

Model Pengembangan Short Message Service (SMS) pada jaringan Seluler

Munir Tubagus

Institut Agama Islam Negeri (IAIN) Manado, Sulawesi Utara

munir.tubagus@iain-manado.ac.id

Abstract. *Short Message Service (SMS) is a service that is widely applied to wireless communication systems, allowing it to send messages in alphanumeric form between customer terminals or between customer terminals and external systems such as email, paging, voice mail. In the SMS system, the main mechanism carried out in the system is sending short messages from one customer terminal to another. This can be done thanks to the existence of an entity in the SMS system called Short Message Service Center (SMSC), also called the Message Center (MC). SMSC is a device that performs store tasks and forward short message traffic. It includes determining or searching the route for the final destination of the short message. An SMSC is usually designed to be able to handle short messages from various sources such as Voice Mail System (VMS), Web-based messaging, Email Integration, External Short Messaging Entities (ESME), and others. In interconnecting with entities in wireless communication networks such as Home Location registers (HLR) and Mobile Switching Centers (MSC), SMSCs usually always use Signal Transfer Points (STP).*

Keywords : *Information Technology, Short Message Service, Web-Based, Wireless Networks, Cellular Communication*

Abstrak. *Layanan Pesan Singkat (SMS) adalah layanan yang secara luas diterapkan pada system komunikasi nirkabel, yang memungkinkannya mengirim pesan dalam bentuk alfanumerik antara terminal pelanggan atau antara terminal pelanggan dan system eksternal seperti email, paging, voice email. Dalam sistem SMS, mekanisme utama yang dilakukan dalam sistem adalah mengirim pesan singkat dari satu terminal pelanggan ke terminal pelanggan lainnya. Ini bisa dilakukan berkat keberadaan entitas dalam system SMS yang disebut Pusat Layanan Pesan Singkat (SMSC), juga disebut Pusat Pesan (MC). SMSC adalah perangkat yang melakukan tugas menyimpan dan meneruskan lalu lintas pesan singkat. Ini termasuk menentukan atau mencari rute untuk tujuan akhir dari pesan singkat. Sebuah SMSC Biasanya dirancang untuk dapat menanggapi pesan singkat dari berbagai sumber seperti Voice Mail Sistem (VMS), pesan berbasis Web, Integrasi Email, Entri Perpesanan Singkat Eksternal (ESME), dan lainnya. Dalam Interkoneksi dengan entitas dalam jaringan komunikasi nirkabel seperti Home Location register (HLR) dan Mobile Switching Centers (MSC), SMSC biasanya selalu menggunakan Signal Transfer Points (STP).*

Keywords : *Informasi Teknologi, Layanan Pesan Singkat, Berbasis Web, Jaringan Nirkabel, Komunikasi Seluler*

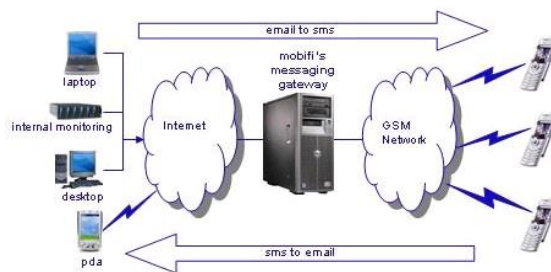
Pendahuluan

SMS atau Short Message Service merupakan salah satu layanan teknologi yang digunakan dalam melakukan pertukaran pesan melalui perangkat mobile seperti telepon genggam. SMS adalah cara penyimpanan dan langkah maju dalam melakukan transmisi pesan ke dan dari ponsel atau telepon genggam. Pesan dari ponsel pengirim disimpan dalam pusat pesan singkat (SMS Center) yang kemudian diteruskan ke perangkat tujuan. Setiap pesan singkat tidak lebih dari 160 karakter, sementara karakter ini dapat berupa teks (alfanumerik) atau biner pesan pendek Non-Tek.¹

Layanan SMS merupakan sebuah layanan yang bersifat nonreal time di mana sebuah short message dapat di submit ke usatu tujuan, tidak peduli apakah tujuan tersebut aktif atau tidak. Bila dideteksi bahwa tujuan tidak aktif, maka sistem akan menunda pengiriman ke tujuan hingga tujuan aktif kembali. Pada dasarnya sistem SMS akan menjamin delivery dari suatu short message hingga sampai tujuan. Kegagalan

pengiriman yang bersifat sementara seperti tujuan tidak aktif akan selalu teridentifikasi sehingga pengiriman ulang short message akan selalu dilakukan kecuali bila diberlakukan aturan bahwa short message yang telah melampaui batas waktu tertentu harus dihapus dan dinyatakan gagal terkirim.

