

TEKNOLOGI SMS

Salah satu layanan yang disediakan ponsel dan sedang menjadi *trend* di masyarakat saat ini adalah SMS (*Short Message Service*) atau layanan pesan singkat. SMS merupakan teknologi yang memungkinkan untuk menerima dan mengirim pesan antar telepon bergerak (ponsel). Teknologi ini pertama kali diperkenalkan pada tahun 1992 di Eropa oleh ETSI (*European Telecommunications Standards Institute*) dan pada awalnya menjadi standar untuk telepon *wireless* yang berbasis GSM (*Global Systems for Mobile Communications*).

Sebagaimana namanya, SMS yang berarti layanan pesan pendek, maka besar data yang dapat ditampung oleh SMS ini sangatlah terbatas. Untuk satu SMS yang dikirimkan, hanya dapat menampung paling banyak sebesar 140 byte atau 1120 bit. Apabila diubah ke dalam bentuk karakter, maka untuk satu SMS hanya dapat berisi paling tidak 160 karakter untuk karakter latin, dan 70 karakter untuk karakter non-latin, seperti karakter Cina maupun Jepang.

SMS merupakan sebuah revolusi di mana layanan yang tidak berbasis suara malah lebih meledak, dalam industri komunikasi bergerak yang sesungguhnya berbasis suara. Keberhasilan dan popularitas SMS antara lain disebabkan oleh:

a. Pengguna dapat mengirimkan SMS ke nomor tujuan walaupun ponsel tujuan sedang tidak aktif.

Hal ini dikarenakan SMS memiliki masa tunggu. Jadi, selama masa tunggu SMS tersebut belum habis, SMS akan tetap terkirim ke nomor tujuan walaupun terlambat.

b. SMS adalah layanan yang pasti ada pada setiap ponsel.

Hal ini dikarenakan SMS merupakan sebuah standar yang pasti ada untuk setiap ponsel yang berbasis GSM.

c. Pengguna tidak dapat menolak SMS yang masuk ke ponsel.

Berbeda dengan panggilan langsung yang dapat ditolak apabila tidak ingin menerimanya. Sampai saat ini, belum ada cara khusus dalam ponsel maupun operator untuk menolak SMS yang telah dikirimkan.

d. Harganya yang relatif murah.

MEKANISME KERJA SMS

Ketika pengguna mengirimkan SMS ke suatu nomor, SMS tersebut tidak akan langsung dikirimkan ke nomor tujuan, namun akan masuk terlebih dahulu ke SMS *Center* (SMSC) operator telepon yang digunakan oleh pengguna. SMS *Center* sendiri dapat diartikan sebagai sebuah server yang bertanggung jawab pada proses pengiriman SMS dalam suatu operator. SMS yang dikirimkan dari suatu ponsel

akan masuk ke SMSC ini, kemudian baru diteruskan ke nomor tujuan SMS tersebut. Apabila nomor yang dituju sedang mati (*offline*), SMSC ini akan menyimpan SMS tersebut untuk sementara waktu hingga nomor tujuan hidup kembali. Lamanya waktu penyimpanan SMS, sangat tergantung dari lamanya waktu yang telah ditetapkan oleh operator untuk menyimpan SMS tersebut.

Nomor yang telah menerima SMS akan mengirimkan laporan ke SMSC bahwa SMS telah diterima. Laporan tersebut kemudian akan diteruskan kembali ke nomor pengirim SMS.

SMS GATEWAY

Istilah *gateway* dapat diartikan sebagai pintu gerbang. Namun pada dunia komputer, *gateway* dapat diartikan sebagai jembatan penghubung antara satu sistem dengan sistem yang lain, sehingga dapat terjadi pertukaran data antar sistem tersebut. Dengan demikian, SMS *gateway* dapat diartikan sebagai penghubung untuk lalu lintas data-data SMS.

