaskan beberapa prinsip dasar dari platform tersebut.

2.2.3.1. *Wireless Messaging Application* menggunakan Teknologi J2ME

Teknologi Java merupakan sebuah teknologi yang berkembang pada beberapa tahun belakangan ini. Teknologi Java yang pada awalnya dikenal untuk aplikasi pada desktop (J2SE) ataupun pada *application server* (J2EE), kini hadir dengan teknologi terbarunya, J2ME™ *Platform*, untuk pembangunan aplikasi pada *mobile device* seperti *mobile phone* dan PDA. Pada saat ini teknologi Java telah diadopsi oleh berbagai vendor mobile devices seperti ponsel maupun PDA. Selain J2ME™ *Platform* yang termasuk baru, terdapat pula satu buah platform teknologi Java yang termasuk baru pula yaitu *Java Card*. *Java Card* merupakan seperangkat kakas untuk membangun aplikasi pada sebuah *card electronic* seperti *SIM Card* pada ponsel. *SIM Card* yang dipergunakan oleh *Mobile Banking* BCA menjadi salah satu contoh dari teknologi *Java Card*. Struktur pengembangan *Java Application* dapat dilihat pada Gambar 2.6.



Gambar 2.6. Struktur pengembangan Java Aplikasi.

. Sumber : [WIC02]

Pada platform J2ME digunakan Kilo Virtual Machine (KVM) yang dimasukkan kedalam *handheld devices*. J2EE merupakan superset dari J2SE, sebaliknya J2SE merupakan superset bagi J2ME. Artinya API yang ada pada J2ME sebagian mengadopsi yang ada pada API J2SE, selain juga mengimplementasikan API spesifik untuk teknologi J2ME itu (*javax.microedition.**).

2.2.3.2. Configurations dan Profiles

Dalam J2ME dibagi menjadi dua bagian besar yang dikenal sebagai *configuration* dan *profile*. Dua istilah ini sangatlah penting dalam pengembangan aplikasi *wireless* dengan Java sehingga perlu dipahami dengan baik Configuration mendefinisikan lingkungan kerja J2ME *runtime*, minimum Java Libraries dan

kapabilitas yang dipunyai oleh para *developer* J2ME. Artinya antara *mobile device* yang *Java enabled* maka akan ditemui *configuration* yang sama. Jadi *Configuration* hanyalah mengatur hal-hal yang berkaitan dengan “kesamaan” bukan mengatur hal-hal yang “membedakan”, sehingga *Configuration* memastikan portabilitas antar *devices*. Saat ini telah didefinisikan dua buah *configuration* yaitu CDC & CLDC dengan perbandingannya dapat dilihat pada Tabel 2.4.

Tabel 2.4. Tabel perbandingan CLDC dan CDC

<i>CLDC (Connected Limited Device Configuration)</i>	<i>CDC (Connected Device Configuration)</i>
<ul style="list-style-type: none"> • Mengimplementasikan subset dari J2SE. 	<ul style="list-style-type: none"> • Mengimplementasikan seluruh fitur dari J2SE.
<ul style="list-style-type: none"> • JVM yang digunakan adalah KVM. 	<ul style="list-style-type: none"> • JVM yang digunakan adalah CVM.
<ul style="list-style-type: none"> • Digunakan pada perangkat <i>handheld</i> (<i>handphone</i>, PDA, <i>two way pager</i>) dengan memori terbatas (160-512 kb). 	<ul style="list-style-type: none"> • Digunakan pada perangkat <i>handheld</i> (<i>internet TV</i>, <i>Nokia Communicator</i>, <i>car TV</i>) dengan memori minimal 2 Mb.
<ul style="list-style-type: none"> • Prosesor : 16/ 32 bit. 	<ul style="list-style-type: none"> • Prosesor : 32 bit.

Profile merupakan kebalikan dari *Configuration* yaitu mengatur hal-hal yang spesifik untuk sebuah *device* atau tipe market misalnya pengaturan tentang *persistent storage* dan *UI*, . Saat ini telah didefinisikan lima buah *Profile*, salah satunya yaitu MIDP yaitu *profile* yang digunakan pada banyak *mobile devices*.

MIDP menyediakan *library-library* Java untuk implementasi dasar *interface* (*GUI*), implementasi *networking*, *database*, dan *timer*. MIDP dirancang khusus untuk *wireless phone* dan *pager*.

2.2.4. Pengujian Hasil Kompresi

Secara umum, kompresi adalah transformasi informasi untuk menurunkan ukuran penyimpanan file. Proses transformasi disebut dengan model. Model didefinisikan dengan metode kompresi yang sesuai. Karakteristik dari kompresi adalah waktu kompresi, waktu dekompresi, efisiensi kompresi, kesalahan kompresi, reliabilitas kompresi [SUL06].

Pada pengembangan aplikasi ini dibuat sebuah analisis untuk menghitung kehandalan algoritma dan aplikasi kompresi data sebagai ukuran kinerja. Adapun kriteria – kriteria tersebut antara lain :

- Kualitas data hasil enkoding : ukuran lebih kecil, data tidak rusak untuk kompresi lossy.
- Kecepatan, ratio, dan efisiensi proses kompresi dan dekompresi.
- Ketepatan proses dekompresi data: data hasil dekompresi tetap sama dengan data sebelum dikompres (kompresi *loseless*)

Pada umumnya tingkat kesalahan dari sebuah metode kompresi ditunjukkan oleh beberapa faktor yaitu rasio kompresi, faktor kompresi, dan MSE. Selain itu, kecepatan kompresi dari sebuah algoritma ditentukan dari ukuran file input dibagi dengan waktu komputasi yang dibutuhkan, dan ditulis dalam satuan KByte/sec.

Rasio kompresi menyatakan besarnya kapasitas data sesudah dikompres yang dirumuskan sebagai berikut:

$$\text{Rasio kompresi} = \frac{\text{Kapasitas Data Keluaran (byte)}}{\text{Kapasitas Data Masukan (byte)}} \times 100\% \dots\dots\dots(1)$$

Apabila rasio kompresi memiliki nilai 0,6 berarti berarti kapasitas data setelah kompresi menjadi 60% dari kapasitas asli. Semakin kecil nilai rasio kompresi berarti kompresi semakin bagus, karena rasio kompresi yang kecil menunjukkan kapasitas data sesudah kompresi menjadi kecil dan apabila nilai rasio kompresi kompresi lebih besar dari satu ini berarti kapasitas data sesudah kompresi lebih besar dari kapasitas data asli.

Faktor kompresi adalah kebalikan dari kompresi yang dirumuskan sebagai berikut:

$$\text{FaktorKompresi} = \frac{\text{Kapasitas Data Masukan (byte)}}{\text{Kapasitas Data Keluaran (byte)}} \dots\dots\dots(2)$$

Semakin besar nilai faktor kompresi semakin bagus dan apabila nilai faktor kompresi kurang dari satu ini berarti kapasitas data sesudah kompresi lebih besar dari kapasitas asli.

BAB III

METODOLOGI

3.1 Analisis Kebutuhan Perangkat Lunak

3.1.1 Metode Analisis

Tahap analisis digunakan untuk mengetahui dan menterjemahkan semua permasalahan serta kebutuhan perangkat lunak yang akan dibangun. Metode yang digunakan untuk menganalisis kebutuhan perangkat lunak yaitu dengan menggunakan metode analisis berorientasi objek yang dilakukan untuk mendapatkan pemahaman yang lebih baik mengenai kebutuhan dan persyaratan fungsional perangkat lunak yang akan dibangun. Metode ini digunakan karena memiliki beberapa kelebihan. Kelebihan metode analisis berorientasi objek antara lain :

1. Kemudahan untuk mengelola kompleksitas aplikasi yang dibuat.
2. Kemudahan untuk melakukan pengembangan aplikasi.
3. Dapat menghasilkan aplikasi dengan struktur yang kokoh.
4. Dapat dilakukan pemakaian ulang kode.
5. Pemeliharaannya lebih mudah.

3.1.2 Hasil Analisis

Berdasarkan hasil analisis kebutuhan, diperoleh gambaran tentang aplikasi yang akan dibangun. Secara garis besar gambaran aplikasi yang akan dibangun adalah aplikasi layanan SMS pada *handheld device* yang digunakan untuk

melakukan kompresi teks SMS dengan mengimplementasikan metode Dynamic Markov Compression.

