

SMS Gateway Dengan Bluetooth Menggunakan Mesin Unix Machintosh

Hamdani & Anindita Septiarini

Program Studi Ilmu Komputer, FMIPA Universitas Mulawarman
Jl. Barong Tongkok no.5 Kampus Unmul Gn. Kelua Sempaja Samarinda 75119

Abstrak

SMS Gateway merupakan media yang sangat efektif untuk menyampaikan pesan singkat melalui perangkat lunak computer ke ponsel. Perangkat keras untuk mengirim data menggunakan teknologi jaringan nirkabel yaitu teknologi bluetooth. Dalam proses pengiriman atau penerimaan pesan pendek (SMS), data dikirim maupun diterima menggunakan salah satu dari 2 mode yang ada, yaitu: mode teks, atau mode PDU (Protocol Data Unit), namun antara ponsel dengan operator masih menggunakan PDU. Dan metode AT Command (Hayes AT Command) digunakan untuk berkomunikasi dengan terminal (modem) melalui gerbang serial pada komputer. Dengan penggunaan perintah AT, dapat diketahui atau dibaca kondisi dari terminal, seperti mengetahui kondisi sinyal, kondisi baterai, mengirim pesan, membaca pesan, menambah item pada daftar telepon, dan sebagainya.

Kata Kunci: SMS gateway, Ponsel, AT Command, PDU, Bluetooth, Unix, Macintosh.

PENDAHULUAN

Penggunaan telepon seluler (ponsel) beberapa tahun ini sangat marak, terutama dalam penggunaan SMS (*Sort Message Service*) untuk mengirim pesan singkat di ponsel merupakan media yang paling banyak diminati oleh pengguna ponsel. Pada saat ini banyak perangkat teknologi untuk mengirim suatu pesan singkat ke beberapa media penerima pesan, antara lain yaitu personal computer (PC), ponsel dan pager. Umumnya metode yang sering digunakan adalah, komputer ke komputer atau ponsel ke ponsel. SMS gateway yang dimaksud disini adalah pengiriman pesan dari komputer ke ponsel atau sebaliknya. Seseorang dapat mengirim dan menerima pesan dari atau ke beberapa nomor dengan cepat dalam waktu yang singkat.

Dalam proses pengiriman atau penerimaan pesan pendek (SMS) menggunakan alat bantu komputer sebagai pemrosesan data, data yang dikirim maupun diterima menggunakan salah satu dari 2 mode yang ada, yaitu: mode teks, atau mode PDU (*Protocol Data Unit*), namun antara ponsel dengan operator masih menggunakan PDU. Semua informasi yang terdapat dalam PDU, dituliskan dalam bentuk pasangan-pasangan bilangan heksadesimal yang disebut dengan pasangan *oktet*. PDU yang diterima dari ponsel berbeda dengan PDU yang untuk dikirim, namun tetap menggunakan aturan yang sama. Perbedaannya terutama pada *field* yang dikandung, baik nilai maupun fungsinya.

Metode AT Command (*Hayes AT Command*) digunakan untuk berkomunikasi

dengan terminal (*modem*) melalui gerbang serial pada komputer. Dengan penggunaan perintah AT, dapat diketahui atau dibaca kondisi dari terminal, seperti mengetahui kondisi sinyal, kondisi baterai, mengirim pesan, membaca pesan, menambah item pada daftar telepon, dan sebagainya.

Sedangkan standard *bluetooth* dikembangkan untuk mengirim data melalui komputer atau telepon seluler hingga keyboard dan headphone dapat membuat koneksi sendiri, tanpa melalui kawat, kabel atau melalui terminal yang terhubung. Data akan dikirim melalui dengan model *wireless bluetooth* untuk menerima dan mengirim teks.

HASIL DAN PEMBAHASAN

Untuk aplikasi SMS ada beberapa komponen penting yang diperlukan, yaitu:

1. PC (*personal computer*) dan atau *notebook* dengan sistem operasi macintosh
2. SMS engine
3. Koneksi jaringan ke *provider* telekomunikasi
4. *Software* aplikasi
5. Teknologi ponsel memiliki bluetooth

Sistem kerja SMS

Dibalik tampilan menu *messages* pada ponsel sebenarnya adalah AT Command yg bertugas mengirim/menerima data ke/dari SMS-center.