Pada awalnya, SMS *gateway* dibutuhkan untuk menjembatani antar SMSC. Hal ini dikarenakan SMSC yang dibangun oleh perusahaan yang berbeda memiliki protokol komunikasi sendiri, dan protokol tersebut bersifat pribadi. SMS *gateway* ini kemudian ditempatkan di antara kedua SMSC yang berbeda protokol tersebut, yang akan menerjemahkan data dari protokol SMSC satu ke protokol SMSC lainnya yang dituju. Gambar 3.1 menunjukkan ilustrasi SMS *gateway*:



Gambar 3.1 Ilustrasi SMS *gateway*

Namun seiring perkembangan teknologi komputer dan perkembangan teknologi komunikasi, SMS *gateway* tidak lagi dimaksudkan sebagaimana yang tampak pada gambar 3.1 tersebut. Dewasa ini, masyarakat lebih mengartikan SMS *gateway* sebagai suatu jembatan komunikasi yang menghubungkan perangkat komunikasi (dalam hal ini ponsel) dengan perangkat komputer. Gambar 3.2 menunjukkan ilustrasi aplikasi SMS *gateway*:



Gambar 3.2 Ilustrasi aplikasi SMS *gateway*

SMS *gateway* kemudian lebih mengarah kepada sebuah program yang mengkomunikasikan sistem operasi komputer dengan perangkat komunikasi yang terpasang untuk mengirim atau menerima SMS.

PERANGKAT KOMUNIKASI

Perangkat komunikasi SMS *gateway* merupakan perangkat yang dapat digunakan untuk mengirim atau menerima SMS. Perangkat tersebut dapat berupa:

a. Telepon seluler (ponsel)

Agar dapat terhubung ke komputer, ponsel harus memiliki dukungan konektivitas ke perangkat lain, seperti kabel data, *bluetooth*, atau infra merah.

b. GSM modem

GSM modem merupakan sebuah modem *wireless* yang bekerja dengan GSM. Bila modem biasa menggunakan kabel telepon untuk transfer data, maka GSM modem menggunakan gelombang radio sebagai medianya. GSM modem yang digunakan dapat berupa PC *card*/PCMCIA *card* maupun berupa *device* eksternal yang menggunakan kabel serial atau USB untuk koneksi ke komputer. Sebagaimana namanya, GSM modem memerlukan SIM *card* GSM untuk mengoperasikannya.

c. GPRS modem

Perbedaan mendasar antara GSM modem dan GPRS modem adalah tambahan dukungan teknologi GPRS pada transmisi datanya. Kecepatan proses SMS pada GPRS modem lebih cepat dibandingkan dengan GSM modem.

MEDIA KONEKSI

Media koneksi digunakan untuk mengirimkan data dari perangkat komunikasi ke komputer atau sebaliknya. Media-media tersebut di antaranya adalah:

a. Kabel Data

Ada banyak jenis kabel data yang dapat digunakan untuk menghubungkan sebuah ponsel ke komputer, tergantung merk dan tipe ponsel. Berikut ini adalah beberapa jenis kabel data yang dapat dijumpai di pasaran:

- DAU9(P)

Kabel jenis ini merupakan jenis kabel yang tertua. Kabel data ini mendukung ponsel Nokia seri 6210 atau 7110 serta protokol FBUS.

- DLR3(P)

Bentuk mirip dengan kabel DAU9(P) dan telah mendukung mode AT. Ponsel yang dapat digunakan dengan kabel jenis ini adalah Nokia seri 6310, 6310i, 7110.

- DKU5

Pada kabel ini terdapat *chip* yang berfungsi mengemulasikan (emulator) port serial melalui konektor USB. Kabel ini digunakan untuk ponsel Nokia seri 2285, 3100, 3120, 3200, 3205, 3220, 3570, 3585, 3585i, 3586i, 3587i, 3588i, 3589i, 5100, 5140, 6015i, 6016i, 6019i, 6100, 6200, 6225, 6560, 6585, 6610, 6800, 6820, 7210, dan 7250i.