Karakteristik utama SMS adalah SMS merupakan sebuah sistem pengiriman data dalam paket yang bersifat out-of-band dengan bandwidth kecil. Dengan karakteristik ini, pengiriman suatu burst data yang pendek dapat dilakukan dengan efisiensi sangat tinggi. Arsitektur dan Elemen Jaringan SMS Layanan SMS dibangun dari berbagai entitas yang saling terkait dan mempunyai fungsi dan tugas masing-



masing. Tidak ada satu pun dalam sistem SMS yang dapat bekerja secara parsial. Entitas dalam jaringan ini disebut juga elemen jaringan SMS. Secara umum arsitektur sistem SMS, khususnya untuk

¹ Katankar, V.K., Thakare, V.M., Short Message Service using SMS Gateway, (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 04, 2010, 1487-1491

sistem yang diintegrasikan dengan jaringan wireless adalah sebagai berikut:

Gambar 01: Arsitektur Sistem Short Message Service (SMS)

1) Protocol Data Unit

Dalam pengiriman dan penerimaan pesan SMS terdapat dua mode yaitu mode teks dan mode Protocol Data Unit (PDU). Mode teks adalah format pesan dalam bentuk teks asli yang dituliskan pada saat mengirimkan pesan. Sesungguhnya kode teks ini adalah hasil pengkodean dari mode PDU. Sedangkan mode PDU adalah format pesan dalam bentuk oktet heksadesimal dan oktet semidesimal dengan panjang mencapai 160 (7 bit) atau 140 (8 bit) karakter. Pada pengiriman pesan terdapat dua jenis mobile, yaitu Mobile Terminated (Handphone Penerima) dan Mobile Originated (Handphone Pengirim).

SMS PDU Pengirim (Mobile Originated) SMS PDU Pengirim adalah pesan yang dikirim dari handphone ke terminal yang kemudian dikirimkan ke SMSC. Pada prinsipnya apabila kita mengirim pesan ke nomor tujuan, pesan itu akan melalui SMSC. Pesan yang akan dikirimkan oleh terminal masih dalam

bentuk teks, sedangkan dalam pengiriman ke SMSC harus dalam bentuk PDU. Untuk itu sebelum dikirim terminal atau handphone akan melakukan perubahan dari format teks menjadi format PDU, proses ini sering disebut proses encodec. Adapun skema dari format PDU Pengirim telah diatur dan ditetapkan oleh ETSI sebagai berikut:

SC A	PDU Type	MR	DA	PID	DCS	VP	UD L	UD
---------	-------------	----	----	-----	-----	----	---------	----

Skema Format SMS PDU Pengirim

Misalnya kita mengirim pesan SMS ke nomor 628122898840 dengan isi pesan "pesan pendek" dengan batas waktu pengiriman (waktu penyimpanan pesan di SMSC, jika nomor tujuan tidak dapat menerima pesan) 6 hari. Maka format PDU adalah:

0011000000C912618229888040000AB0
CD0F23CEC06C1CB6E72790D

2) Service Centre Adress (SCA)

SCA adalah informasi dari alamat (nomor) SMSC. SCA memiliki tiga komponen utama, yaitu len, type of number, dan service center number.

Dalam pengiriman pesan SMS, nomor SMSC tidak dicantumkan²

Oktet	Hasil	Keterangan
Len	00	Panjang informasi SMSC dalam oktet
Type of number	<none>	Format nomor dari SMSC 81 hexa = format lokal 91 hexa = format international
Service center number	<none>	Nomor SMSC dari operator pengirim. Jika panjangnya ganjil maka pada karakter terakhir ditambahkan 0F hexa

Tabel 01: Service Center Address

Jadi pada SCA hasilnya adalah 00.

PDU Type Nilai default dari PDU Type untuk SMS pengirim adalah 11 hexa, yang memiliki arti bahwa 11 hexa = 00000100

Bit no	7	6	5	4	3	2	1	0
Nama	RP	UDHI	SRR	VPF	VPF	RD	MTI	MTI
Nilai	0	0	0	1	0	0	0	1

Tabel 02: PDU Type

Keterangan;

RP : Reply Path. Parameter yang menunjukkan bahwa alur jawaban yang ada.

UDHI : User Data Header indicator. Bit ini bernilai 1 jika data pengirim dimulai dengan suatu judul/tema.

SRR : Status Report Request. Bit ini bernilai 1 jika laporan status pengiriman diminta.

VPF : Validity period format, Format dari batas waktu pengiriman jika pesan gagal diterima. 00 → Jika pesan tidak disimpan di SMSC.