3.1.2.1 Masukan Sistem (*Input*)

Input yang diberikan pengguna ke sistem adalah

1. Teks SMS yang akan dikirimkan berupa karakter ASCII.
2. Nomor Telepon Penerima pesan.

3.1.2.2 Keluaran Sistem (*Output*)

Output yang dihasilkan oleh sistem adalah :

1. Teks SMS Kompresi yang telah didekompresi.
2. Daftar SMS masuk yang telah diterima dalam Inbox telepon.
3. Daftar SMS keluar yang telah disimpan pada Draft telepon.
4. Bentuk keluaran daftar SMS meliputi nomor pengirim, isi pesan dan mode pesan.

3.1.2.3 Antarmuka (*Interface*)

Kebutuhan antarmuka (*interface*) yang akan dirancang :

1. Antarmuka Halaman Utama
2. Antarmuka Submenu Pesan Baru
3. Antarmuka Perbandingan mode Normal dan mode Kompresi
4. Antarmuka Halaman Laporan Hasil Kompresi
5. Antarmuka Form Inputan Nomor Telepon
6. Antarmuka Menu Daftar Inbox

7. Antarmuka Menu Daftar Draft
8. Antarmuka Halaman Help
9. Antarmuka Halaman About

3.1.2.4 Fungsi yang diharapkan

Fungsi dan kinerja yang diharapkan dalam sistem ini antara lain :

1. Perangkat lunak yang dibuat diharapkan mampu melakukan proses kompresi teks pesan SMS yang akan dikirimkan serta dapat melakukan proses dekompresi pesan SMS yang masuk / diterima.
2. Dapat melakukan proses pengiriman dan proses penerimaan pesan SMS antar perangkat yang menggunakan aplikasi ini.
3. Dapat melakukan proses penyimpanan pesan SMS yang masuk maupun yang keluar pada database lokal.
4. Dapat melakukan analisis kinerja hasil kompresi.

3.2 Perancangan Perangkat Lunak

3.2.1 Metode perancangan

Setelah melakukan tahap analisis kebutuhan perangkat lunak, maka dilanjutkan dengan perancangan perangkat lunak.

Berdasarkan data yang diperoleh, maka perancangan aplikasi kompresi SMS menggunakan metode perancangan berorientasi objek.

3.2.2 Hasil Perancangan

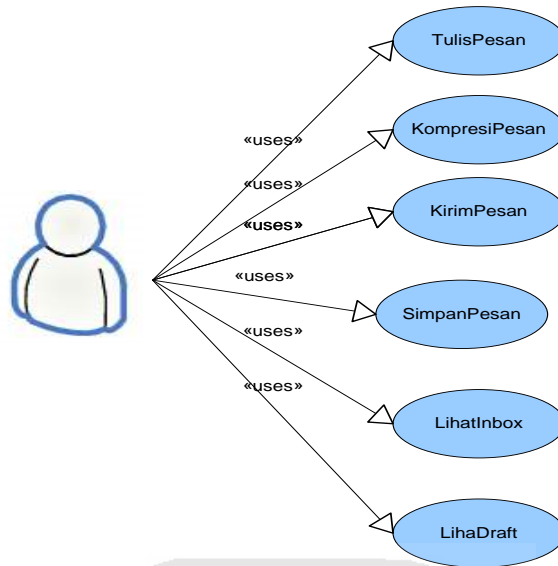
Hasil perancangan sistem ini dibedakan menjadi beberapa bagian sesuai dengan bagian-bagian yang digunakan pada metode perancangan berorientasi objek. Perancangan aplikasi kompresi SMS menggunakan bahasa pemodelan UML (*Unified Modelling Language*). Hal ini dilakukan karena dengan menggunakan bahasa pemodelan UML dapat membantu dalam perancangan dan analisa perangkat lunak yang akan dibangun.

Perancangan aplikasi kompresi SMS terdiri dari beberapa diagram antara lain *use case diagram*, *class diagram*, *sequence diagram*, *activity diagram*.

3.2.2.1 Use Case Diagram

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri dari diagram untuk *use case* dan *actor*. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Actor* memrepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case* [MUN06].

Use case diagram pada perancangan aplikasi kompresi SMS terdiri dari aktor user (pengguna aplikasi kompresi SMS) dan tujuh *use case* yaitu *use case* TulisPesan, *use case* KompresiPesan, *use case* KirimPesan, *use case* SimpanPesan, *use case* LihatInbox dan *use case* LihatDraft. *Use case diagram* pada perancangan aplikasi kompresi SMS ditunjukkan pada Gambar 3.1.



Gambar 3.1 Use case diagram

3.2.2.2 Class Diagram

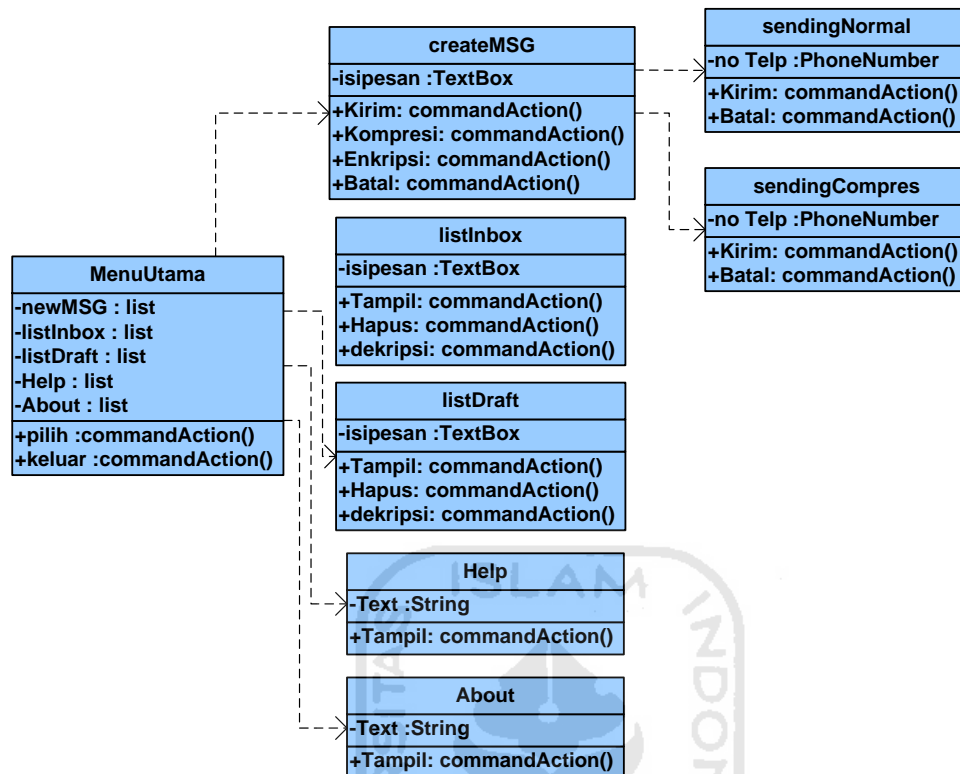
Class diagram menggambarkan *class* dan hubungan antar *class* di dalam sistem. *Class diagram* dibangun berdasarkan *use case diagram* yang telah dibuat sebelumnya.

Class merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara : *associated* (berhubungan satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (kelompok bersama sebagai satu unit).

Class digambarkan dengan sebuah kotak dibagi menjadi tiga bagian. Bagian yang paling atas diisikan nama *class*, bagian tengah diisikan variabel yang dimiliki *class*, dan bagian bawah diisikan method-method dari *class*. Nama *class* digunakan untuk mendeklarasikan *class*. Variabel digunakan untuk

mendesripsikan variabel dari *class*. Variabel memiliki tiga macam *acces level* yaitu: *public* (variabel dapat diakses secara langsung dari objek yang diturunkan), *private* (tidak dapat diakses secara langsung dari objek yang diturunkan, variabel *private* hanya dapat diakses di dalam class itu saja namun tidak dapat diakses oleh *subclass*), dan *protected* (seperti juga variabel *private*, hanya tersedia di dalam class di mana variabel itu dideklarasikan. Perbedaannya variabel *protected* dapat juga diakses di dalam subclass dari *class* di mana variabel *protected* ini dideklarasikan). Method menunjukkan operasi-operasi yang dapat dilakukan oleh *class*. Method terdiri dari dua bagian yaitu bagian deklarasi method (mendefinisikan nama *method* dan attribut lainnya) dan bagian badan method (berisikan semua operasi yang akan dilakukan oleh method). Method juga memiliki tiga tingkatan *acces* yaitu *public*, *private* dan *protected* [MUN06].