- DKU2

Bentuk kabel ini mirip dengan DKU5, namun tidak memiliki *chip* sebagaimana yang dimiliki oleh kabel DKU5. Ponsel yang didukung kabel DKU2 ini adalah Nokia seri 9500, 9300, 7710, 7700, 7610, 7600, 7270, 6682, 6670, 6651, 6650, 6620, 6255i, 6230, 6170, dan 3300.

- CA-42

Jenis kabel baru pengganti DKU5.

- R-DATA

Ada beberapa jenis kabel seperti ini. Untuk R-Data-3310, digunakan untuk ponsel Nokia seri 3330, 3390, 3395, 3410, dan 5550. Untuk R-Data-3510, digunakan untuk ponsel Nokia seri 3595, 3590, 3580, 3510.

- Kabel data untuk Motorola

- Kabel data untuk Sony Ericsson

Untuk keterangan lebih lengkap dapat diakses melalui situs <http://www.suncell-online.com/>.

b. Bluetooth

Bluetooth adalah sebuah standar spesifikasi yang mendefinisikan bagaimana antar perangkat (telepon, komputer, PDA atau perangkat digital lain) dapat saling berhubungan dan berkomunikasi tanpa melalui kabel (*wireless*). Media yang dipakai dalam transmisi datanya adalah gelombang radio frekuensi 2,4 GHz. Jarak maksimal yang mungkin untuk komunikasi antar perangkat ini adalah 100 meter. Agar sebuah komputer dapat berkomunikasi dengan perangkat lain yang memiliki *bluetooth*, diperlukan sebuah *bluetooth adapter*.

c. IrDA

IrDA (*Infrared Data Association*) adalah sebuah organisasi nonprofit yang dibentuk untuk membuat sebuah standar transmisi data tanpa kabel dengan sinar infra merah sebagai medianya. Untuk selanjutnya, penggunaan istilah IrDA adalah mengacu pada teknologi itu sendiri. Secara umum, jarak maksimal

antar perangkat IrDA agar dapat berkomunikasi adalah sepanjang 1 meter tanpa penghalang.

GAMMU

PENGENALAN GAMMU

Software yang akan digunakan untuk koneksi ponsel ke komputer dalam penelitian ini adalah Gammu (*GNU All Mobile Management Utilities*). Gammu merupakan *software* yang bersifat *open source* yang digunakan sebagai *tool* untuk mengembangkan aplikasi SMS *Gateway*, cukup mudah diimplementasikan, dan tidak berbayar. Kelebihan Gammu dari *tool* SMS *gateway* lainnya adalah:

- a. Gammu dapat dijalankan di sistem operasi Linux maupun Windows.
- b. Banyak *device* yang kompatibel di Gammu.
- c. Gammu menggunakan *database* MySQL untuk menyimpan SMS yang ada pada kotak masuk (*inbox*) maupun untuk mengirim pesan, sehingga dapat dibuat *interface* yang berbasis web maupun desktop.
- d. Baik kabel data USB maupun serial, semuanya kompatibel di Gammu.

Informasi mengenai ponsel apa saja yang didukung oleh Gammu, dapat dicari melalui situs <http://google.com> dengan kata kunci "gammu phone support".

FILE KONFIGURASI GAMMU

1. gammurc

File *gammurc* digunakan untuk konfigurasi port yang digunakan media koneksi untuk terhubung ke komputer. Selain itu, file *gammurc* juga digunakan untuk mendefinisikan tipe koneksi yang digunakan oleh media koneksi.

2. smsdrc

File *smsdrc* digunakan untuk konfigurasi *database* yang akan digunakan oleh aplikasi *gammu*. Nama *database*-nya adalah "smsd".