10 → Format relatif (satu oktet)

01 → Format enhanced (tujuh oktet)

11 → Format absolut (tujuh oktet)

RD: Reject Duplicates. Parameter yang menandakan ya atau tidaknya Service Center akan menerima suatu pengiriman pesan SMS untuk suatu pesan yang masih disimpan dalam Service Center tersebut.

MTI : Message Type Indicator. Bit bernilai 0 untuk menunjukkan bahwa PDU ini adalah suatu SMS-DELIVER.

3) Message Reference (MR)

Message Reference adalah acuan dari pengaturan pesan SMS. Untuk membiarkan pengaturan pesan SMS dilakukan sendiri oleh handphone tujuan, maka nilai yang diberikan adalah '00'. Jadi pada Message Reference hasilnya adalah 00. Destination Address adalah alamat (nomor) tujuan, yang terdiri atas panjangnya nomor tujuan (Len), format

² Gwenaél Le Bodic, *Mobile Messaging Technologies and Services SMS, EMS and MMS*. (England. John Wiley and Sons, Ltd., 2005) h. 114.

dari nomor tujuan (Type Number), dan nomor tujuan (Destination Number).

Oktet	Nilai	Keterangan
Len	12	0C
Type of number	Format Internasional	91
Service center number	628122898840	261822988804

Tabel 02: Destination Address

Jadi pada Destination Address hasilnya adalah 0C91261822988804.

4) Protocol Identifier (PID)

Protocol Identifier adalah tipe atau format dari cara pengiriman pesan, yang biasanya diatur dari handphone pengirim. Misalnya tipe standard text, fax, E-mail dan lain-lainnya. Nilai default dari PID adalah 00 = "Standard Text". Pada contoh ini, pesan SMS yang akan dikirim menggunakan format teks standar, jadi pada Protocol Identifier hasilnya adalah 00.

5) Data Coding Scheme (DCS)

Data Coding Scheme adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa SMS teks standar, Flash SMS, atau Blinking SMS. Pada contoh ini pesan SMS yang dikirim berupa teks standar, jadi pada Data Coding Scheme hasilnya adalah 00.

6) Validity Period (VP)

Validity Period adalah lama waktu pesan SMS disimpan di SMSC apabila pesan tersebut gagal diterima oleh handphone penerima.

Waktu VP	Nilai VP
5 menit - 720 menit (12 jam)	$(Waktu\ VP / 50 - 1)$
12,5 jam - 24 jam	$143 + (Waktu\ VP - 12 * 2)$
2 - 30 hari	$166 + Waktu\ VP$
Lebih dari 4 minggu	$192 + Waktu\ VP$

Tabel 03: Validity Period

Pada contoh diatas, waktu VP-nya 5 hari, maka nilai VP adalah $166 + 5 = 171$ d = AB h. Jadi pada Validity Period hasilnya adalah AB.

7) User Data (UD)

User Data adalah isi pesan yang akan dikirim dalam format heksadesimal. Pada contoh ini isi pesan SMS yang akan dikirim adalah 'Pesan pendek'. Pengkodean dari nilai teks standar menjadi heksadesimal dilakukan dengan bantuan Default Alphabet yang dibakukan oleh ETSI (European Telecommunications Standards Institute) GSM 0.3.38 yang dapat dilihat pada tabel dibawah ini:

Nilai	Dec	Septet (7 bit)	Oktet (8 bit)	Hasil
P	80	1010000	11010000	D0
E	101	1100101	11110010	F2
s	115	1110011	00111100	3C
a	97	1100001	11101100	EC
n	110	1101110	00000110	06
spasi	32	0100000	11000001	C1
p	112	1110000	11001011	CB
e	101	1100101		
N	110	1101110	01101110	6E
D	100	1100100	01110010	72
E	101	1100101	01111001	79
K	107	1101011	00001101	0D

Tabel 04: User Data

Dari table terlihat bahwa hasil dari pengkodean adalah D0F23CEC06C1CB6E72790D. Jadi pada User Data hasilnya adalah D0F23CEC06C1CB6E72790D.

Dari penjelasan di atas diperoleh hasil untuk pengiriman SMS dalam format PDU untuk contoh tersebut adalah:

0011000000C912618229888040000AB0
CD0F23CEC06C1CB6E72790D

SMS PDU Penerima (Mobile Terminated) SMS PDU Penerima adalah terminal menerima pesan yang datang atau masuk dari SMSC ke handphone dalam format PDU. Pada prinsipnya pesan yang kita terima dari SMSC masih dalam format PDU setelah itu terminal handphone yang menerima pesan akan melakukan pengkodean menjadi teks, proses ini sering disebut decode. Cara pengkodean format PDU sudah diatur dan

disandarkan oleh ETSI. Format PDU dari SMS penerima adalah:

SCA	PDU Type	OA	PID	DCS	SCT S	UDL	UD
-----	----------	----	-----	-----	-------	-----	----

Tabel 05:Skema Format SMS PD Penerima

Contoh: Kita menerima pesan dari 628122888374 dengan isi pesan SMS adalah "helohelo" pada tanggal 6 Januari 2004 pukul 16.22 WIB. Maka format PDU adalah:

06912618010000040C91261822883847
0000401060612202820AE8329BFD4697
D9EC37

8) Service Centyer Address (SCA)

SCA adalah alamat (nomor) dari SMSC. SCA memiliki tiga komponen utama, yaitu len, type of number, dan service center number. Pada contoh di atas nilai dari SCA adalah 06912618010000.³

³ Gwenael Le Bodic, *op. cit.*, h. 131.

Oktet	Hasil	Keterangan
Len	06	Panjang informasi SMSC dalam oktet
Type of number	91	Format nomor dari SMSC 81 hexa = format lokal 91 hexa = format internasional
Service center number	261801000 0	Nomor SMSC dari operator pengirim. Jika panjangnya ganjil maka pada karakter terakhir ditambahkan 0F hexa Satelindo = 62816124 (PDU = 26181642) Telkomsel = 6281100000 (PDU = 2618010000) Excelkom = 62818445009 (PDU= 2618455400F9) IM3 = 62855000000 (PDU = 2658050000F0)

Tabel 06: PDU Type

Nilai default dari PDU Type untuk SMS-Deliver adalah 04 hexa, yang memiliki arti 04 hexa = 00000100.

Bit no	7	6	5	4	3	2	1	0
Nama	RP	UDHI	SRI	<nn>	<nn>	MMS	MTI	MTI
Nilai	0	0	0	1	0	0	0	1

Tabel 07: PDU Type

Keterangan:

RP : Reply Path. Parameter yang menunjukkan bahwa alur jawaban ada.

UDHI : User Data Header Indicator. Bit ini bernilai 1 jika data pengirim dimulai dengan suatu judul/tema.

SRI : Status Report Indication. Bit ini bernilai 1 jika status laporan akan dikembalikan ke SME.

MMS : More Message to Send. Bit ini bernilai 0 jika ada pesan lebih yang akan dikirim.

MTI : Message Type Indicator. Bit bernilai 0 untuk menunjukkan bahwa PDU ini adalah suatu SMS-Deliver.

9) Origanator Address (OA)

OA adalah alamat (nomor) dari pengirim, yang terdiri atas panjangnya nomor pengirim (Len), format dari nomor tujuan (Type Number), dan nomor pengirim (Origanator Number). Nilai OA pada contoh diatas adalah 0C91261822883847.

Oktet	Hasil	Keterangan
Len	06	Panjang nomor pengirim
Type of number	91	Format dari nomor pengirim 81 hexa = format lokal 91 hexa = format internasional
Origanator number	2618228838 47	Nomor pengirim dari operator pengirim. Jika panjangnya ganjil maka pada karakter terakhir ditambahkan 0F hexa

Tabel 08: Origanator Address

Protocol Identifier (PID)

Protocol Identifier adalah tipe atau format dari cara pengiriman pesan, yang biasanya diatur dari handphone pengirim. Misalnya tipe standard text, fax, E-mail dan lain-lainnya. Nilai deafult dari PID adalah 00 = "Standard Text". Pada contoh ini, pesan SMS yang akan dikirim menggunakan format teks standar, jadi

pada Protocol Identifier hasilnya adalah 00.

10) Data Coding Scheme (DCS)

Data Coding Scheme adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa SMS teks standar, Flash SMS, atau Blinking SMS. Pada contoh ini pesan SMS yang dikirim berupa teks standar, jadi pada Data Coding Scheme hasilnya adalah 00. Service Center Time Stamp adalah waktu dari penerimaan pesan oleh SMSC penerima. SCTS terdiri atas tahun, bulan, tanggal, jam, menit dan detik, serta zona waktu. Nilai SCTS pada contoh di atas adalah 40106061220282

Nama	Nilai	Hasil
Year	40	04(2004)
Month	10	05 (Mei)
Date	60	06
Hour	61	16
Minute	22	22
Second	02	20
Time Zone	82	28, di mana 1 unit = 15 menit. Jadi $(15 \times 28)/60 = 7$ jam. Sehingga menjadi GMT + 07.00 = WIB

Tabel 09: Service Center Time Stamp

11) User Data Length (UDL)

User Data Length adalah panjang dari pesan yang diterima dalam bentuk teks standar. Pada contoh ini pesan SMS yang diterima 0A, yang memiliki 10 karakter. User Data adalah isi pesan yang diterima dalam format heksadesimal.