Class pada aplikasi kompresi SMS terdiri dari beberapa class sesuai dengan fungsinya masing-masing. Adapun classs yang terdapat pada aplikasi kompresi SMS antara lain *class* MenuUtama, *class* CreateMSG, *class* listInbox, *class* listFolder, *class* listHelp, *class* About, *class* sendingNormal dan *class* sendingCompress. Pada pembuatan aplikasi kompresi SMS, *class diagram* ditunjukkan pada Gambar 3.2



Gambar 3.2 Class diagram

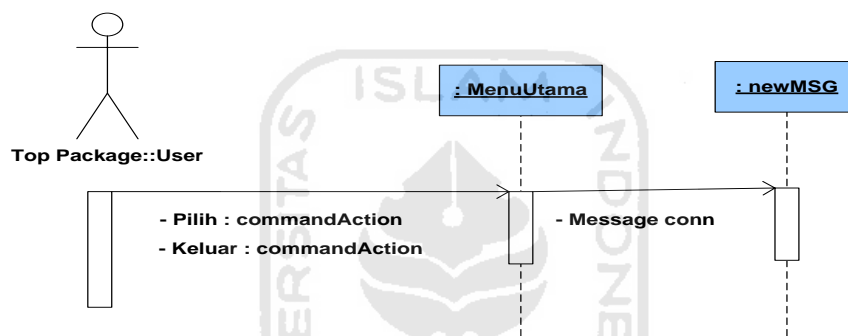
3.2.2.3 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan sekitar sistem (termasuk pengguna, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri antara dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Tujuan penggunaan *sequence diagram* yaitu untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

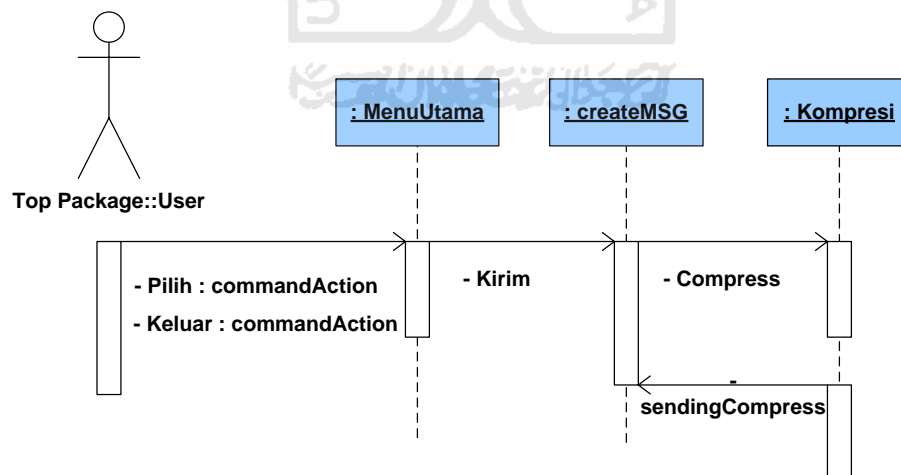
Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai *respons* dari sebuah keadaan

untuk menghasilkan *output* tertentu. Diawali dari proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

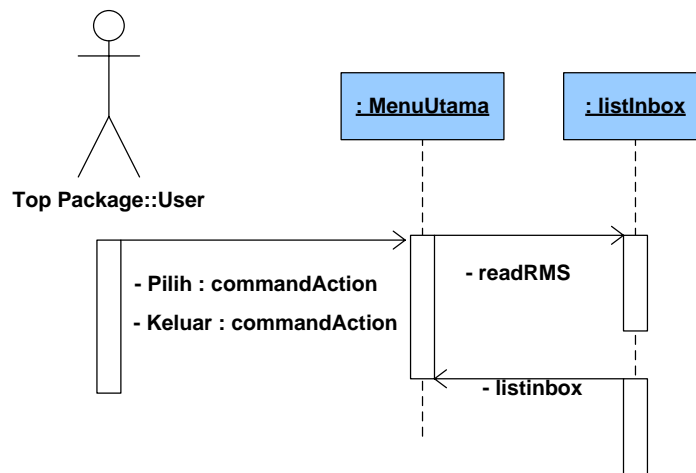
Pada pembuatan aplikasi kompresi SMS, *sequence diagram* pesan baru ditunjukkan pada Gambar 3.3, *sequence diagram* kompresi SMS ditunjukkan pada Gambar 3.4, *sequence diagram* tampil Inbox ditunjukkan pada Gambar 3.5, *sequence diagram* tampil Draft ditunjukkan pada Gambar 3.6



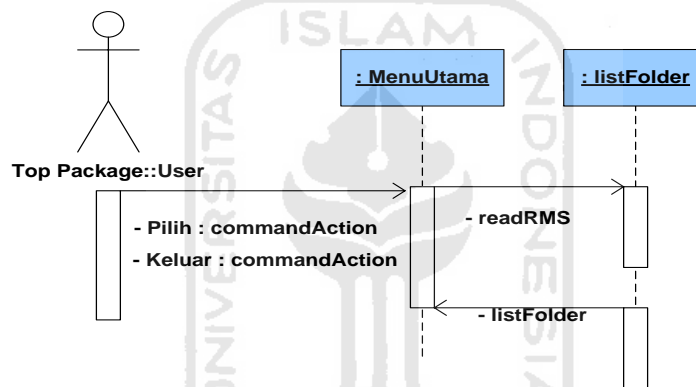
Gambar 3.3 Sequence diagram pesan baru [HAJ09]



Gambar 3.4 Sequence diagram kompresi SMS [HAJ09]



Gambar 3.5 Sequence diagram tampil Inbox [HAJ09]

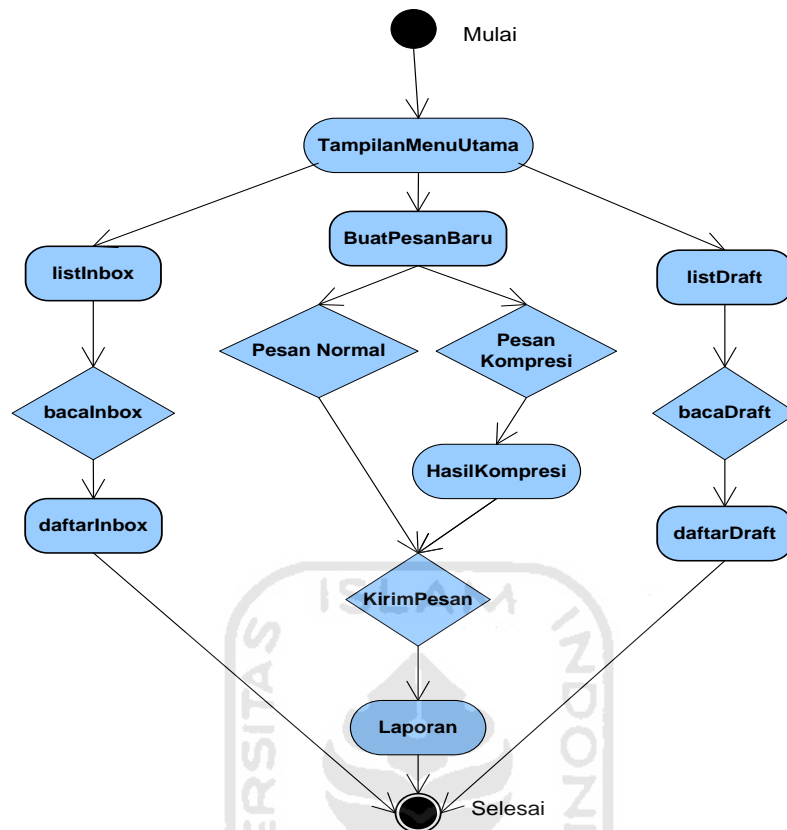


Gambar 3.6 Sequence diagram tampil Draft [HAJ09]

3.2.2.4 Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas actor, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case*.

Pada umumnya *activity diagram* tidak menampilkan secara detail urutan proses, namun hanya memberikan gambaran global bagaimana urutan prosesnya. Sehingga sering kali diagram ini digunakan untuk memodelkan aktivitas bisnis dalam *level* konseptual [HEN06].



