

APLIKASI MESSAGE CENTER: MODUL ANTAR MUKA ANTARA HANDPHONE DENGAN KOMPUTER

Budi Laksono Putro

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Sangga Buana YPKP

Jl. PHH Mustafa 68 Bandung 40123; Tlp. (022) 7275489, Faks. 7201756

E-mail: putromail@yahoo.com

ABSTRAK

SMS (Short Message Service) merupakan sebuah fitur aplikasi GSM (Global System For Mobile Communication), yang dikembangkan dan distandardisasi oleh ETSI (European Telecommunication Standards Institute). Meskipun sudah banyak fitur dari GSM seperti: MMS (Multimedia Message Service) dan GPRS (General Packet Radio Service), tetapi keberadaan jasa dan industri yang menggunakan SMS semakin lama semakin banyak dijumpai karena biayanya relatif murah. Hal tersebut didukung oleh faktor hardware yang semakin terjangkau. Adapun, jasa atau servis yang menggunakan SMS antara lain perbankan, seperti SMS Banking. Pada industri TI (Teknologi Informasi), data center yang memanfaatkan SMS untuk mengawasi kinerja dari server. Masih banyak lagi jasa dan industri yang menggunakan media ini.

Adapun yang permasalahan pada penelitian ini yaitu bagaimana membuat interface antara Komputer dengan HP (Hand Phone) sehingga bisa mengirim dan menerima SMS dengan GSM modem atau HP pada komputer yang kemudian dikenal dengan message center. Tujuan yang hendak dicapai dalam penelitian ini adalah membuat antar muka message center antara HandPhone GSM dengan komputer yang nantinya bisa diterapkan untuk suatu aplikasi (tergantung pada pemanfaatan end user).

Kata Kunci: SMS (Short Message Service, GSM (Global System For Mobile Communication), ETSI (European Telecommunication Standards Institute), MMS (Multimedia Message Service), GPRS (General Packet Radio Service), GSM Modem, Message Center

1. PENDAHULUAN

1.1 Latar Belakang

SMS (Short Message Service) merupakan sebuah fitur aplikasi GSM (Global System For Mobile Communication), yang dikembangkan dan distandardisasi oleh ETSI (European Telecommunication Standards Institute). Meskipun sudah banyak fitur dari GSM seperti: MMS (Multimedia Message Service) dan GPRS (General Packet Radio Service), tetapi keberadaan jasa dan industri yang menggunakan SMS semakin lama semakin banyak dijumpai karena biayanya relatif murah. Hal tersebut didukung oleh faktor hardware yang semakin terjangkau. Adapun, jasa atau servis yang menggunakan SMS antara lain perbankan, seperti SMS Banking. Pada industri TI (Teknologi Informasi), data center yang memanfaatkan SMS untuk mengawasi kinerja dari server. Masih banyak lagi jasa dan industri yang menggunakan media ini.

1.2 Rumusan Masalah

Dari latar belakang di atas, rumusan masalah pada penelitian ini yaitu bagaimana membuat *interface* antara Komputer dengan HP (Hand Phone) sehingga bisa mengirim dan menerima SMS dengan GSM modem atau HP pada komputer yang kemudian dikenal dengan *message center*.

1.3 Batasan Masalah

Batasan masalah pada penelitian ini adalah mengenai pembuatan *message center*, yaitu:

- Aplikasi yang dibuat sebagai *interface* antara HP dengan komputer.
- Jenis HP yang digunakan adalah jenis HP yang mempunyai format PDU dengan *interface* ke komputernya menggunakan *com serial(RS-232)*.
- Membuat penterjemah PDU (*Protocol Data Unit*) Format yang dihasilkan oleh HP ke dalam *Text Format*, pengkonversian kembali *Text Format* ke PDU Format, kemudian pembuatan aplikasi *server* sebagai *terminal* untuk proses penerimaan dan pengiriman pesan.

1.4 Tujuan

Tujuan penelitian ini adalah membuat antar muka *message center* antara HandPhone GSM dengan komputer yang nantinya bisa diterapkan untuk suatu aplikasi (tergantung pada pemanfaatan *end user*).