Pada contoh ini nilainya adalah E8329BFD4697D9EC37. AT Command adalah perintah-perintah yang digunakan dalam komunikasi dengan Serial Port. Dengan AT Command kita dapat mengetahui vendor dari handphone yang digunakan, kekuatan sinyal, membaca pesan yang ada pada Sim Card, mengirim pesan, mendeteksi pesan SMS baru yang masuk secara otomatis, menghapus pesan pada SIM card, dan masih banyak lagi.

Dalam program SMS Server yang akan kita buat nanti, tidak semua perintah AT digunakan. Kita hanya menggunakan beberapa perintah AT yang ada hubungannya dengan sistem kerja dari program SMS Server. Adapun perintah yang akan digunakan adalah sebagai berikut;

AT Command	Keterangan
AT	Mengecek apakah handphone telah terhubung
AT+CMGF	Menetapkan format mode dari terminal
AT+CSCS	Menetapkan jenis encoding
AT+CNMI	Mendeteksi pesan SMS baru masuk secara otomatis
AT+CMGL	Membuka daftar SMS yang ada pada SIM Card
AT+CMGS	Mengirim pesan SMS
AT+CMGR	Membaca pesan SMS
AT+CMGD	Menghapus pesan SMS

Tabel 09: sistem kerja dari program SMS Server

Pembahasan

Komponen yang digunakan dalam pembuatan aplikasi ialah komputer yang memiliki port COM, handphone Siemens C55, dan kabel data Siemens RS232. Program SMS Server ini nanti akan digunakan dalam pembuatan aplikasi Sistem Informasi Akademik berbasis SMS. Adapun arsitektur sistem dari program utama SMS ini adalah sebagai berikut:

1. SMS dikirim oleh pengguna (client) ke nomor SMS Server.
2. SMS Server menerima pesan SMS yang masuk kemudian pesan itu diambil oleh program utama SMS Server.
3. Program utama SMS Server melakukan query ke database berdasarkan isi dari pesan SMS dari client.
4. Hasil query dari database kemudian dikirimkan ke program utama SMS Server.
5. Program utama SMS Server mengirim hasil database ke nomor client.

Proses-proses yang akan dibuat dalam kelas utama pada program SMS Server ini adalah sebagai berikut:

- 1) Metode main

- 2) Penyambungan database
- 3) Penyambungan dan pengaturan terminal
- 4) Membalik karakter pada format PDU
- 5) Mengubah nilai desimal menjadi heksadesimal
- 6) Mengubah dari 7 bit menjadi 8 bit
- 7) Mengubah dari 8 bit menjadi 8 bit
- 8) Mengubah format pesan SMS dari PDU menjadi teks
- 9) Mengubah format pesan SMS dari teks menjadi PDU
- 10) Mengirim At Command ke terminal
- 11) Menerima respons dari terminal
- 12) Memproses respons yang diterima dari terminal
- 13) Menerima pesan SMS yang masuk
- 14) Penyambungan Database

Proses selanjutnya dari program utama SMS Server adalah melakukan koneksi dengan database. Proses ini dilakukan agar SMS Server dapat melakukan query dengan database. Sumber data yang digunakan untuk SMS Server ini disimpan dalam database dengan menggunakan program Microsoft Access. Untuk itu dalam penyambungan dengan DBMS (Data Base management

system), kita menggunakan driver JDBC-ODBC Bridge.⁴

Kelas-kelas yang diimpor untuk koneksi database

```
import java.sql.SQLException;
import java.sql.Connection;
import java.sql.DriverManager;
metode set Database dibuat untuk
melakukan hubungan atau
penyambungan dengan database. Adapun
proses-proses yang dilakukan pada
metode ini adalah sebagai berikut:
```

1. Mencetak pesan kelayar bahwa server dimulai dan sedang membaca konfigurasi.
2. Lakukan proses untuk registrasi driver database dan penyambungan ke database dengan pengaturan URL, username, dan password.
3. Jika berhasil melakukan proses registrasi driver dan penyambungan database maka cetak proses keberhasilan ke layar dan aktifkan tombol tutup dan data serta nonaktifkan tombol mulai. Nilai statusServer dijadikan 1 yang menandakan bahwa server telah

berhasil melakukan penyambungan ke database.

4. Apabila proses penyambungan gagal yang disebabkan tidak ditemukan driver yang dibutuhkan maka cetak pesan kesalahan tersebut ke layar.
5. Apabila proses penyambungan gagal karena disebabkan tidak diteukan URL dari driver yang dibutuhkan maka cetak pesan kesalahan tersebut ke layar.