AT Command mirip perintah > pada dos. Untuk mengetes AT Command dapat menggunakan windows HyperTerminal yang biasanya terdapat pada windows installer. Dengannya kita bisa membuka koneksi melalui port (terminal) dengan SMS gateway.

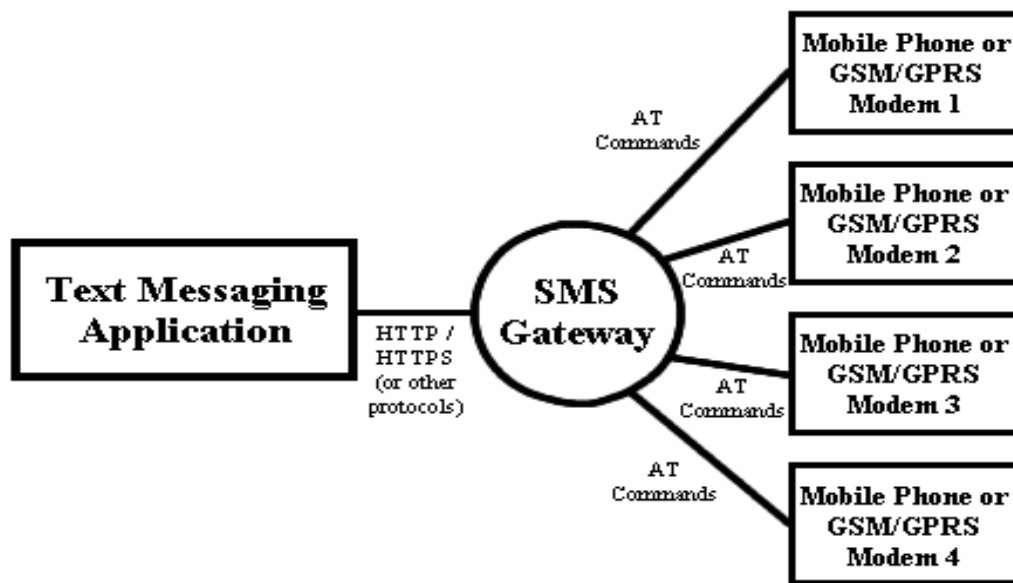
AT Command untuk SMS biasanya diikuti oleh data I/O (input dan output) yang diwakili oleh unit-unit PDU. Data yang mengalir ke/dari SMS-Center harus berbentuk PDU (Protocol Data Unit). PDU berisi bilangan-bilangan heksadesimal yang mencerminkan bahasa I/O. PDU sendiri terdiri dari beberapa

header.

Misalnya untuk mengirim SMS ada 7 header yang didalamnya terdapat lagi subheader-subheader. Yakni:

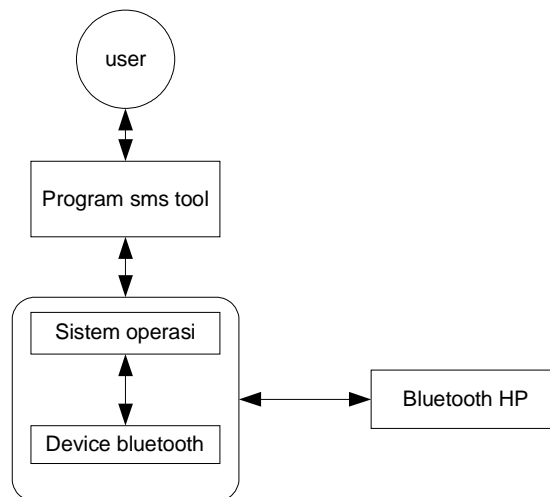
1. Nomor SMS-Center
2. Type SMS (misalnya send atau recieve)
3. Nomor Referensi SMS
4. Nomor ponsel penerima
5. Bentuk SMS (sebagai sms atau voiceSMS)
6. Skema encoding data I/O
7. Isi SMS

Berikut mekanisme pengiriman sms melalui dengan menggunakan SMS gateway :



Gambar 1. Skema SMS Gateway dalam menerima dan mengirim pesan

Dapat dilihat dari skema program secara umum dalam proses pengiriman dan penerimaan pesan melalui ponsel bluetooth ke sistem operasi komputer.