APLIKASI SMS GATEWAY

1. Install Gammu

Gammu yang digunakan adalah versi Windows yang dapat diperoleh di situs berikut http://dl.cihar.com/MIRROR/gammu/zips/gammu/gammu_win32.zip. Gammu tersebut merupakan versi 1.09. Namun, yang terbaru telah mencapai versi 1.19. Setelah Gammu for Windows tersebut didapatkan, maka dapat dilakukan langkah-langkah berikut:

a. Ekstrak Gammu

Ekstrak Gammu ke direktori C:\, maka akan muncul sebuah direktori bernama "win32" yang berisi semua file-file yang dibutuhkan oleh Gammu untuk dapat berfungsi dengan baik. Di antaranya adalah:

- gammu.exe
Untuk menjalankan *service* Gammu.
- gammurc
- smsdrc
- mysql.sql
Untuk membuat *database* "smsd" di MySQL.

b. Konfigurasi Database

Gammu menggunakan *database* MySQL dan untuk mempermudah konfigurasi, dapat digunakan *software* tambahan seperti XAMPP atau Apache2Triad sebagai *interface*-nya.

- Buat *database* "smsd" melalui XAMPP atau Apache2Triad dan import dari file mysql.sql.
- Setelah itu, terbentuk 9 tabel dan yang akan digunakan cukup 3 tabel saja, yaitu:
 - **Tabel inbox**
Tabel yang digunakan untuk menyimpan SMS yang masuk.
 - **Tabel outbox**
Tabel yang digunakan untuk mengirimkan SMS.
 - **Tabel sentitems**
Tabel yang digunakan untuk menyimpan SMS yang telah terkirim.
- Konfigurasi *field* yang ada pada ketiga tabel tersebut, yaitu *field* **Coding** dan **Default**. Ganti *field* **Coding** menjadi `enum('Default_No_Compression')` dan *field* **Default** menjadi `Default_No_Compression`. Hal ini dilakukan agar isi SMS yang masuk tidak berupa karakter kosong maupun karakter aneh.

c. Konfigurasi file gammurc dan smsdrc

Contoh file konfigurasi gammurc dan smsdrc apabila menggunakan ponsel Nokia 8210 dan kabel data serial:

```
[gammu]
port = com1:
#model = 6110
connection = fbus
#synchronizetime = yes
#logfile = gammulog
#logformat = textall
#use_locking = yes
#gammuloc = locfile
#startinfo = yes
#gammucoding = utf8
#rsslevel = teststable
#usephonedb = yes

[gammu1]
port = com1:
#model = 6110
connection = fbus
```

Gambar 7.1 File konfigurasi gammurc

Keterangan:

- **port**
Port yang digunakan ponsel untuk terhubung ke komputer melalui media koneksi (dalam hal ini kabel data). Untuk melihat port mana yang digunakan oleh ponsel, dapat dilakukan melalui Device Manager.
- **connection**
Tipe koneksi yang digunakan oleh media koneksi, dalam hal ini kabel data serial adalah "fbus". Untuk melihat macam-macam media koneksi dan tipe koneksinya, dapat dilihat di file gammurc.

```
# ----- SETTINGS FOR --smsd MYSQL
user = root
password = 
pc = localhost
database = smsd
```

sesuaikan dengan password database MySQL

isikan smsd

Gambar 7.2 File konfigurasi smsdrc

Keterangan:

- **user**
Nama user yang mengoperasikan database MySQL.
- **password**

Password yang digunakan oleh *database* MySQL. Apabila tidak menggunakan password, maka tidak perlu diisi.

- **pc**
Komputer yang digunakan untuk install *database* MySQL.
- **database**
Nama *database* yang digunakan oleh Gammu.

d. Identifikasi perangkat komunikasi

Sebelum identifikasi dilakukan, pastikan perangkat komunikasi dan media koneksinya (kabel data, IrDA, atau *bluetooth*) sudah terpasang di komputer. Dan pastikan juga file **gammurc** dan **smsdrc** telah dikonfigurasi dengan baik sesuai perangkat komunikasi dan media koneksi yang digunakan. Kemudian, lakukan langkah-langkah berikut:

- Buka *command prompt*
- Pindah direktori ke C:\win32
- Ketikkan perintah berikut:

```
gammu --identify
```