Gambar 3.7 Activity diagram

3.2.2.5 Rancangan Antarmuka

Rancangan antarmuka halaman utama aplikasi kompresi SMS akan ditunjukkan pada Gambar 3.8.

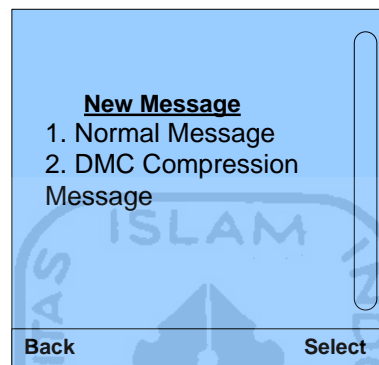


Gambar 3.8. Rancangan Antarmuka Halaman Utama Aplikasi

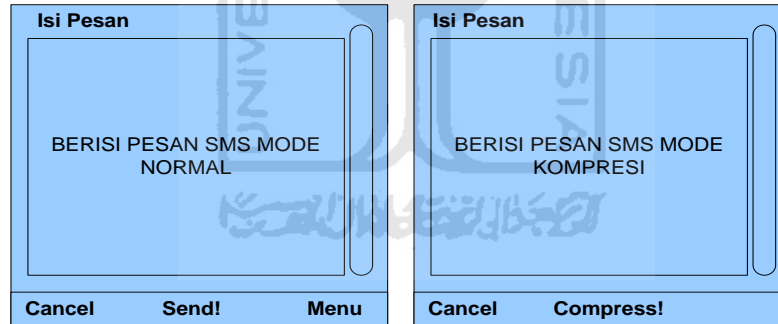
Rancangan antarmuka pada halaman utama terdiri dari:

1. Menu New Message

Pada menu pesan baru, akan dirancang 2 buah opsi bagi pengguna dalam memilih jenis pesan yang akan dikirimkan yaitu : pesan Normal dan pesan Kompresi. Selanjutnya, akan tampil form untuk pengetikan isi pesan SMS sesuai dengan jenis pesannya.



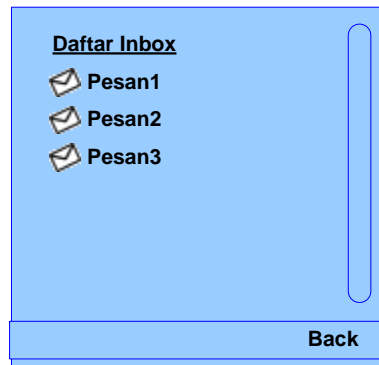
Gambar 3.9. Rancangan Antarmuka Menu Buat pesan baru



Gambar 3.10. Rancangan Antarmuka Input Isi pesan

2. Menu Inbox

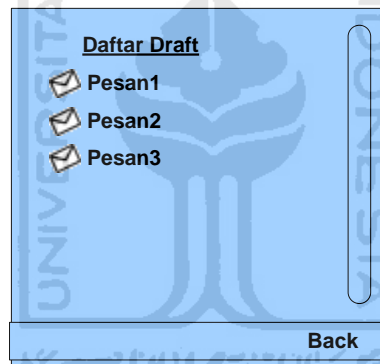
Pada menu Inbox, pemilih dapat melihat daftar pesan yang masuk, membuka pesan yang diinginkan atau menghapusnya.



Gambar 3.11. Rancangan Antarmuka Menu Inbox

3. Menu Draft

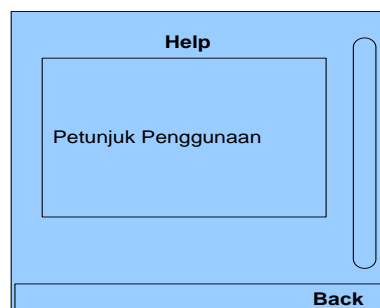
Pada menu Draft, pemilih dapat melihat daftar pesan yang masuk, membuka pesan yang diinginkan atau menghapusnya.



Gambar 3.12. Rancangan Antarmuka Menu Draft

4. Menu Help

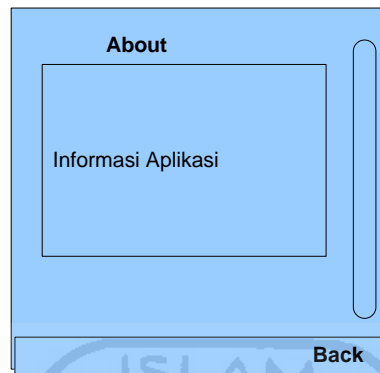
Pada menu Bantuan, pengguna dapat mengetahui informasi cara penggunaan aplikasi.



Gambar 3.13. Rancangan Antarmuka Menu help

5. Menu About

Pada menu Bantuan, pengguna dapat mengetahui informasi umum mengenai aplikasi.



Gambar 3.14. Rancangan Antarmuka Menu About

6. Menu Minimize

Pada menu minimize berfungsi untuk meminimize aplikasi tanpa menutup aplikasi tersebut.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan tahapan yang bertujuan untuk memastikan apakah perangkat lunak yang dibangun dapat bekerja secara efektif dan efisien sesuai yang diharapkan.

Sebelum perangkat lunak digunakan dan diimplementasikan, maka perangkat lunak harus bebas dari kesalahan. Kesalahan yang mungkin terjadi antara lain kesalahan penulisan bahasa, kesalahan tampilan ataupun kesalahan proses. Setelah perangkat lunak bebas dari kesalahan, kemudian dapat dilakukan pengujian dengan menjalankan perangkat lunak.

4.1.1 Batasan Implementasi

Aplikasi yang dibuat adalah sebuah aplikasi berbasis *mobile* yang *Java-enabled*. Aplikasi ini digunakan untuk melakukan kompresi teks SMS dengan mengimplementasikan metode DMC. Dalam proses pembuatan dan pengembangan aplikasi kompresi SMS pada *mobile device* ini banyak ditemui asumsi baru yang lebih beragam dari batasan yang telah direncanakan, batasan-batasan adalah aplikasi yang dikembangkan dengan kakas pengembangan Nokia Developer sehingga fitur yang terdapat pada aplikasi lebih optimal digunakan pada ponsel bermerk Nokia walaupun dapat juga digunakan pada merk lain.

4.1.2 Implementasi

Bagian ini memuat dokumentasi dan penjelasan tentang implementasi perangkat lunak yang meliputi proses-proses yang dijalankan pada aplikasi.

5.1.2.1 Kebutuhan Perangkat Lunak

Perangkat lunak (*software*) yang digunakan untuk pembuatan aplikasi kompresi SMS adalah :

1. Microsoft Windows XP

Microsoft Windows XP merupakan sistem operasi yang digunakan dalam pembuatan aplikasi kompresi SMS.

2. JDK-6-Windows-i586

JDK-6-Windows-i586 merupakan Java SDK versi terbaru untuk sistem operasi Windows. Java SDK adalah satu set perangkat lunak yang digunakan untuk manajemen dan membangun berbagai macam aplikasi Java. Pembuatan aplikasi kompresi SMS menggunakan bahasa pemrograman Java dan JDK yang digunakan adalah JDK-6-windows-i586. Alasan penggunaan JDK-6-windows-i586 yaitu JDK ini merupakan JDK versi terbaru dari Sun Microsystems yang mana JDK ini lebih *powerfull* dibanding versi sebelumnya.

3. JCreator LE 2.50

JCreator LE 2.50 merupakan software open source yang didirikan oleh Sun Microsystems. Pada pembuatan aplikasi kompresi SMS ini menggunakan JCreator LE 2.50 sebagai IDEnya. Alasan penggunaan

JCreator LE 2.50 yaitu dapat memudahkan penulisan program, kompilasi program dan menjalankan program Java.

4. Sun J2ME Wireless Toolkit 2.5

Sun J2ME Wireless Toolkit 2.5 (J2ME WTK) adalah kaskas yang menyediakan lingkungan emulator, dokumentasi beserta contoh-contoh aplikasi Java untuk perangkat kecil (*small device*). J2ME WTK berbasiskan CLDC dan MIDP yang meniru kerja ponsel yang mendukung MIDP atau yang biasa disebut Emulator.