Gambar 2. Skema Program Secara Umum

Cara mengirim sms dari komputer ke ponsel

Pertama-tama yang harus dilakukan adalah menghubungkan HP ke komputer. Mengaktifkan jaringan *wireless bluetooth* baik pada komputer dan HP untuk dapat menerima dan mengirim pesan yang dikirim melalui SMS.

Setelah *bluetooth* terhubung, dapat mengendalikan HP dengan mengirim instruksi *shell* bawaan unix sistem operasi macintosh. Instruksi-instruksi ini disebut *AT commands*. Tabel berikut menampilkan beberapa *AT commands* untuk mengontrol pengiriman dan penerimaan SMS.

Tabel 1. *AT Command* mengirim dan menerima pesan

AT command	Arti
+CMGS	Mengirim pesan
+CMSS	Mengirim pesan dari <i>storage</i>
+CMGW	Menulis pesan ke memori
+CMGD	Menghapus pesan
+CMGC	Mengirim perintah
+CMMS	Mengirim pesan lagi

Skrip service smsd.conf untuk mengenali perangkat ponsel

```

devices = T610
logfile = /var/log/smsd.log
loglevel = 5
eventhandler = /home/sms/bin/smsevent

[SONYERICSSON T610]
device = CU.daniT610-serialport1-1
incoming = yes
baudrate = 19200
report = yes

[T610]
device = /dev/rfcomm0
incoming = yes
baudrate = 115200
report = yes
init = AT+CPMS="ME", "ME", "ME"
    
```

AT Command Set yang dapat dicoba saat

debugging:

- AT+CREG?: Status network. 0,1 berarti OK, 0,2 berarti searching, lainnya berarti error
- AT+CPMS="ME","ME","ME": Set pembacaan SMS ke ME (memori HP)
- AT+CPMS="SM","SM","SM": Set pembacaan SMS ke SM (memori SIM)
- AT+CSQ?: Kekuatan sinyal
- AT+CPMS?: ada sms tidak?

CU.daniT610-serialport1-1 pada **smsd.conf** merupakan *port bluetooth* yang tersambung ke T610.

Pengiriman SMS dengan mode PDU

Setiap pengiriman SMS, baik dari ponsel menuju operator, atau sebaliknya, selalu menggunakan format PDU (*Protocol Data Unit*), yaitu paket data dimana pesan SMS dikemas, bersama informasi tanggal, nomor tujuan, nomor pengirim, nomor operator, jenis skema SMS, masa valid SMS, dan beberapa hal lain (tergantung jenis paketnya).

Berikut ini contoh struktur PDU untuk dikirim :

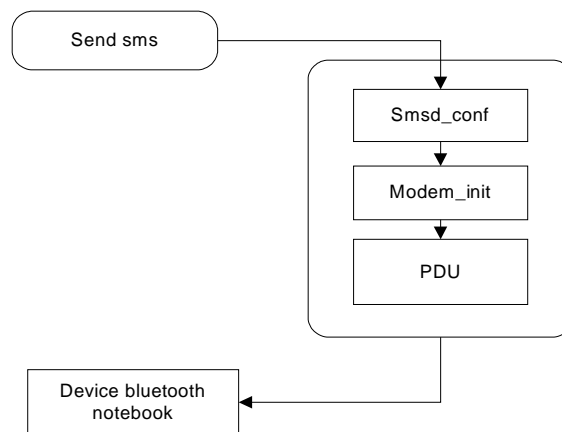
```
07 91 2658050000F0 11 00 0C 91 265836164900
00 00 FF 04 C830FB0D
```

Dengan keterangan sebagai berikut:

Tabel 2. Format PDU untuk mengirim pesan

Oktet / Digit Hexa	Keterangan
07	Panjang atau jumlah pasangan digit dari nomor SMSC (service number) yang digunakan, dalam hal ini adalah 7 pasangan (14 digit berikutnya)
91	Jenis nomor SMSC. Angka 91 menandakan format nomor internasional (misal +6281xxx). Untuk 081xxx menggunakan angka 81.
2658050000F0	Nomor SMSC yang digunakan. Karena jumlah digit nomor SMS adalah ganjil, maka digit paling belakang dipasangkan dengan huruf F. Kalau diterjemahkan, nomor SMSC yang digunakan adalah +6285500000 (IM3)
11	Oktet pertama untuk PDU SMS untuk dikirim (SMS SUBMIT).
00	TP-Message-Reference. Diisi "00" agar diisi otomatis oleh ponsel.
0C	Panjang digit dari nomor penerima (0C hex = 12 desimal)
91	Jenis nomor penerima (sama dengan jenis nomor SMSC)
265836164900	Nomor penerima SMS, yang jika diterjemahkan adalah +628563619400
00	Pengenal protokol, dalam hal ini adalah 0.
00	Skema pengkodean SMS, juga bernilai 0.
FF	Validitas waktu. FF berarti maksimum.
04	Panjang dari pesan SMS, dalam hal ini adalah 4 huruf (dalam mode 7 bit).
C830FB0D	Pesan SMS dalam mode 7 bit. Jika diterjemahkan kedalam 8 bit, lalu dirubah ke ASCII, maka didapat pesan 'Halo'

Perhatikan mekanisme proses pengiriman pesan dibawah.



Gambar 3. Mekanisme Pengiriman SMS

Perintah AT (AT COMMAND)

Untuk ponsel tertentu, dapat menghilangkan pemberian nomor SMSC pada PDU, kemudian akan diisi oleh ponsel sesuai dengan kartu yang sedang digunakan. Tapi untuk lebih amannya, dengan menyebutkan nomor SMSC yang digunakan.

Ketika PDU SMS selesai dibentuk, maka langkah selanjutnya adalah melakukan pengiriman PDU tersebut melalui port serial yang digunakan. Perintah yang pertama kali digunakan

adalah AT+CMGS, dengan aturan sebagai berikut:

AT+CMGS=<jumlah oktet PDU>

<jumlah oktet PDU> diisi dengan jumlah pasangan dalam PDU yang terbentuk, dengan dikurangi SMSC. Dengan kata lain, dapat menghitung jumlah oktet mulai dari oktet pertama SMS untuk dikirim (11). Untuk contoh di atas, berarti jumlah digit dalam PDU adalah 36, yang berarti terdiri dari 18 oktet. Sehingga perintah yang digunakan adalah:

AT+CMGS=18

Coding 7-bit data septets menjadi oktet

Berikut merupakan contoh dan langkah untuk mengkodekan text ke dalam *oktet*:

Text yang akan dikirim adalah "**hamdani**" terdiri dari 7 karakter, disebut sebagai *septets* ketika dinyatakan dalam 7 bit. Seluruh *septet* ini akan ditransformasikan ke dalam bentuk *oktet* untuk selanjutnya SMS dikirim.

Berikut langkah- langkah yang dinyatakan dalam tabel :

Tabel 3. Mengirim data berubah menjadi kode biner (*septet*)

Chr	M	A	K	H	M	u	d
ASCII (dec)	109	97	107	104	109	117	100
Bin(<i>septet</i>)	1101101	1100001	1101011	1101000	1101101	1110101	1100100
	1101101	110000 <u>1</u>	110101 <u>1</u>	110100 <u>0</u>	110110 <u>1</u>	111010 <u>1</u>	110010 <u>0</u>

Karakter pertama septet (**m**) diubah ke dalam *oktet* dengan menambahkan *rightmost bit* dari *septet* kedua. Bit ini disisipkan pada bagian kiri sehingga 1 + 1101101 = 11101101 ("ED"). *Rightmost bit* pada karakter ke dua dipakai,

sehingga karakter kedua (*septet*) membutuhkan dua buah bit (digaris bawah) dari karakter ke tiga untuk berubah menjadi 8 bit *oktet*. Proses berlanjut pada setiap karakter-karakter berikutnya:

Tabel 4. Mengirim biner (*oktet*) menjadi **hexadesial**

Bin(<i>oktet</i>)	<u>1</u> 1101101	<u>11</u> 110000	<u>000</u> 11010	<u>1101</u> 1101	<u>10101</u> 110	<u>100100</u> 11	1
Hex	ED	F0	1A	DD	AE	93	01