Untuk identifikasi perangkat komunikasi (contoh: ponsel). Apabila telah muncul nomor IMEI (*International Mobile Equipment Identity*) yang merupakan sejumlah nomor unik yang biasanya terdapat pada telepon GSM ataupun UMTS, maka koneksi telah berhasil. Nomor IMEI pada ponsel dapat dilihat dengan cara mengetikkan **#06#*.

e. Menjalankan *service* Gammu

Setelah nomor IMEI terdeteksi, maka ketikkan perintah berikut (tetap di *command prompt*):

```
gammu --smsd MYSQL smsdrc
```

Bila tidak ada *error*, maka *service* Gammu telah berjalan dengan baik. Minimize window *command prompt* (jangan di-*close*). Lihat di dalam folder C:\win32, akan tercipta sebuah file baru bernama "smsdlog" yang merupakan log dari *service* Gammu. File ini akan bertambah terus ukurannya seiring dengan *service* Gammu yang terus dijalankan.

2. Menerima dan mengirim SMS

Untuk mempermudah proses penerimaan dan pengiriman SMS, maka dapat digunakan XAMPP atau Apache2Triad sebagai *interface*-nya.

a. Menerima SMS

Apabila *service* Gammu masih berjalan, SMS yang diterima tidak akan masuk ke inbox ponsel karena langsung dimasukkan ke tabel inbox di *database*. Nomor pengirim terdapat pada *field* **SenderNumber** dan isi SMS

terdapat pada *field* **TextDecoded**. Begitu juga jika sebelumnya inbox ponsel yang digunakan telah berisi SMS, maka SMS-SMS tersebut akan masuk ke tabel inbox di *database*, sehingga inbox ponsel kosong.

b. Mengirim SMS

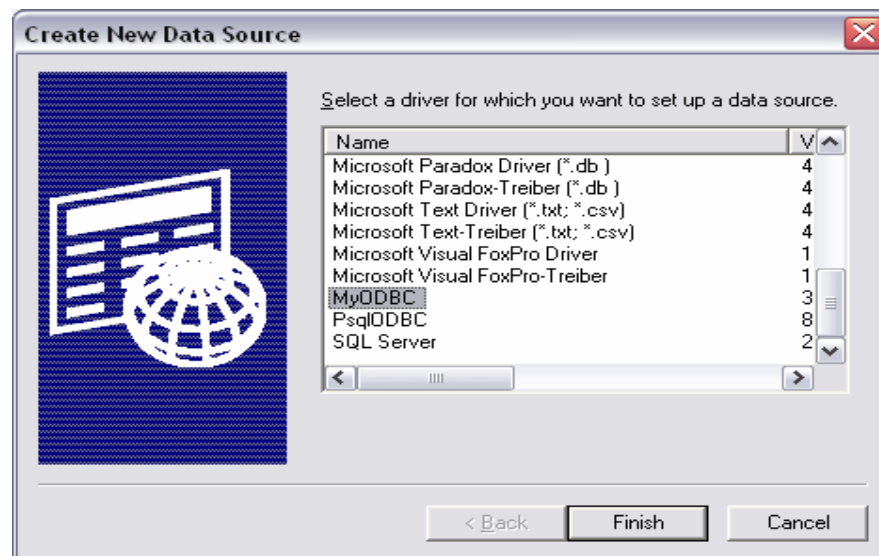
Apabila akan mengirim SMS, maka cukup isikan (*insert*) saja ke dalam tabel outbox berupa nomor tujuan pada *field* **DestinationNumber** dan isi SMS-nya pada *field* **TextDecoded**. SMS yang telah dikirimkan, akan dipindahkan ke tabel sentitems, sehingga setelah terkirim, tabel outbox akan selalu kosong.