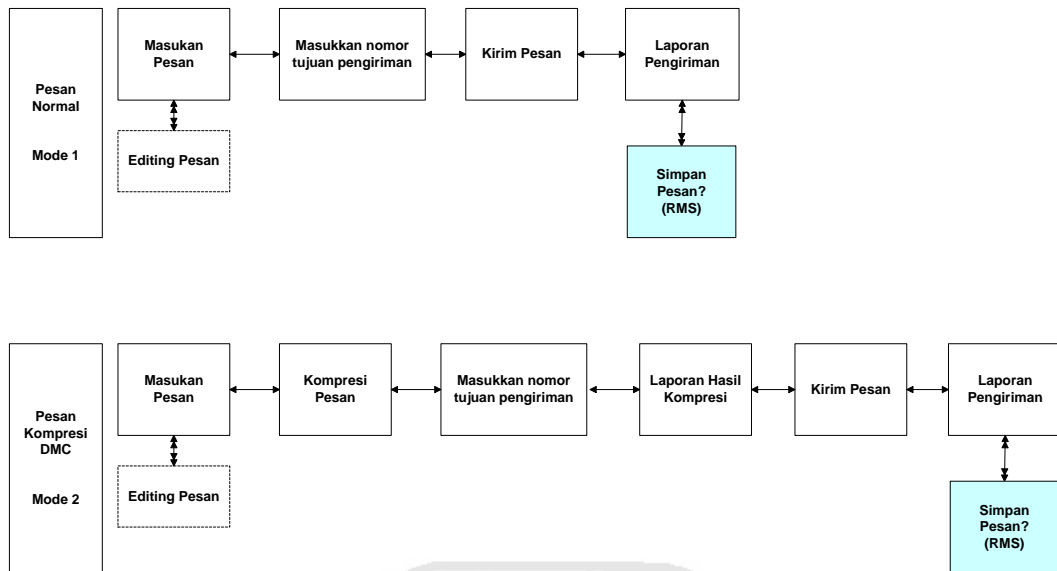
5.1.2.2 Kebutuhan Perangkat Keras

Spesifikasi perangkat keras (*hardware*) yang digunakan untuk pembuatan aplikasi kompresi SMS adalah:

1. Komputer dengan processor Intel P4 2.4 Ghz.
2. RAM 1300MB pc3200.
3. Hardisk 40 GB.
4. Monitor.
5. Mouse.
6. Keyboard.
7. Telepon Seluler.

4.1.2.3 Mekanisme Penggunaan Aplikasi

Pada setiap mode pengiriman, masing-masing memiliki mekanisme yang berbeda pada proses pengiriman pesan. Adapun skema pengiriman pesan dalam berbagai mode pengiriman dapat dilihat pada Gambar 4.1.



Gambar 4.1 Mekanisme Pengiriman dalam berbagai mode pesan.

Mekanisme penggunaan aplikasi kompresi SMS dapat digambarkan sebagai berikut:

1. Pengguna memilih jenis pesan yang tersedia dalam 2 mode pengiriman yaitu mode Normal dan mode Kompresi.
2. Pada mode Normal, pengguna dapat mengirimkan teks SMS biasa ke *mobile device* lain yang berbentuk *plaintext*. Pesan ini akan dikirimkan ke inbox ponsel (*default*) karena SMS-Port penerima tidak diinisialisasikan.
3. Pada mode Kompresi, setelah pengguna mengetikkan isi pesan yang diinginkan maka aplikasi tidak langsung mengirimkan pesan tetapi meminta pengguna untuk melanjutkan pada proses kompresi teks yang telah diketik. Untuk mengatur agar SMS yang dikirim tidak masuk ke inbox ponsel (*default*), maka digunakan port tertentu yang didefinisikan pada bagian SMS-Port yang pada aplikasi ini menggunakan nomor port

5000. Nilai dari SMS-Port ini berkisar dari 1000 – 9999 (tergantung pada spesifikasi tiap ponsel).

4. Pada mode Kompresi, pengguna akan menerima Laporan analisa hasil proses kompresi berupa informasi ratio kompresi dan panjang pesan yang dikirimkan.
5. Pesan yang telah dikirimkan dapat disimpan dalam Arsip dan pesan yang masuk disimpan dalam Inbox. Pada platform J2ME penyimpanan pesan ini dapat dilakukan pada memori ponsel dalam bentuk *record* sehingga teknologi penyimpanan ini disebut RMS (Record Management System).
6. Pada sisi penerima pesan, penerima tidak dapat membaca pesan asli yang dikirim oleh pengirim, apabila penerima tidak mempunyai alat / device yang ter-*install* atau terpasang aplikasi SMSCompress kecuali pada mode Normal. Sedangkan pada mode kompresi, penerima hanya akan menerima pesan yang tidak terbaca. Dengan kata lain, ketika ingin melakukan komunikasi menggunakan aplikasi SMSCompress, baik sisi pengirim maupun penerima harus telah terpasang aplikasi ini.

4.1.2.4 Implementasi Pada Script (Source Code)

Pada *source code* akan menampilkan tabel probabilitas kumulatif yang diperlukan untuk kompresi DMC. Disimpan dalam *array* sepanjang 128. Index dari *array* menunjukkan kode *integer* ASCII dari 92 huruf yang mungkin muncul. 92 huruf yang mungkin muncul itu adalah karakter A-Z, karakter a-z, angka 0-9 dan 30 karakter tambahan.

Dengan menggunakan probabilitas kumulatif ini maka *array* untuk menyimpan *range* probabilitas batas atas dan batas bawah dari tiap karakter tidak perlu. Hal itu dikarenakan terdapatnya korelasi antara batas atas dan batas bawah karakter yang saling berurutan.

Adapun tabel ini menyimpan angka probabilitas dalam *string* yang akan sering diakses dalam proses kompresi maupun dekompresi. Kedua proses sebelumnya akan mengambil isi *array string* karakter dan mengubahnya ke bentuk tipe data *double* untuk diproses dalam perhitungan *encoding*.

```
package source;

import java.io.PrintStream;
public class Compressi
{
    static String cc[] = new String[128];
    public Compress ()
    {
        {
            cc[31] = "0.0";
            cc[32] = "0.1738"; // space
            cc[33] = "0.1778"; // !
            cc[34] = "0.1783"; // "
            cc[36] = "0.1786"; // $
            cc[37] = "0.1788"; // %
            cc[38] = "0.1799"; // &
            cc[39] = "0.1804"; // '
            cc[40] = "0.1812"; // (
            cc[41] = "0.1817"; // )
            cc[42] = "0.1822"; // *
            cc[43] = "0.1827"; // +
            cc[44] = "0.1958"; // ,
            cc[45] = "0.1963"; // -
            cc[46] = "0.2274"; // .
            cc[47] = "0.2286"; // /
            cc[48] = "0.2341"; // 0
            cc[49] = "0.2384"; // 1
            cc[50] = "0.2446"; // 2
            cc[51] = "0.2476"; // 3
            cc[52] = "0.2483"; // 4
            cc[53] = "0.2499"; // 5
            cc[54] = "0.2522"; // 6
            cc[55] = "0.2530"; // 7
            cc[56] = "0.2532"; // 8
            cc[57] = "0.2557"; // 9

            .
            .
            .
            (tidak semua ditampilkan)
        }
    }
}
```

Gambar 4.2 Script Implementasi Tabel Probabilitas

Selanjutnya karakter yang masuk berupa string diubah ke dalam *code number* yang selanjutnya dikonversikan ke *string biner* dan disimpan ke dalam *array byte* dan siap untuk dikirim.

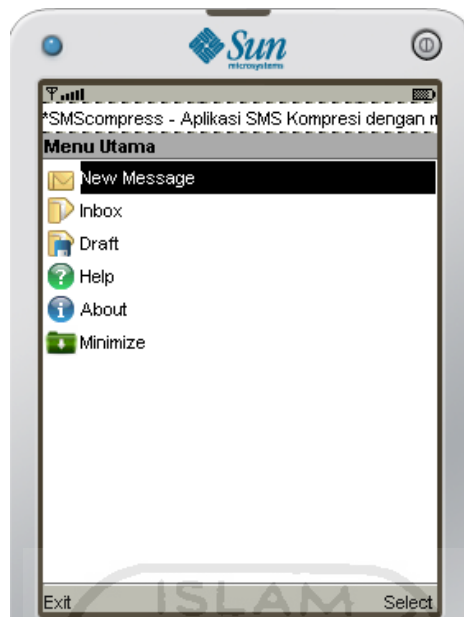
```
static byte[] codeKestring (String cod)
{
    String cod = Char.toBinaryString(cod)
    for(i = 1 ; i >= cod.length ; i++)
    {
        if (s %2) {
            String sebiner = cod.substring(i, i++);
            byte s = (byte)Integer.parseInt(sebyte, 0);
            code[i/2] = cod;
        }
    }
    return code;
}

static byte[] stringKeByteA (String st)
{
    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;
    }
    return hs;
}
```

Gambar 4.3 konversi code number dan string ke byte

4.1.2.5 Implementasi Antarmuka

Antarmuka halaman utama adalah tampilan pertama kali dari aplikasi kompresi SMS setelah pengguna menjalankan aplikasi terlihat pada Gambar 4.4.