2. LANDASAN TEORI

2.1 Global System For Mobile Communication (GSM)

Mobile communication (komunikasi bergerak) mulai ramai dibicarakan sejak tahun 1982, pada waktu dibentuknya suatu badan khusus, yaitu GSM (*Group Special Mobile*) dalam CEPT (*Conference Uropeenne des Postes et Telecommunications*), untuk menentukan standar yang digunakan dalam jaringan komunikasi bergerak.

Sejak tahun 1990, terdapat spesifikasi esensi dari GSM, dan mulai saat itu GSM merupakan

singkatan dari *Global System For Mobile Communication*. Pada tahun 1991, jaringan operasional dari GSM baru dimulai, sehingga masyarakat umum mengenal istilah GSM sejak tahun 1992. Selanjutnya, teknologi telekomunikasi bergerak ini berkembang terus hingga muncul sebagai penawaran layanan yang lebih baik.

Pada awalnya teknologi ini dirancang pada frekuensi 900 MHz (GSM 900). Pada perkembangan selanjutnya, teknologi GSM mulai dioperasikan pada frekuensi 1800 MHz dan desain GSM ini memiliki suatu layanan tambahan berupa SMS.

2.2 Short Message Service (SMS)

SMS diperkenalkan pertama kali pada tahun 1991 di Eropa. SMS merupakan salah satu *feature* GSM yang dapat mentransfer *short message* antar GSM MS (*Mobile Station*) dan SME (*Short Message Entity*) melalui SC (*Service Center*).

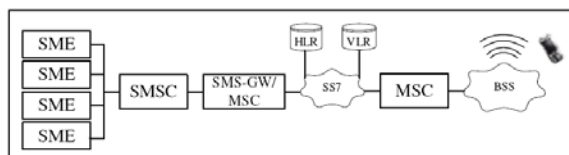
Adapun karakteristik dari SMS yaitu:

- SMS merupakan sebuah pesan singkat terdiri atas 160 karakter yang menampung huruf, angka dan *alpha numeric*.
- Prinsip kerja SMS adalah “menyimpan” dan “menyampaikan” (*Store And Forward*) pesan. Dengan kata lain pesan tidak langsung dikirim ke tujuan akan tetapi disimpan dahulu di SC.
- Fasilitas SMS memiliki layanan informasi tentang pengiriman pesan.
- Apabila pesan berhasil dikirim maka SC akan mengirimkan *delivery report* kepada MS, jika pesan gagal dikirim maka SC akan mengirimkan *failure report* kepada MS.

Arsitektur Jaringan SMS

Untuk implementasi layanan SMS operator menyediakan SMSC, secara fisik SMSC dapat berwujud sebuah PC biasa yang mempunyai interkoneksi dengan jaringan GSM.

Pada Gambar 2.1, di sebelah kiri dapat dilihat SMSC memiliki interkoneksi dengan SME yang dapat berupa jaringan *e-mail*, *Web*, dan *voice e-mail*. SMSC inilah yang akan melakukan manajemen pesan SMS, baik untuk pengiriman, pengaturan antrian SMS, atau penerimaan SMS.



Gambar 2.1 Elemen Jaringan dan Arsitektur Dasar SMS

Short Messaging Entities

SME adalah suatu piranti yang dapat menerima atau mengirim pesan pendek. SME dapat berada dalam jaringan *fixed*, sebuah piranti bergerak, atau pusat layanan (*service center*) lainnya seperti:

- VMS.

- Web.
- E-Mail
- Lain-lain, seperti jaringan penyeranta, perangkat lunak khusus untuk penulisan pesan berbasis PC dan operator.

Short Message Service Center (SMSC)

SMSC adalah kombinasi perangkat keras dan perangkat lunak yang bertanggung jawab memperkuat, menyimpan dan meneruskan pesan pendek antara SME dan piranti bergerak.

SMSC harus memiliki kehandalan, kapasitas pelanggan, dan *throughput* pesan yang tinggi. Selain itu, SMSC juga harus dapat diskalakan dengan mudah untuk mengakomodasi peningkatan permintaan SMS dalam jaringan yang ada.

Faktor lain yang harus dipertimbangkan adalah kemudahan pengoperasian dan pemeliharaan aplikasi, dan juga fleksibilitas untuk mengaktifkan layanan baru dan meng-*upgrade* perangkat lunak versi terbaru.