3.3 Development Aplikasi

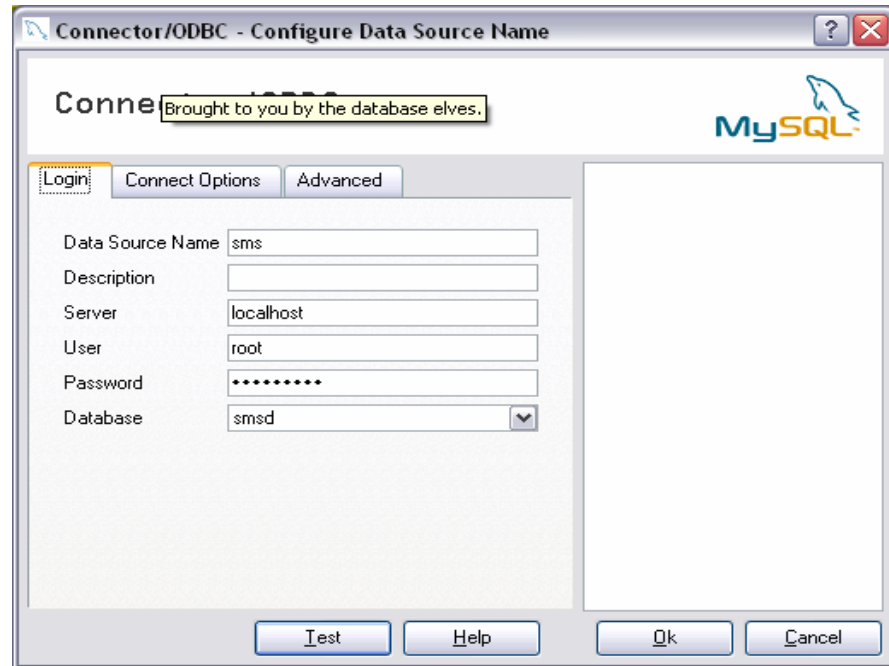
3.3.1 Konfigurasi Koneksi Database melalui ODBC

Sebelum memulai langkah development aplikasi maka lakukan konfigurasi koneksi database melalui ODBC. Software yang dibutuhkan adalah mysql-connector-odbc-3.51.15-win32. Langkah pertama yang harus dilakukan adalah penginstalan mysql-connector-odbc-3.51.15-win32, setelah proses penginstalan sukses, maka lakukan langkah-langkah konfigurasi ODBC berikut ini:

- Klik Start > Control Panel > Administrative Tools > Data Source (ODBC), kemudian muncul window ODBC data source administration, pada tab User DSN, tekan tombol Add. Muncul Window Create new data source berikut



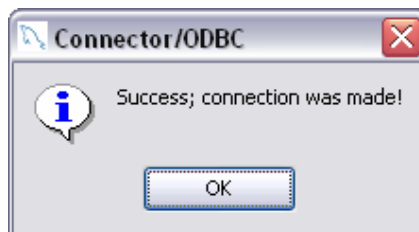
- Dari gambar di atas seleksi MyODBC, lalu klik finish. Muncul window Connector/ODBC – Add Data Source Name seperti gambar di bawah ini:



Kemudian lakukan konfigurasi seperti dibawah ini:

Data Source Name	sms
Description	-
Server	localhost
User	Root
Password	*Isi password sesuai dengan konfigurasi pada mysql server
Database	smsd