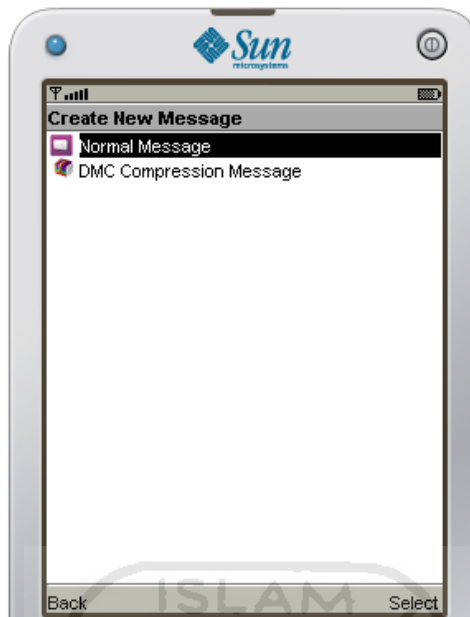


Gambar 4.4 Antarmuka Halaman Utama

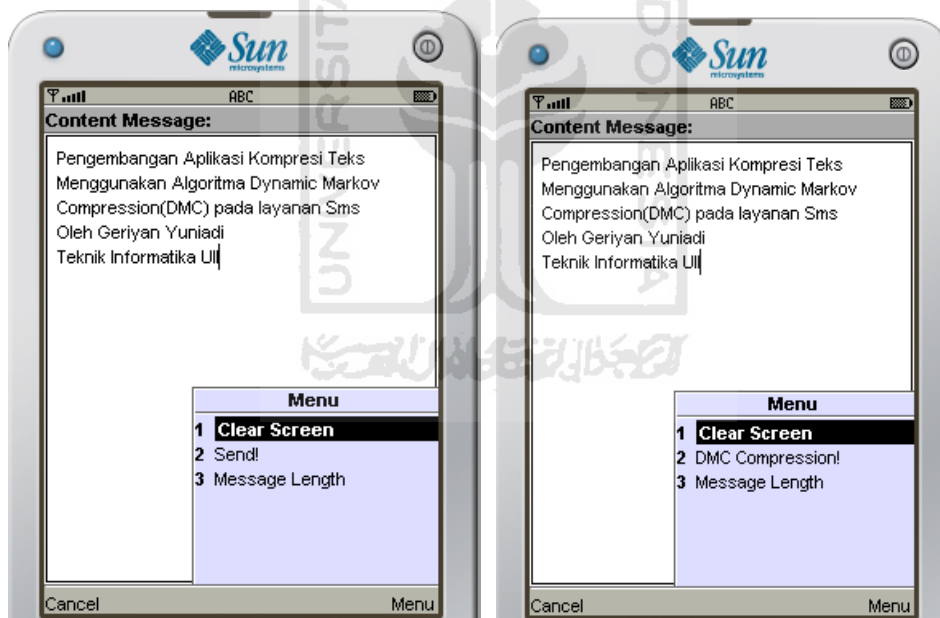
Pada halaman utama aplikasi kompresi SMS terdiri dari 5 submenu, dimana masing-masing komponen memiliki fungsi yang berbeda-beda. Adapun fungsi dari masing-masing menu adalah sebagai berikut :

1. Submenu Pesan Baru

Submenu pesan baru digunakan pengguna untuk melakukan pengetikan teks untuk pengiriman SMS. Ketika pengguna ingin menulis pesan baru dengan memilih menu ini, maka pengguna akan diberikan opsi 2 jenis pesan yang akan dikirimkan yaitu : mode Normal dan mode Kompresi.

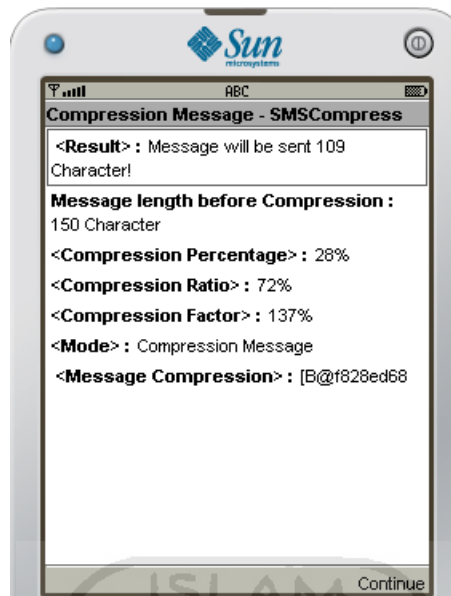


Gambar 4.5 Antarmuka Submenu Pesan Baru



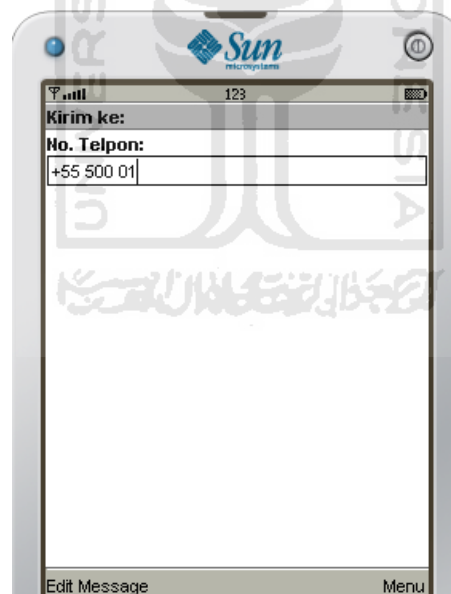
Gambar 4.6 Perbandingan mode Normal dan mode Kompresi

Selanjutnya pada mode Kompresi, aplikasi akan memberikan Laporan analisa hasil kompresi.



Gambar 4.7 Halaman Laporan Hasil Kompresi

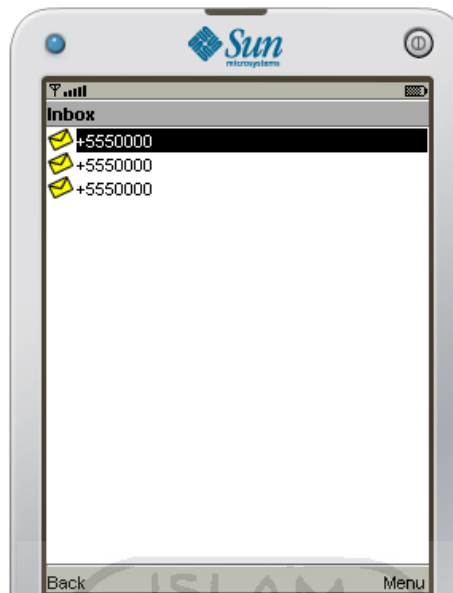
Selanjutnya, pengguna akan diminta menginputkan nomor telepon penerima SMS.



Gambar 4.8 Form Inputan Nomor Telepon

2. Submenu Inbox

Submenu Kotak Masuk akan menampilkan daftar pesan SMS yang masuk baik memakai mode Normal dan Mode Kompresi.



Gambar 4.9 Menu Daftar Inbox

3. Submenu Draft

Submenu Draft akan menampilkan daftar pesan SMS yang keluar baik memakai mode Normal dan Mode Kompresi



Gambar 4.10 Menu Daftar Draft

4. Submenu Help

Menu ini menampilkan petunjuk penggunaan aplikasi.



Gambar 4.11 Halaman Help

5. Submenu About

Menu ini menampilkan informasi Tentang aplikasi.