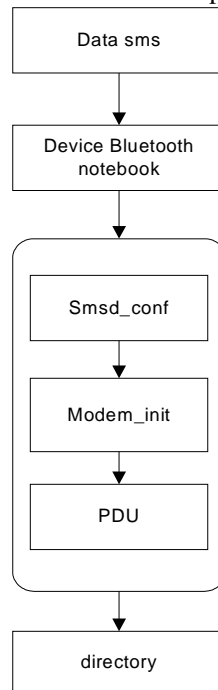
"makhmud" = EDF01ADDAE9301
 Penerimaan SMS dengan mode PDU
 Berikut adalah contoh PDU yang diterima oleh ponsel (*New SMS* atau *Inbox*) :

07 91 2658050000F0 04 0C 91 265836164900 00
 00 506020 31133180 04 C830FB0D
 Dengan keterangan sebagai berikut:

Tabel 5. Format PDU untuk menerima pesan

Oktet / Digit Hexa	Keterangan
07	Panjang atau jumlah pasangan digit dari nomor SMSC (service number) yang digunakan, dalam hal ini adalah 7 pasangan (14 digit berikutnya)
91	Jenis nomor SMSC. Angka 91 menandakan format nomor internasional (misal +6281xxx). Untuk 081xxx menggunakan angka 81.
2658050000F0	Nomor SMSC yang digunakan. Karena jumlah digit nomor SMS adalah ganjil, maka digit paling belakang dipasangkan dengan huruf F. Kalau diterjemahkan, nomor SMSC yang digunakan adalah +62855000000 (IM3)
04	Oktet pertama untuk pesan SMS yang diterima
0B	Panjang digit dari nomor pengirim (0C hex = 12 desimal)
91	Jenis nomor pengirim (sama dengan jenis nomor SMSC)
265836164900	Nomor pengirim SMS, yang jika diterjemahkan adalah +628563619400
00	Pengenal protokol, dalam hal ini adalah 0
00	Skema pengkodean SMS, juga bernilai 0
506020 311331 80	Waktu pengiriman, yang berarti 05-06-02 (2 Juni 2005), dan jam 13:31:13. Sedangkan 80 adalah Timezone yang digunakan.
04	Panjang dari pesan SMS, dalam hal ini adalah 4 huruf (dalam mode 7 bit).
C830FB0D	Pesan SMS dalam mode 7 bit. Jika diterjemahkan kedalam 8 bit, lalu dirubah ke ASCII, maka didapat pesan 'Halo'

Perhatikan gambar dibawah dalam proses PDU menerima data pesan dari komputer.



Gambar 4. Mekanisme Penerimaan SMS

Coding 7-bit data oktet menjadi septet

Seperti pada contoh di atas sms berisi data **hexadesial** C830FB0D. Langkah pertama yang dilakukan adalah mengubah pasangan-

pasangan *oktet* menjadi kode biner, sedemikian hingga dinyatakan dalam 8 bit. Jika jumlah bit kurang dari 8, kita dapat tambahkan bit "0" pada bagian kiri untuk melengkapi menjadi 8 bit.

Tabel 6. Menerima hexadesial mnejadi biner (*oktet*)

Hex	C8	30	FB	0D
Bin(<i>oktet</i>)	<u>1</u> 1001000	00 <u>1</u> 10000	<u>111</u> 11011	<u>0000</u> 1101

Dari tiap pasangan *oktet* diambil sejumlah bit untuk ditempatkan pada karakter selanjutnya. Pada karakter pertama diambil 1 bit pertama dari kiri, karakter kedua diambil dua

buah, dan seterusnya sehingga diperoleh bit data *septet*. Data *septet* kemudian diterjemahkan kedalam bentuk-bentuk karakter dari nilai desimalnya dengan kode ASCII.

Tabel 7. Menerima biner (*septet*) menjadi desimal (ASCII)

Bin(<i>septet</i>)	1001000	110000 <u>1</u>	11011 <u>00</u>	1101 <u>111</u>
Dec (ASCII)	72	97	108	111

Sehingga diperoleh text "**Halo**"

browser internet tanpa mengubah file teks aslinya.