- Klik Test, jika koneksi berhasil dilakukan maka akan muncul tampilan berikut



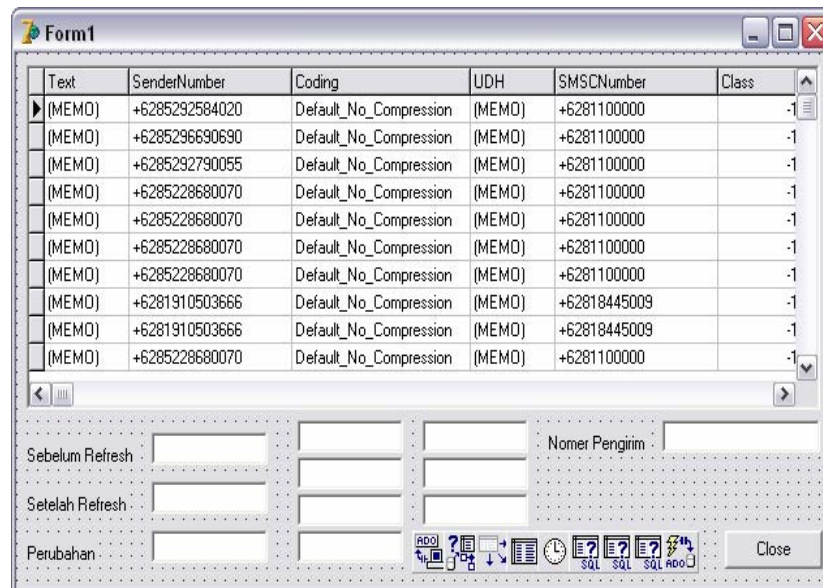
3.3.2 Langkah-langkah Development

Pada bagian ini, akan dibahas langkah-langkah development aplikasi SMS Gateway menggunakan bahasa pemrograman Visual Borland Delphi 7. Sedangkan untuk tool SMS Gateway yang digunakan adalah gammu. Dan mysql server sebagai database servernya

Aplikasi yang akan dibangun nanti, dapat:

- Membaca SMS masuk
- Parsing kalimat SMS yang masuk
- Mengirim balasan SMS

a. Desain aplikasi



- Buat aplikasi baru dari File > New > Application
- Desainlah form seperti pada tampilan di atas yang berisi komponen-komponen sebagai berikut:

Komponen	Properties	Nilai Properties	Component Palete
DBGrid1	Name DataSource	DBGrid1 sms	DataControl
GroupBox1	Caption	Kontrol Record	Standart
GroupBox2	Caption	Hasil Parsing	Standart
Label1	Caption	Sebelum Refresh	Standart
Label2	Caption	Setelah Refresh	Standart

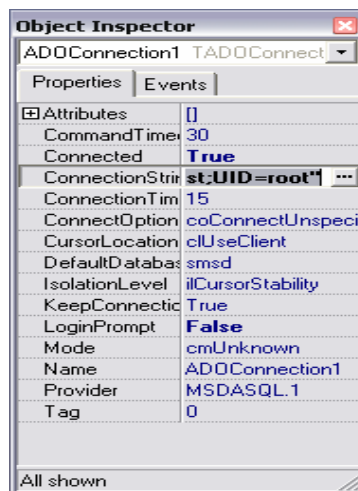
Label3	Caption	Perubahan Record	Standart
Edit1	Name Text	txtSebelum -	Standart
Edit2	Name Text	txtSesudah -	Standart
Edit3	Name Text	txtPerubahan -	Standart
Edit4	Name Text	txtKata1 -	Standart
Edit5	Name Text	txtKata2 -	Standart
Edit6	Name Text	txtKata3 -	Standart
Edit7	Name Text	txtKata4 -	Standart
Edit8	Name Text	txtKata5 -	Standart
Edit9	Name Text	txtKata6 -	Standart
Edit10	Name Text	txtKata7 -	Standart
Edit11	Name Text	txtNo -	Standart
ADODConnection1	Name ConnectionString Connected	ADODConnection1 *Lihat keterangan true	ADO
ADODCommand1	Name Connection	ADODCommand1 ADODConnection1	ADO
ADODCommand2			ADO
ADODDataSet	Name DataSource CommandText Active	ADODDataSet1 Sms *Keterangan true	
DataSource1	Name DataSet	Sms TableInbox	DataAccess
Tabel1	Name	TableInbox	BDE

	TableName Active AutoRefresh DatabaseName	Inbox True True sms	
Timer1	Name Interval	Timer1 10000	System
Query1	Name DatabaseName AutoRefresh	Query1 Sms True	BDE
Query2	Name DatabaseName AutoRefresh	Query1 Sms True	BDE
Query3	Name DatabaseName AutoRefresh	Query1 Sms True	BDE