Gambar 4.12 Halaman About

4.2 Analisis kerja perangkat lunak

4.2.1 Pengujian Perangkat Lunak

Pengujian perangkat lunak diperlukan dalam membangun sebuah perangkat lunak. Hal ini dilakukan untuk mendeteksi kesalahan-kesalahan yang mungkin terdapat dalam perangkat lunak yang dibuat. Kesalahan yang ada pada perangkat lunak selanjutnya diperbaiki agar perangkat lunak yang dibuat dapat bekerja secara maksimal. Pengujian dilakukan pada setiap fungsi yang ada pada aplikasi dengan memasukkan data-data yang tidak *valid* agar dapat diketahui bagaimana sistem dalam menangani data-data yang tidak *valid* atau kesalahan-kesalahan yang mungkin dilakukan oleh pengguna sistem. Materi pengujian meliputi komponen masukan, proses dan keluaran.

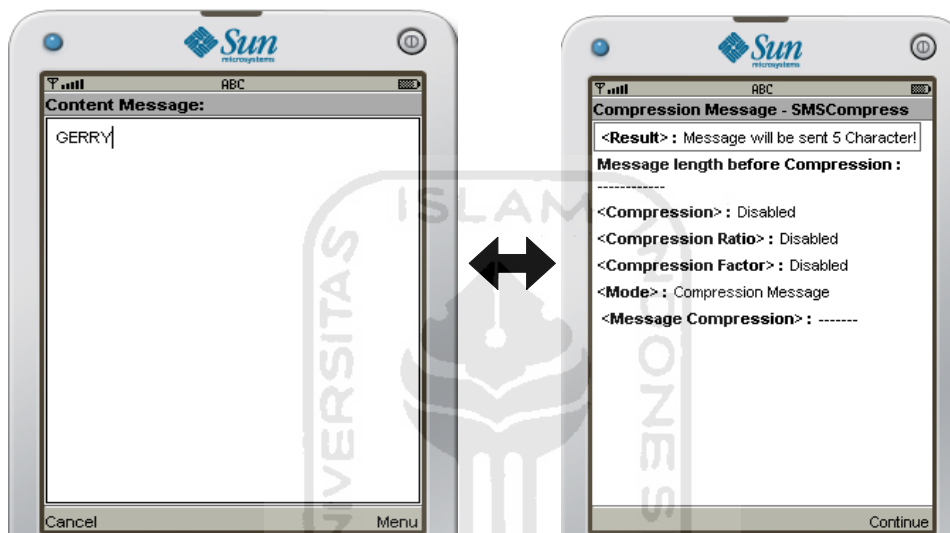
4.2.1.1 Pengujian Peng-kompresan Pesan

Untuk mengetahui kehandalan aplikasi, dilakukan beberapa pengujian dalam beberapa kasus pengiriman pesan pada mode kompresi dan mode enkripsi berikut ini :

Pada mode kompresi, sebelum pesan dikirim maka panjang pesan akan dianalisa terlebih dahulu untuk mengetahui efektifitas dan efisiensi kompresi. Berikut ini adalah beberapa contoh atau beberapa asumsi pengiriman berbagai pesan yang berbeda menurut panjang pesan/karakter yang akan dikompresi, untuk mengukur kehandalan aplikasi :

- a. Apabila hasil analisa aplikasi terhadap pesan yang dikirimkan, setelah dikompresi panjang pesan asli ternyata sama atau lebih banyak daripada pesan hasil kompresi maka pesan dikirim tanpa melakukan

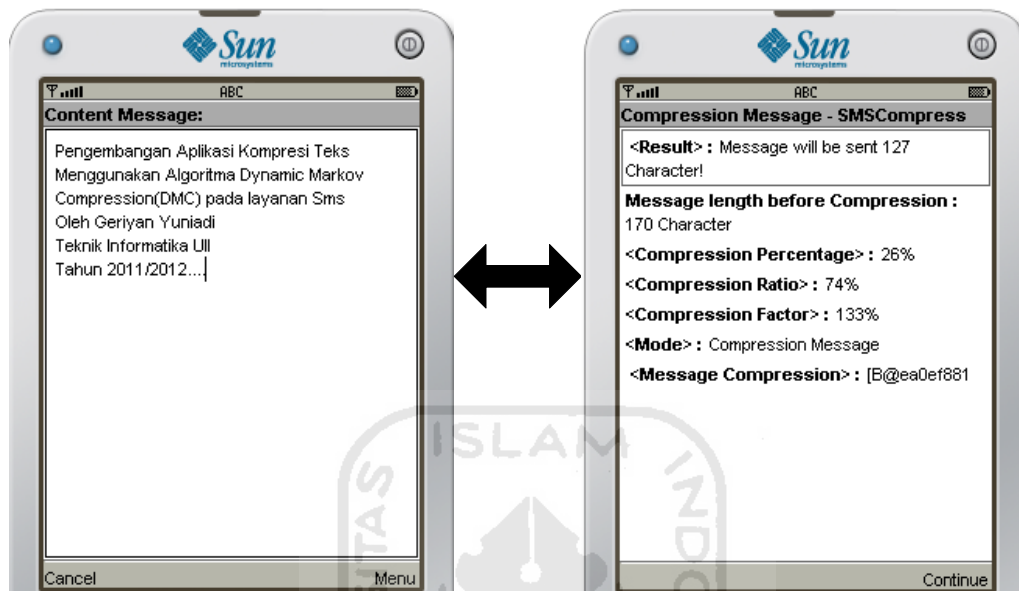
kompresi. Sehingga dengan cara ini, aplikasi dapat dicegah agar tidak melakukan kompresi SMS yang tidak perlu. Hal ini dicontohkan dengan mengirim sebuah pesan yang aslinya berjumlah 5 karakter yang terdapat pada Gambar 4.13. Pada contoh, aplikasi tetap mengirim pesan sebanyak 5 karakter tanpa melakukan kompresi.



Gambar 4.13 Pengujian pertama kompresi

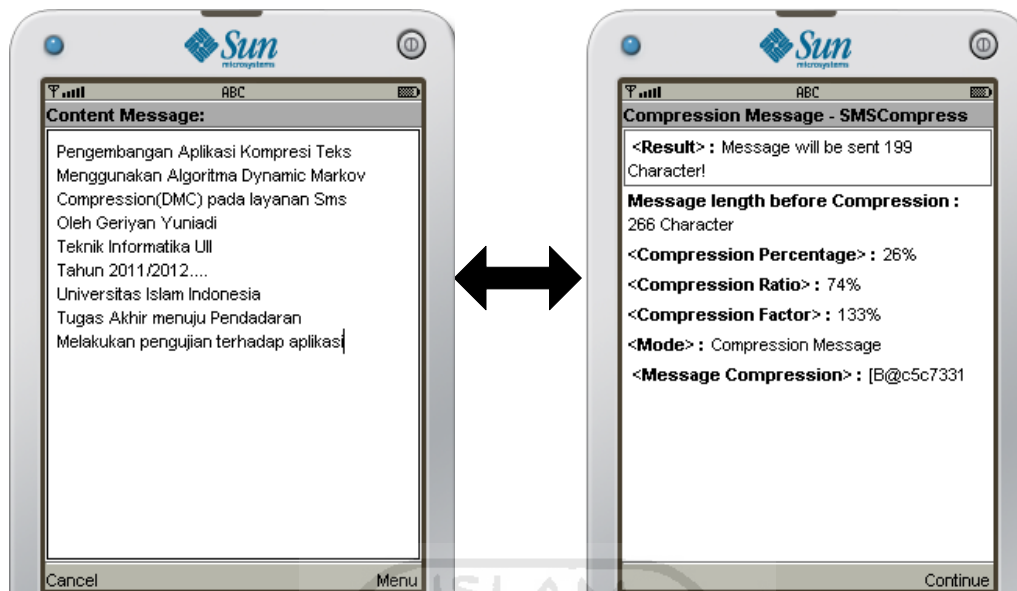
- b. Apabila pesan yang dikirim mempunyai panjang karakter yang melebihi panjang maksimal karakter dalam 1 paket SMS (biasanya sebanyak 160 karakter), pada mode Normal akan dikirimkan sebanyak 2 paket SMS. Namun pada mode kompresi ini dapat menghemat pesan sampai dengan 40%, sehingga pesan SMS dapat dihemat dengan hanya mengirimkan 1 buah paket SMS. Hal ini dicontohkan pada Gambar 4.12 dengan mengirim sebuah pesan yang aslinya berjumlah 170 karakter atau dikirim dalam 2 paket SMS. Setelah dikompresi pesan hanya mengirim 127 karakter, atau hanya dikirim dalam 1 paket SMS

sehingga terjadi penghematan penggunaan karakter sebesar 26% dan penghematan biaya pulsa SMS sebesar 50%.