SMS-Gateway dan SMS-Interworking Mobile Switching Center

SMS Gateway Mobile Switching Center (SMS-GMSC) adalah sebuah aplikasi MSC yang mampu menerima pesan singkat dari SMSC, menginterogasi *home location register* (HLR) untuk informasi *routing*, dan mengirimkan pesan pendek tersebut ke MSC dari piranti bergerak yang dituju.

SMS Interworking Mobile Switching Center (SMS-IWMSC) adalah aplikasi MSC yang mampu menerima pesan pendek dari jaringan bergerak dan mengirimkannya ke SMSC yang tepat. SMS-GMSC/SMS-IWMSC biasanya terintegrasi dengan SMSC.

Home Location Register

HLR adalah basis data yang digunakan untuk penyimpanan permanen, pengelolaan langganan dan profil layanan. Ketika diterogasi oleh SMSC, HLR memberikan informasi *routing* mengenai pelanggan yang ingin dituju.

HLR juga dapat memberitahu SMSC, yang sebelumnya mengalami kegagalan usaha pengiriman pesan pendek ke piranti bergerak tertentu, bahwa sekarang piranti *mobile* tersebut telah dikenali oleh jaringan bergerak, dan dengan demikian pesan telah dapat dikirimkan.

Mobile Switching Center

Mobile Switching Center (MSC) melakukan fungsi penyaklaran sistem dan mengendalikan panggilan ke dan dari sistem telepon dan data yang lain. MSC akan mengirimkan pesan pendek ke pelanggan tertentu melalui *base station* yang sesuai.

Visitor Location Register

Visitor Location Register (VLR) adalah basis data yang berisi informasi temporal mengenai

pelanggan yang berasal dari suatu HLR yang *roaming* ke HLR lainnya. Informasi ini dibutuhkan oleh MSC untuk melayani pelanggan yang berkunjung.

Base Station System

Semua fungsi yang terkait dengan transmisi sinyal radio elektromagnetis antara MSC dan piranti bergerak dilakukan di *Base Station system* (BSS). BSS terdiri dari *Base Station Controllers* (BSCs) dan *Base Transceiver Stations* (BTSs), juga dikenal sebagai wilayah sel/sel.

BSC dapat mengendalikan satu atau lebih BTS dan bertanggung jawab dalam pemberian sumber data yang semestinya ketika pelanggan bergerak dari satu sektor suatu BTS ke sektor lain, terlepas dari apakah sektor berikutnya tersebut berada dalam BTS yang sama atau berbeda.

Layanan Pelanggan

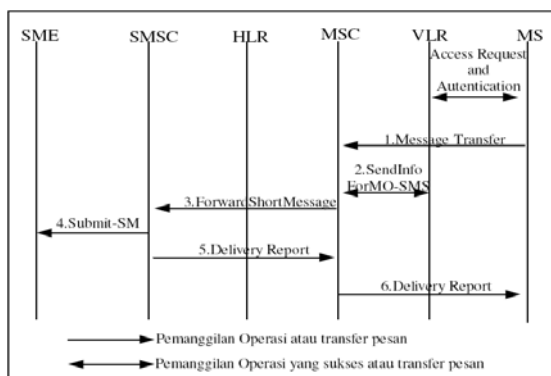
Sistem SMS memiliki dua layanan dasar *point-to-point* bagi pelanggan, yaitu:

A. Mobile-Originated Short Message

Mobile-Originated (MO) *Short Message* dikirimkan dari suatu handset yang *MO-capable* ke SMSC dan dapat ditujukan ke pelanggan bergerak lainnya atau pada pelanggan di jaringan *fixed* seperti jaringan penyeranta atau jaringan *Internet Protocol* (IP).

Mobile-Terminated (MT) *Short Messages* dikirimkan dari SMSC ke *handset* dan dapat sampai ke SMSC dari pelanggan bergerak yang lain melalui MO-SM atau sumber lain seperti sistem *voice-mail*, jaringan penyeranta, atau operator.

Pada layanan MO-SM selalu ada laporan yang dikirimkan ke handset, baik yang mengkonfirmasi pengiriman pesan pendek ke SMSC maupun menginformasikan kegagalan pengiriman dan mengidentifikasi penyebabnya. Adapun gambaran dari proses dari MO-SM tertera pada Gambar 2.2.