Java adalah Pemrograman berorientasi objek (suatu model pengembangan perangkat lunak yang sangat ini sangat populer). Sebagai mana bahasa pemrograman berorientasi objek, Java menggunakan kelas untuk membentuk suatu objek. Sejumlah kelas sudah tersedia dan Anda dapat menggunakannya dengan mudah, dan bahkan Anda dapat mengembangkannya lebih jauh melalui konsep pewarisan. Pewarisan adalah salah satu sifat yang ada pada bahasa pemrograman berorientasi objek, yang memungkinkan sifat-sifat suatu objek diturunkan dengan mudah ke objek lain.⁵ Selain itu Bahasa Java memungkinkan pengembangan membuat aplikasi Web yang dieksekusi

⁴ Abdul Kadir, Penuntun Praktis Belajar Database Menggunakan Microsoft Access. (ANDI Yogyakarta, 2004) h. 26

⁵ Abdul Kadir Dasar Pemrograman Java 2 (ANDI Yogyakarta 2005) h. 4

pada sisi klien, JSP menggunakan pendekatan pemrograman di sisi server.⁶

```

proses.add("Server Mulai.....",i);
proses.add("Server sedang membaca konfigurasi...",++i);
proses.select(i);
try{
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    koneksi=DriverManager.getConnection("jdbc:odbc:SMS","niken","sepeda");
    proses.add("Server telah terhubung ke Database ",++i);
    proses.add("Server sedang melakukan hubungan dengan terminal",++i);
        proses.select(i);
        mulai.setVisible(false);
        tutup.setVisible(true);
        data.setVisible(true);
        statusServer=1;
    }
    catch(ClassNotFoundException not){
        proses.add("Tidak menemukan Driver JDBC-ODBC",++i);
        proses.add("Proses Penyambungan Gagal",++i);

        proses.add("Penyambungan ke Database gagal",++i);
        proses.add("Kesalahan terjadi pada"+not,++i);
        proses.select(i);
        proses.add("",++i);
    }
    catch(SQLException sqln){
        proses.add("Tidak menemukan alamat URL",++i);
    }

```

```

        proses.add("Proses Penyambungan Gagal",++i);

        proses.add("Penyambungan ke Database gagal",++i);
        proses.add("Kesalahan terjadi pada"+not,++i);
        proses.select(i);
        proses.add("",++i); }

```

1) Penyambungan dan Pengaturan Terminal

Proses koneksi dengan terminal dilakukan agar handphone yang dijadikan sebagai penerima pesan SMS dapat berkomunikasi dengan program utama SMS Server. Metode setTerminal digunakan untuk koneksi dengan terminal serta melakukan pengaturan terhadap Serial Port dan terminal. Proses-proses yang dilakukan adalah;

1. Mencari dan mengambil daftar port yang terdapat di dalam komputer. Kemudian jika nama port yang dicari ditemukan, lakukan pembukaan terhadap port tersebut.
2. Untuk membaca data yang masuk dari terminal digunakan metode InputStream, sedangkan untuk menulis data ke terminal digunakan metode OutputStream
3. Setelah proses pembukaan terhadap Stream berhasil dilakukan maka

⁶ Abdul Kadir Dasar Pemrograman Web Dinamis dengan JSP (ANDI Yogyakarta 2005) h.5

dilakukan pengaturan parameter terminal yang terdiri atas bits per second, databits, stopbits, parity, dan flowcontrol.