Gambar 4.14 Pengujian kedua kompresi

- c. Apabila hasil analisa aplikasi terhadap pesan yang dikirimkan, setelah dikompresi panjang pesan asli ternyata lebih banyak daripada jumlah maksimal pesan (160 karakter), tetap terjadi penghematan karakter tetapi tidak terjadi penghematan pulsa. Hal ini dicontohkan pada Gambar 4.13 dengan mengirim sebuah pesan yang aslinya berjumlah 266 karakter atau dikirim dalam 2 paket SMS. Setelah dikompresi pesan hanya mengirim 199 karakter, atau tetap dikirim dalam 2 paket SMS sehingga terjadi penghematan penggunaan karakter sebesar 26% tetapi penghematan pulsa 0%.



Gambar 4.15 Pengujian ketiga kompresi

4.2.1.2 Pengujian Pengiriman Pesan

Pada teknologi SMS terdapat dua jenis pengiriman pesan yaitu pengiriman jenis text dan pengiriman jenis binary (bit-bit data). Pada mode Normal, pesan akan dikirim dengan jenis pesan text dan pada mode Kompresi dilakukan pengiriman dengan jenis binary.

Apabila komponen komponen SMS seperti koneksi port komunikasi, format pesan yang dimasukkan, nomor telepon penerima yang tidak valid maka akan muncul laporan bahwa pengiriman pesan gagal.



Gambar 4.16 Pengiriman pesan berhasil

Apabila komponen komponen SMS seperti koneksi port komunikasi, format pesan yang dimasukkan, nomor telepon penerima maka akan muncul laporan bahwa pengiriman pesan gagal.



Gambar 4.17 Pengiriman pesan gagal

4.2.1.3 Pengujian Perbandingan Range Jumlah karakter SMS

Pada pengujian ini dilakukan 20 kali pengujian dengan 20 variasi jumlah karakter >160 karakter. Pengujian ini dilakukan menggunakan *Mobile Phone* Nokia N95. Pengujian ini merupakan kunci keberhasilan aplikasi SMSCompress sebagai aplikasi kompresi SMS yang memenuhi tujuan dari Tugas Akhir ini.

Adapun hasil dari pengujian ini mempunyai tujuan untuk mengetahui Range batas SMS untuk 1 kali SMS dan kelipatannya yang akan sangat berpengaruh terhadap biaya pengiriman SMS dan untuk mengetahui tingkat kompresi terbaik untuk jumlah karakter di atas 160 karakter.

Selain melakukan pengujian menggunakan mobile phone, juga dilakukan pengujian menggunakan *emulator*. Setiap pesan yang akan dikirimkan akan ditampilkan info pesan bersamaan ketika form nomor telepon muncul dan info pesan tentang jumlah SMS yang ditampilkan sesuai dengan biaya SMS yang terpotong menggunakan mobile phone.

Tabel 4.1 Pengujian Perbandingan Range Jumlah karakter SMS

No Pengujian	JUMLAH KARAKTER	SMS BIASA	SMS KOMPRESI	KOMPRESI Dalam %
1	161	2	1	19
2	162	2	1	20
3	163	2	1	21
4	164	2	1	31
5	165	2	1	25
6	166	2	1	21
7	167	2	1	20
8	168	2	1	26
9	169	2	1	21
10	197	2	2	26
11	202	2	2	24

12	321	3	2	16
13	322	3	2	16
14	323	3	2	16
15	324	3	2	16
16	374	3	2	34
17	482	4	3	32
18	483	4	3	34
19	580	4	3	33
20	642	5	4	25

Berdasarkan Tabel 4.1 dapat diketahui bahwa rata-rata hasil kompresi untuk setiap range sejumlah 30% untuk jumlah karakter di atas 160 karakter.

Berdasarkan pengujian menggunakan mobile phone dan emulator maka dapat diketahui batas rata-rata range SMS untuk tiap jenisnya seperti pada tabel 4.2.

Tabel 4.2 Rata-rata Range Jumlah Karakter SMS

RANGE	SMS BIASA	SMS KOMPRESI
1 KALI SMS	0	0
	160 karakter	170 Karakter
2 KALI SMS	161 Karakter	171 Karakter
	320 Karakter	330 Karakter
3 KALI SMS	321 Karakter	331 Karakter
	480 karakter	490 Karakter
4 KALI SMS	481 Karakter	491 Karakter
	640 Karakter	650 Karakter

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat diperoleh kesimpulan sebagai berikut:

1. Aplikasi SMSCompress dapat diimplementasikan dengan baik pada Mobile Phone yang mendukung MIDP 2.0 dan konfigurasi CLDC 1.0.
2. Algoritma Dynamic Markov Compression dapat diimplementasikan serta cukup optimal dalam melakukan proses pengompresan teks SMS dengan menghasilkan rasio file hasil kompresi mencapai rata-rata 60%.
3. Perancangan aplikasi dengan menggunakan metode berorientasi objek dapat mempermudah dalam pengembangan sistem secara keseluruhan, sebagai contoh ketika ingin menambahkan suatu fungsi tertentu, cukup dengan menambahkan fungsi tersebut ke dalam *class*. Demikian pula ketika diimplementasikan dalam kode program, dengan menggunakan pemrograman berorientasi objek, fungsi tambahan tersebut cukup ditambahkan ke dalam *class* yang sesuai.
4. Aplikasi kompresi SMS dapat memudahkan pengguna telepon seluler dalam melakukan penghematan penggunaan karakter dan biaya pulsa dalam melakukan pengiriman SMS tergantung jumlah karakter yang digunakan.

5.2 Saran

Beberapa hal yang menjadi kelemahan aplikasi dari pengujian yakni :

1. Hasil kompresi terhadap teks masih belum maksimal sehingga belum mencapai kompresi dari 3 *page* menjadi 1 *page* dalam pengiriman SMS dengan jumlah karakter yang besar.
2. Menu dalam aplikasi kompresi SMS masih relatif sederhana.
3. Tabel probabilitas dalam aplikasi yang dibangun karakternya masih sedikit sehingga mempengaruhi kompresi teks.

Beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Memaksimalkan hasil kompresi dengan mengembangkan algoritma kompresi yang sudah ada atau menggunakan teknik algoritma kompresi lainnya.
2. Perlu adanya pengembangan menu yang lain agar lebih bermanfaat bagi pengguna.
3. Memaksimalkan fungsi Aplikasi dengan penambahan karakter yang lebih banyak dengan tabel probabilitas yang lebih baik dan penanganan floating point.
4. Perlu dilakukan pengujian analisis mengenai metode-metode kompresi yang lebih mendalam agar mendapatkan metode kompresi yang terbaik untuk digunakan dalam aplikasi *mobile*.

DAFTAR PUSTAKA

- [HOR86] R. N. Horspool and G. V. Cormack, *Data Compression Using Dynamic Markov Modelling*, International Conference on the System Sciences, 1986, Honolulu.
- [WIC02] Ady Wicaksono, *Pemrograman Aplikasi Wireless dengan Java*, Elex Media Computindo, 2002, Jakarta.
- [HAR03] A Aditya Hartanto, *Java 2 Micro Edition Mobile Interface Device Programming*, Elex Media Computindo, 2003, Jakarta.
- [HEN05] Arnoud Hendry, Vincent Jonack, *SMS and MMS Interworking in Mobile Networks*, TeAM YYePG, 2005.
- [HEN06] Hendrik, *Modul Praktikum Rekayasa Perangkat Lunak*, Lab Sirkel, Teknik Informatika UII, 2006, Yogyakarta.
- [MUN06] Rinaldi Munir, *Diktat Kuliah Strategi Algoritmik*, Program Studi Teknik Informatika ITB, 2006, Bandung.
- [SHA06] Shalahudddin, M, Rosa A.S. *Pemograman J2ME, Belajar Cepat Pemograman Perangkat Telekomunikasi Mobile*. Informatika. 2006., Bandung.
- [SUL06] Sulistyanto, Hernawan. *Kompresi Data Lossless dengan Metode Lempel-Zip*. Surakarta: Jurnal Teknik Elektro dan Komputer EMITOR. 2006., Bandung.
- [HAJ09] Ibnu Hajar, *Implementasi Metode Lempel Ziv Welch (LZW) dalam aplikasi kompresi teks pada layanan sms*, Teknik Informatika UII, 2009, Yogyakarta