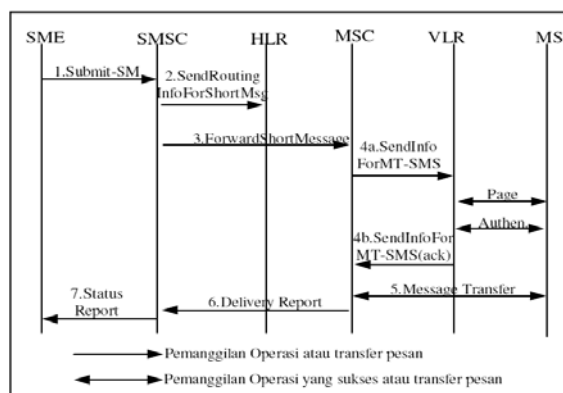


Gambar 2.2 Skenario MO-SM

B. Mobile-Terminated Short Message

Untuk MT-SM, juga selalu ada laporan yang diberikan kepada SMSC yang isinya bisa berupa konfirmasi pengiriman pesan pendek ke *handset*

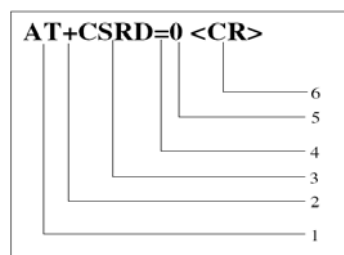
maupun informasi kegagalan pengiriman pesan dan mengidentifikasi penyebab kegagalan tersebut (*cause code*). Adapun gambaran proses dari MT-SM seperti yang tertera pada Gambar 2.3.



Gambar 2.3 Skenario MT-SM

2.3 AT Commands

AT Commands yaitu perintah untuk menjalankan fungsi HP melalui jalur *serial*. *AT Command* merupakan sebuah kumpulan *command* sesuai dengan ETSI GSM 07.05 dan GSM 07.07. Penulisan *syntax* dari *AT Command* sesuai dengan aturan yang ditentukan oleh ETSI dan GSM. Aturan penulisan *syntax* sebagai berikut:



Gambar 2.4 Aturan Penulisan *Syntax AT Commands*

1. Setiap *command* diawali oleh string "AT".
2. Dapat dilanjutkan dengan tanda "+", "*", atau kosong.
3. *Command* yang diinginkan.
4. Diikuti tanda sama dengan ("="), tetapi tergantung *command* yang digunakan.
5. Nilai yang diberikan pada *command* tersebut.
6. Diakhiri dengan ASCII "CR" (0D h).

2.4 PDU Format

SMS adalah salah satu *feature* yang diberikan oleh jaringan GSM. Pengiriman SMS berdasarkan format sesuai dengan ETSI (GSM 03.40 dan GSM 03.38) dapat dilakukan dengan tiga cara yaitu :

- Pengiriman Pesan Dengan *TEXT Format*,
- Pengiriman Pesan Secara Blok,
- Pengiriman Pesan dengan *PDU Format*.

Dalam aplikasinya pengiriman pesan secara blok di Indonesia belum diterapkan sehingga pembahasan tidak akan diikutsertakan. Pengiriman pesan dengan menggunakan Format Teks dapat dilakukan oleh beberapa tipe dari HP tertentu.

Format PDU merupakan suatu cara pengiriman pesan yang banyak digunakan oleh HP. Hal ini dikarenakan penulisan menjadi lebih mudah karena menggunakan data 7 bit. Dengan PDU ini pengiriman menjadi lebih cepat karena PDU memungkinkan 8 data dikirimkan hanya dalam 7 data. Dalam proses pengiriman dan penerimaan data mempunyai perbedaan tersendiri sehingga diperlukan suatu cara guna meng-encoding data tersebut.

3. IMPELMANTASI

3.1 Menerima Pesan dalam Bentuk PDU

Menerima pesan dalam bentuk PDU tidak hanya isi pesan saja, melainkan terdapat berbagai data di dalamnya seperti informasi mengenai pengirimnya (nomor telepon pengirim), SMSC, dan Waktu Pengiriman Pesan. Data yang masuk berupa Hexa – Decimal Octets. Contoh pesan yang diterima oleh HP sebagai berikut:

206912618010000240D91261823408055F60000408
0318054408010D4FA393C0705D7E8B41C142497
5D2.