4. Memberitahukan bahwa ada data di terminal, proses ini bertujuan agar SMS Server dapat menampilkan proses yang terjadi pada terminal.
5. Mencetak pesan ke layar jika SMS Server berhasil terhubung dengan terminal.
6. Setelah proses penyambungan ke terminal berhasil, lakukan pengiriman perintah AT Command untuk pengaturan terminal
7. Cetak pesan kesalahan jika penyambungan ke terminal gagal disebabkan oleh kesalahan memasukkan data Serial Port.

```
portName="COM1";
proses.add("server Sedang melakukan
pencarian Port",++i);
Enumeration
portList=CommPortIdentifier.getPortIdentifiers();
while(portList.hasMoreElements()){
CommPortIdentifier
portId=(commPortIdentifier)portList.nextElement();
    if(portId.getPortType() ==
commPortIdentifier.PORT_SERIAL){

        if(portId.getName().equals(portName)){
            try{
```

```
port=(SerialPort)portId.open("SMS",5000
);
        proses.add("Server berhasil
membuka port: "+portName,++i);
        proses.select(i);
        }
catch(PortInUseException pieu){
    proses.add("Port :
"+portname+"sedang digunakan ",++i);
    proses.add("penyambungan ke
terminal gagal ..... ,++i);
    proses.add("Terjadi kesalahan
pada :"+pieu,++i);
    proses.select(i);
        }
    }
}
    try{
output=port.getOutputStream();
input=port.getInputStream();
    }

catch(IOException ioe){
    proses.add("gagal membuka
Stream",++i);
    proses.add("Terjadi kesalahan
pada :"+ioe,++i);
    proses.select(i);
    }
try{
    port.setSerialportParams(19200,S
erialPort.DATABITS_8,SerialPort.STOPBIT
S_1,SerialPort.PARITY_NONE);
    port.setFlowControlmode(SerialPo
rt.FLOWCONTROL_NONE);
    port.notifyOnDataAvailable(true);
    proses.add("Server Melakukan
hubungan Ke Port: "+portName,++i);
    proses.add("Server Berhasil
Terhubung Ke Port: "+portName,++i);
    proses.add("Server Sedang
Melakukan Pengaturan terminal,++i);
    proses.add("Tunggu Sebentar .....
",++i);
    proses.select(i);
```

```

    kirimAT("AT+\15",1250);
    kirimAT("AT+CMGF=0"+"\15",125
0);
        kirimAT("AT+CSCSF=GSM"
        +"\15",12 50);
    kirimAT("AT+CNMI=1,1,2,2,1"+"\1
5",1250);
    kirimAT("AT+CMGL=0"+"\15",125
0);
        statusServer=2;
    }
    catch(UnsupportedCommOperationExcep
tion ucoe){
        proses.add("Pengaturan Data
Seraial port gagal",++i);
        proses.add("Kesalahan terjadi
pada : " +uceo,++i);
        proses.select(i);
    }
    try{
        port.addeventlistener(this);
    }
    catch(TooManylistenerException tmle){
        roses.add("Kesalahan terjadi pada
: " +uceo,++i);
        proses.select(i);
    }
}

```

2) Membalik Karakter pada Format PDU metode ini dibuat untuk melakukan modifikasi dengan cara membalik atau menukar posisi isi karakter yang diterima. Metode ini akan mengembalikan nilai berupa String yang dikembalikan kepada metode yang memanggil proses balikKarakter(). Nilai yang dikembalikan adalah nilai yang telah

dibalik atau ditukar posisi karakternya.

```

panjangKarakter=karakter.length();
stringBuffer=new
StringBuffer(panjangKarakter);
for (int i=0;(i+1)<panjangKarakter;
i=i+2){
    stringBuffer.append(Karakter.char
At(i+1));
    stringBuffer.append(karakter.char
At(i)); }
return new String(stringBuffer);
Mengubah Nilai dari Desimal Menjadi
Heksadesimal
char[] hexa
={ '0','1','2','3','4','5','6','7','8','9','A','B','C','D
','E','F'};
karakter =new char[2];
a=a&225;
karakter[0]=hexa[a/16];
karakter[1]=hexa[a%16];

return new String(karakter);

```

3) Mengubah dari 8 Bit Menjadi 7 Bit

Proses yang terjadi pada metode ini adalah mengubah pesan SMS dari nilai 8 bit menjadi 7 bit atau dengan kata lain mengubah pesan SMS yang berupa heksadesimal (GSM) menjadi karakter (ASCII). Pada proses ini akan terjadi pergeseran bit dari 8 bit menjadi 7 bit dan konversi dari nilai desimal menjadi kode ASCII. Metode ini akan mengembalikan nilai berupa String kepada metode yang memanggil proses delapanKeTujuhbit().Nilai yang

dikembalikan adalah nilai yang telah diubah menjadi format teks.

```
int i,o,r=0,rlen=0,olen=0,charcnt=0;
StringBuffer msg=new StringBuffer(160);
int pesanlen=pesan.length();
String ostr;
char c;

for
(i=0;(i+1)<pesanlen)&&(charcnt<msglen)
;i=i+2){
    ostr=pesan.substring(i,i+2);
    o=Integer.parseInt(ostr,16);
    olen=8;

    0<<=rlen;
    o|=r;
    olen+=rlen;
    c=(char)(0&127);
    0>>=7;
    olen-=7;
    r=0;
    rlen=olen;
    c=gsmToAsciiMap[c];
    msg.append(c);
    charcnt++;

    if(rlen>=7){
        c=(char)(r&127);
        r>>=7;
        rlen-=7;
        msg.append(c);
        charcnt++;
    }
}
if((rlen>0)&&(charcnt<msglen)){
    msg.append((char)r);
}
return msg.toString();
```

4) Mengirim AT Command ke Terminal
Metode kirimAT dibuat untuk mengirim AT Command yang akan

diterima ke terminal. Adapun proses-proses yang dilakukan pada metode ini adalah sebagai berikut:

1. Melakukan sinkronisasi atau antrean terhadap semua input AT Command yang akan dikirim ke terminal.
2. Mengirimkan AT Command ke terminal dengan cara menuliskan AT Command tersebut yang berupa byte ke terminal.
3. Melakukan penundaan waktu untuk mengirim input AT Command berikutnya. Lama waktu penundaan adalah sebesar waktu delay yang didapat dari setiap input AT Command yang akan dikirim.
4. Akhirnya berikan nilai get Delay menjadi true, agar metode ini dapat kembali mengirim input AT Command yang masuk

```
boolean tungguDelay=new boolean(true);
boolean getDelay=false;
synchronized(tungguDelay){
    try{
        output.write(atCmd.getBytes());
        output.flush();
    }
    catch(IOException e){}
    try{
        tungguDelay.walt(delay);
    }
    catch(InterruptedException
ie){
```

```

        getDelay=true;
    }
}

```

5) Menerima pesan SMS yang Masuk

Inti dari program SMS Server adalah pesan SMS yang diterima oleh server. Untuk itu dilakukan sebuah proses yang dapat mengambil data-data yang dibutuhkan untuk proses-proses selanjutnya. Data-data yang akan diambil dari setiap pesan SMS yang masuk adalah nomor telepon pengirim dan pesan SMS.

```

try{
    PDUterimaSMS(Pdu);
}
//Akhir try
catch (Exception e){}
if (dapatNotlp.endsWith("F")){
    dapatNotlp=dapatNotlp.substring(
0,dapatNotlp.length() - 1);
}
 kirimAT("AT+CMGD="+index+"\15",1250
);
 tulisDataTabelTerima(dapatNotlp,pesan);
 dari.setText(" "+dapatNotlp);
 isi.setText(" "+pesan);
 proses.add("Pesan dari nomor :
 "+dapatNotlp,++i);
 proses.add("Isi Pesan : "+pesan,++i);
 proses.select(i);

```

Kesimpulan

Berdasarkan hasil analisis yang telah dilakukan oleh penulis, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Short Message Service (SMS) merupakan salah satu layanan dari berbagai operator Global System for Mobile Communication (GSM) maupun Code Division Multiple Access (CDMA). Teknologi SMS memungkinkan kita mengirim pesan alphanumeric singkat dari sebuah Handphone ke Handphone yang lain. SMS Gateway adalah suatu platform yang menyediakan mekanisme untuk menghantar dan menerima SMS dari peralatan mobile.
2. SMS Gateway merupakan pintu gerbang bagi penyebaran Informasi dengan menggunakan SMS. Anda dapat menyebarkan pesan ke ratusan nomor secara otomatis dan cepat yang langsung terhubung dengan database nomor-nomor ponsel saja tanpa harus mengetik ratusan nomor dan pesan di ponsel karena semua nomor akan diambil secara otomatis dari database tersebut. Selain itu, dengan adanya SMS anda dapat mengustomisasi pesan-pesan yang ingin dikirim. Dengan menggunakan program tambahan yang dapat dibuat sendiri, pengirim pesan dapat lebih fleksibel dalam mengirim berita karena biasanya pesan yang ingin dikirim

berbeda-beda untuk masing-masing penerimanya.

3. Dengan menggunakan aplikasi SMS memberikan informasi kepada pengguna tentang proses yang terjadi dalam sistem informasi Short Message Service (SMS), pada seluler.
4. Pengguna dapat memahami cara kerja sistem Short Message Service (SMS).

Daftar Pustaka

- 1) Katankar, V.K., Thakare, V.M., Short Message Service using SMS Gateway , (IJCSE) International Journal on Computer Science and Engineering, Vol. 02, No. 04, 2010, 1487-1491
- 2) Gwenaël Le Bodic, Mobile Messaging Technologies and Services SMS, EMS and MMS. (England, John Wiley and Sons, Ltd., 2005)
- 3) Abdul Kadir, Penuntun Praktis Belajar Database Menggunakan Microsoft Access. (ANDI Yogyakarta, 2004).
- 4) Abdul Kadir Dasar Pemrograman Java 2 (ANDI Yogyakarta 2005)
- 5) Abdul Kadir Dasar Pemrograman Web Dinamis dengan JSP (ANDI Yogyakarta 2005)
- 6) Bjarne Stroustrup, The C++ Programming Language Third Edition (England, Addison-Wesley 1997).
- 7) Tandon, Sawal. Application on Incoming SMS to a Website, Control the Website to Send SMS Gateway. International Journal of Computer Science and Information Technologies (2012).
- 8) Rosidi, IR. Membuat Sendiri SMS Gateway (ESME) Berbasis Protokol SMPP". (ANDI Yogyakarta, 2007).