Program-program yang ada pada SMS gateway ini diantaranya adalah:

Sebelum memulai proses kirim dan terima SMS, maka program *daemon smsd* harus dipanggil terlebih dulu karena program ini yang akan mengatur kirim dan terima SMS.

- **smsd**
Server yang berfungsi untuk mengirimkan dan menerima pesan SMS dari ke ponsel.
- **sendsms**
Sebagai antarmuka untuk mengirim pesan SMS dari pengguna.
- **sms2html**
Mengubah atau mengonversi file teks menjadi file html supaya dapat dibaca oleh

IMPLEMENTASI SISTEM

Mengirim SMS

Untuk mengirim SMS berarti membuat suatu file SMS. File tersebut sebenarnya merupakan file teks biasa yang berisi nomor tujuan dan isi SMS yang disimpan di folder **Outgoing**. **smsd** akan memeriksa folder setiap beberapa detik, jika file-nya ada maka akan segera dikirimkan ke nomor tujuan. Setelah SMS tersebut dikirim, maka file tersebut akan segera dihapus. Atau ketika sudah terhubung ke HP dengan baik, maka SMS tersebut akan langsung dikirim. Sintak menulis SMS di unix, adalah sebagai berikut:

```
# sendsms noTujuan isiSMS
```

Contoh:

```
# sendsms 628155145193 Testing
saja mas!
```

Atau jika tanpa *noTujuan* dan *isiSMS*, maka setelah perintah **sendsms**, akan menampilkan pesan sebagai berikut :

```
Destination: [diisi dengan
noTujuan]
```

```
Text : [diisi dengan
isiSMS yang akan dikirim]
```

Dan langsung tekan tombol Enter yang artinya kirim SMS.

Terima SMS

Ketika ada SMS yang masuk, maka pertama kali HP yang akan menerima SMS tersebut dan menyimpan SMS tersebut ke memori HP. **smsd** kemudian akan mengambil (*download*) SMS tersebut dari HP ke komputer di folder **Incoming**. SMS tersebut akan disimpan dalam bentuk file teks.

Untuk melihat isi SMS yang sudah masuk, digunakan sebuah perintah UNIX yaitu **cat**, yang berarti melihat atau mencetak isi suatu file, seperti contoh berikut ini:

```
# cat T610.abcde
```

T610.abcde merupakan file teks hasil *download* SMS yang diterima dari HP. Untuk mempermudah orang lain membaca file tersebut, maka file tersebut dapat dikonversi menjadi file html dengan **sms2html**, seperti contoh berikut ini:

```
# sms2html T610.abcde >
abcde.html
```

Artinya, file teks **T610.abcde** dikonversi menjadi file html **abcde.html**, supaya dapat dibaca oleh browser internet.



Gambar 5. iBook Mac OS X dengan T610 terkoneksi dengan *Bluetooth*

KESIMPULAN

Dari hasil uji coba program dan evaluasi penelitian, maka dapat di simpulkan sebagai berikut:

- Metode SMS *gateway* melalui bluetooth dapat digunakan mengirim dan menerima pesan singkat melalui beberapa operator telepon seluler.
- Teknologi model seperti ini dapat dikembangkan untuk beberapa media aplikasi seperti halnya SMS kampus

untuk mengetahui nilai yang terkoneksi kedalam database akademik.

- Auto SMS dalam pengembangan marketing di dunia usaha seperti promosi produk yang sedang dipasarkan.

DAFTAR PUSTAKA

Agrawal, Dharma Prakash., Zeng, Qin-An. 2003. *Introduction to Wireless and Mobile Systems*. Thomson Learning, Inc, Brook Cole.

Internet, tanggal 20 Juni 2007., *Tutorial SMS gateway*.

<http://brightside.wordpress.com/2006/01/20/tutorial-sms-gateway/>

Internet, tanggal 21 Juni 2007., *Alaska Communications Systems*.

<http://www.acsalaska.com/msib21/ClassApps/SendSMS/>

Internet, tanggal 21 Juni 2007., *Alltel Wireless*.
http://www.alltel.com/text_messaging/