Berdasarkan data diatas maka dapat dikelompokkan berdasarkan posisinya sesuai dengan yang tertera pada Tabel 3.1.

Tabel 3.1 Contoh Pengelompokan Data PDU

Oktet(Data)	Keterangan
06	Banyaknya informasi SMSC (Hexa Oktet)
91	Tipe alamat dari SMSC (Hexa Oktet)
2618010000	Nomor SMSC dari pengirim (Desimal semi Oktet)
24	Ocet pertama dari pesan SMS-Deliver (Hexa Oktet)
0D	Banyaknya nomor dari nomor pengirim (Hexa Oktet)
91	Tipe alamat dari nomor pengirim (Hexa Oktet)
261823408055F6	Nomor telepon pengirim pesan (Desimal Semi Oktet)
00	Protokol Identifier (protokol pengenalan) (Hexa Oktet)
00	Data Coding Scheme (Hexa Oktet)
40803180544080	Waktu pengiriman pesan (Desimal semi Oktet)
10	Banyaknya pesan yang dikirim (Hexa Oktet)
D4 FA 39 3C 07 05 D7 E8 B4 1C 14 24 97 5D	Pesan (Hexa Oktet) dalam 7 bit-data

Penjelasan dari keterangan pada data dengan format PDU dari tabel 3.1 yaitu:

- SMSC, yaitu menerangkan banyaknya informasi pengirim yang terdapat pada pesan yang digunakan oleh pengirim. Contohnya “06” hexa ini berarti terdapat 6 byte informasi sesudah “06” yaitu “91 26 18 01 00 00”.
- SMS Service Number, berisikan nomor service center yang digunakan dalam pengiriman pesan. Penulisan dari nomor ini di tukar tempat antara Low nible dan High nible-nya. Apabila jumlah nomornya ganjil maka diakhir nomor ditambahkan “F” berarti ditambah tanda “+” didepan nomor dari service center-nya. Data yang diterima adalah “2618010000 “ bila telah ditukar low nible dan high nible-nya menjadi “628100000”, nomor ini merupakan nomor sebenarnya dari service center yang digunakan.

- Tipe nomor telepon pengirim, merupakan indikasi dari format penulisan nomor telepon pengirim, contohnya “91” Merupakan indikasi bahwa
- nomor telepon pengirim menggunakan format International.
- Nomor telepon pengirim, untuk meng-coding nomor telepon pengirim dilakukan dengan menukar antara low nible dan high nible dari byte-byte informasi nomor telepon tersebut. Dalam contoh diberikan “261823408055F6”. “26” Merupakan contoh dari 1 byte data, kemudian tiap 1 byte data ditukar antara low nible dan high nible-nya, sehingga nomor telepon pengirim dapat diketahui, seperti pada contoh setelah dicoding menjadi “6281320408556F”.
- PID, merupakan elemen informasi yang menyatakan protokol layer tertinggi yang digunakan.
- Time-stamp, waktu pengiriman pesan merupakan suatu informasi yang menyatakan waktu pengiriman dari pesan tersebut. Pengkodean dari time-stamp dimulai dari “Tahun Bulan Hari Jam Menit Detik ZonaWaktu”. Berdasarkan contoh data waktu yang diterima adalah “40803180544080”. Bila pengkodeannya dengan menukar antara low nible dan high nible dari tiap byte data maka akan didapat “04 08 13 08 45 04 08”. Bila dituliskan dengan menggunakan aturan waktu menjadi “2004 Agustus 13 08:45:04 GMT+(untuk waktu Indonesia)”.
- 10 merupakan informasi dari banyaknya karakter pesan yang dikirim (Hexa octet), nilai dari 10 (Hexa Octet) yaitu 16 (desimal) karakter.
- Pesan dalam bentuk oktet dikonversikan menjadi septet (7-bit). Aturan konversinya berdasarkan contoh data 7-bit yang tertera pada Tabel 3.2.

Tabel 3.2 Konversi Format PDU menjadi Format Text

Hex	D4	FA	39	3C	07	05	D7	*****
Oktet	1010100	111010	011001	0101100	0101111	0101011	1010101	
Septet	1010100	111010	11001	1100100	1110101	0101011	1010101	
Decimal	84	117	103	97	115	32	65	107
Character	T	u	g	a	s	<Spasi>	A	k

Hex	E8	B4	1C	14	24	97	5D	*****
Oktet	1101000	110100	111100	1010100	1010100	1010111	1010101	
Septet	1101000	110100	11100	0100100	1001011	1101011	1010101	
Decimal	104	105	114	32	65	100	101	46
Character	h	i	r	<Spasi>	A	d	e	.

- Setiap data dipisahkan bit depan sesuai dengan data tersebut yang keberapa. Contoh data “D4” adalah data pertama susunan bit-nya adalah (11010100). Karena data pertama maka dipisahkan satu bit depannya menjadi (11010100). Kemudian data kedua yang didapat “FA” susunan bit-nya adalah (1111010), karena data kedua maka diambil dua bit

depannya menjadi (11111010). Begitu seterusnya hingga data ke-7 kemudian data ke delapan kembali seperti aturan untuk data ke -1.

- *Bit* depan yang telah dipisahkan kemudian ditambahkan untuk *bit* bawah data yang berikutnya. Data pertama setelah dipisahkan (11010100) sehingga data ke-1 adalah (1010100) = 54(hexa) = 84(desimal). Data kedua setelah dipisahkan adalah (11111010) dan data kedua setelah ditambahkan *bit* bawah dari *bit* depan data pertama menjadi (1110101)=71 hexa = 117(desimal). Bila tidak terdapat data berikutnya berarti *bit* depan data terakhir merupakan tanda dari berhenti pengkodean data tersebut. "00" merupakan tanda berhenti dari pengkodean tersebut.

3.2 Mengirim Pesan dalam format PDU

Pesan yang dituliskan dalam *format text* akan di konversikan terlebih dahulu kedalam format PDU agar bisa di baca oleh HP. Dimana konversinya merupakan kebalikan dari terima pesan, seperti yang tertera pada Tabel 3.3.

Tabel 3.3 Konversi *Format Text* menjadi *Format PDU*

Character	K	i	r	i	m	<spasi>	S
Decimal	75	105	114	105	109	32	83
Septet	1001011	1101001	1110010	1101001	1101101	0100000	1010011
Oktet	1001011	1010100	0011100	1101100	0000010	0100110	1001101
Hex	CB	B4	3C	DD	06	4D	9B

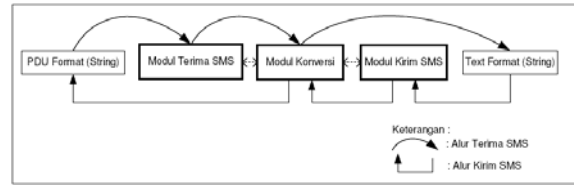
Character	M	S	<spasi>	A	d	e	!
Decimal	77	83	32	65	100	101	33
Septet	1001101	1010011	0100000	1000001	1100100	1110101	0100001
Oktet	0101101	0110100	1001000	1010100	0000110	0000001	0000001
Hex		53	50	90	5C	0E	01

Penjelasan dari tabel 3.3 yaitu:

- karakter yang akan dikirim di konversi kedalam desimal, contohnya karakter "K" desimalnya adalah 75, kemudian di konversi kedalam bentuk septet menjadi (1001011), dan selanjutnya setiap data berikutnya dipisahkan *bit* belakang sesuai dengan data tersebut yang beberapa,
- karakter "K" merupakan karakter pertama yang septetnya (1001011) maka oktetnya akan mengambil 1 digit data terakhir dan diletakan didepan dari karakter berikutnya yaitu "i" yang susunan septetnya (110100 1) menjadi bentuk oktet pertama dengan susunan *bit*-nya (11001011) = "CB" dalam hexa,
- data berikutnya dari karakter "i" karena sudah diambil 1 digit maka menjadi (110100) kemudian untuk oktetnya mengambil 2 digit terakhir data berikutnya yaitu karakter "r" (11100 10) menjadi oktet ke dua dengan susunan *bit*-nya (10110100), dan seterusnya sampai karakter terakhir, dimana jika pada karakter terakhir jumlah *bit*-nya kurang dari 8, maka ditambahkan digit nol "0" di depannya sampai jumlah digitnya 8.

3.3 Perencanaan Modul Sistem

Sistem yang akan dibuat terdiri dari tiga bagian, yaitu modul konversi, terima SMS dan kirim SMS. Adapun Gambar dari ketiga modul tersebut adalah sebagai berikut:



Gambar 3.1 Perencanaan Modul pada Sistem

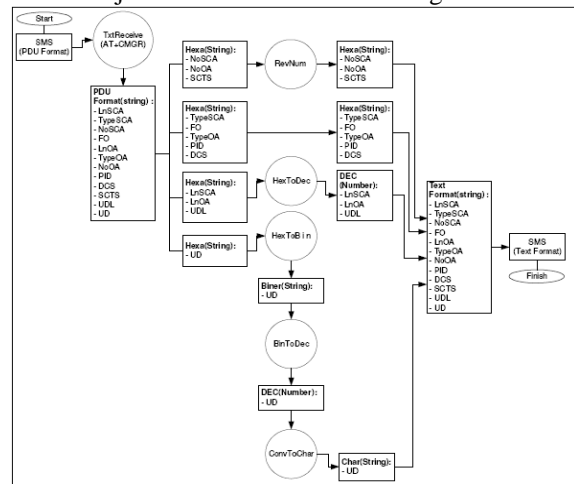
Modul Terima SMS

Modul ini berfungsi untuk menerima data PDU yang masuk dan menampung masing-masing bagian dari data tersebut, *input*-an pada prosedur ini yaitu *string* data PDU yang belum dipisahkan sesuai dengan nama bagiannya, sedangkan *output*-nya berupa *string* PDU yang sudah dipisahkan yang terdiri dari panjang nomor SMSC, tipe alamat SMSC, nomor SMSC, *Octet* pertama dari pesan SMS-Deliver, Banyaknya nomor dari nomor pengirim, tipe alamat dari nomor pengirim, nomor telepon pengirim pesan, *protokol identifier*, *data coding scheme*, waktu pengiriman pesan, banyaknya pesan yang dikirim dan isi pesan sesuai dengan nama bagiannya. *AT Command* yang digunakan untuk membaca pesan yang masuk yaitu:

$$AT+CMGR=[index message] + [Chr(26)]$$

Gambar 3.2 Perintah Membaca Pesan yang masuk

Alur proses terima SMS, yaitu konversi Format PDU menjadi Format Teks adalah sebagai berikut:



Gambar 3.3: Alur Proses konversi Format PDU Menjadi Format Teks AT+CMGR=[index message] + [Chr(26)]

Modul Kirim SMS

Modul ini berfungsi untuk menampung data yang akan dikirimkan yaitu dalam bentuk *format*

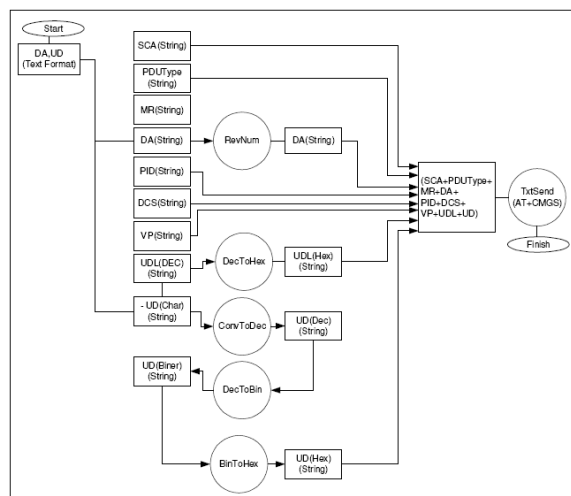
PDU, dimana dalam prosesnya akan memanggil modul konversi untuk merubah data dalam *Format Text* menjadi *Format PDU*. Adapun cara pengiriman pesan yaitu sebagai berikut:

```
AT+CMGS=[Panjang pesan dalam oktet] + [Chr(13) + Chr(10)] + [SCA + PDU(Type) + MR + DA + PID + DCS + VP + UDL + UD]+[Chr(26)]
```

Gambar 3.4 Perintah Kirim Pesan dalam Mode PDU

Respon dari perintah tersebut yaitu "Ok " (jika berhasil) atau "Error+(kode)" (jika gagal).

Alur proses kirim SMS, yaitu dari masukan berupa Format Teks dikonversi ke bentuk Format PDU tertera pada Gambar 3.5.



Gambar 3.5 Alur Proses Kirim SMS

Modul Konversi

Modul ini fungsinya untuk menterjemahkan setiap data yang masuk dari:

- Modul terima SMS (PDU *Fomat*) menjadi (*Text Format*),
 - o membalikan posisi (*reverse*) data PDU yang diperlukan oleh modul terima SMS (*noSCA, noOA, SCTs*),
 - o pengkonversian informasi dari (*LenSCA, LenOA, dan UDL*) dari *string Hexa* menjadi Desimal,
 - o pengkonversian informasi dari (*UD*) dari *string Hexa* menjadi *Biner, Biner* menjadi *Decimal, Decimal* menjadi *Character*.
- Modul Kirim SMS (Format Teks) menjadi (Format PDU),
 - o membalikan posisi(*reverse*) data PDU yang diperlukan oleh modul kirim SMS (*DestinationNo*),
 - o data yang dikonversi yaitu *string Character* menjadi *Decimal, Decimal* menjadi *Biner, Biner* menjadi *Hexa*.

4. KESIMPULAN & SARAN

4.1 Kesimpulan

1. Alat yang dibuat sudah bisa dijadikan antarmuka antara handphone dengan

2. komunikasi serial RS-232 dengan kompatibel/IBM PC *Compatible*,
3. Pembacaan PDU Format didasarkan pada *AT Commands*,
4. Pembacaan isi SMS dengan format PDU, bisa dilakukan dengan melakukan konversi bilangan dari *hexa* ke bentuk *biner* (oktet), kemudian bentuk oktet ke bentuk *biner* (septet), bentuk septet dikonversi ke bentuk desimal, dan akhirnya diperoleh nilai ASCII dari desimal tersebut.

4.2 Saran

1. Pembacaan isi dari SMS sebaiknya dilakukan dengan pembuatan daftar karakter yang terdapat pada HP, sehingga nilai yang didapat setelah konversi ke bentuk desimal bisa menjadi suatu pointer ke daftar karakter yang telah dibuat. Dengan demikian perbedaan posisi karakter bisa diatasi.
2. Proses antara baca dan kirim sebaiknya dilakukan dengan *multi thread* pada masing-masing proses, sehingga proses yang terjadi bisa berjalan tanpa harus memutus suatu proses lain,
3. Pembuatan komponen untuk aplikasi SMS sebaiknya menggunakan suatu aplikasi yang *multi platform* pada suatu sistem operasi, contohnya menggunakan bahasa pemrograman *java/ C*, bisa dipakai pada sistem operasi *Windows* dan *Linux*.

PUSTAKA

1. Atmel, 2003. AT89C52. http://www.atmel.com/dyn/resources/prod_documents/doc0313.pdf (didownload 27 September 2003).
2. Bodic, G.U., 2002. *Mobile Messaging: SMS, EMS, and MMS, IEEE Vehicular Technology Society News*, November, pp 13-21.
3. ETSI, 1996, *Technical Realization of the Short Message Service (SMS) Point-to-Point (PP) (GSM 03.40)*. <www.mobilecity.cz/doc/GSM_03.40_5.3.0.pdf> (Didownload 17 September 2003).
4. Gupta, P., 2003. *Short Message Service: What, How and Where?* <www.wirelessdevnet.com/channels/sms/features/sms.html > (didownload 27 September 2003).
5. Istiyanto, J.E., 1995. *The Application of Architectural Synthesis to the Reconfiguration of FPGA-based Special Purpose Hardware*, Ph.D Thesis Department of Electronic Systems Engineering, University of Essex, U.K.
6. WAVECOM, 2000, An introduction to the SMS in PDU mode GSM Recommendation phase 2+. <www.ascend-tech.com.cn/sustain/SMS_PDU-mode.pdf> (Didownload 17 September 2003).