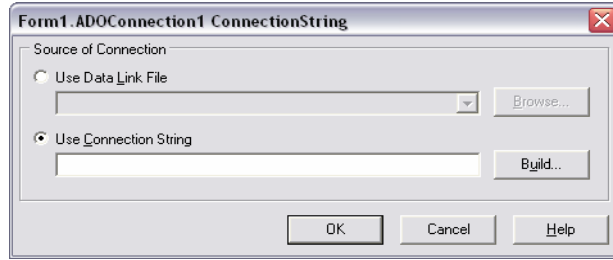
- Sebelumnya anda harus membuat koneksi ke mysql melalui ODBC agar aplikasi dapat mengakses ke database
- Setelah semuanya diatur melalui properti maka sekarang saatnya untuk mengisikan kode-kode program ke dalam desain aplikasi yang telah kita buat sebelumnya

Keterangan:

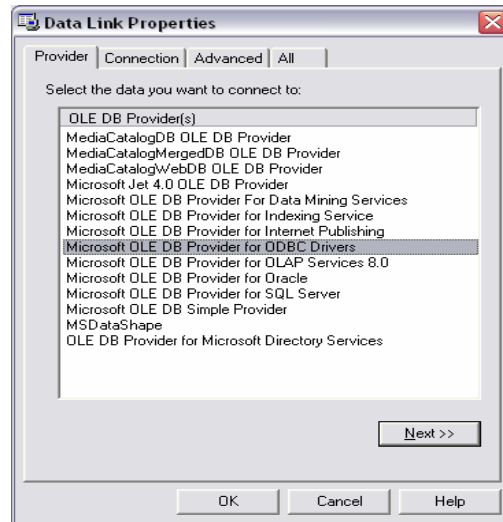
1. Setting properties komponen ADOConnection



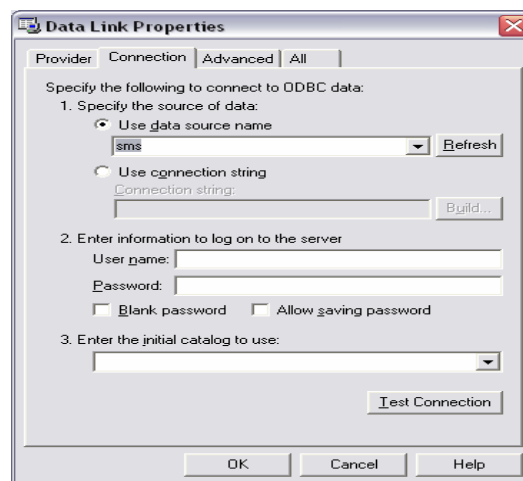
- Pilih tombol ... pada Connection String pada gambar di atas sehingga muncul tampilan sbb:



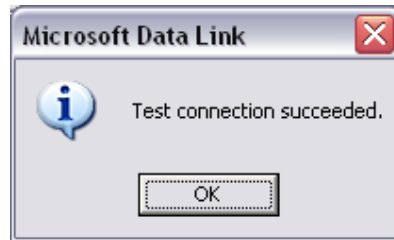
- Pilih Use Connection String lalu pilih tombol build



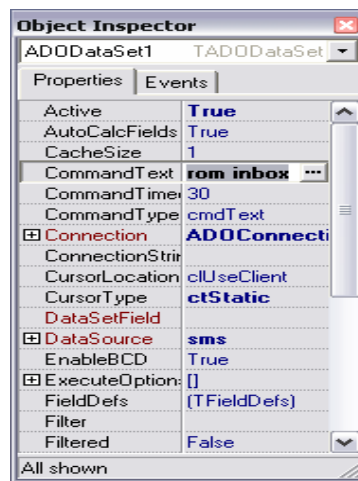
- Pada tab Provider pilih Micosoft OLE DB Provider for ODBC Driver lalu tekan next. Pada tab Connection klik Use data source name lalu seleksi sms



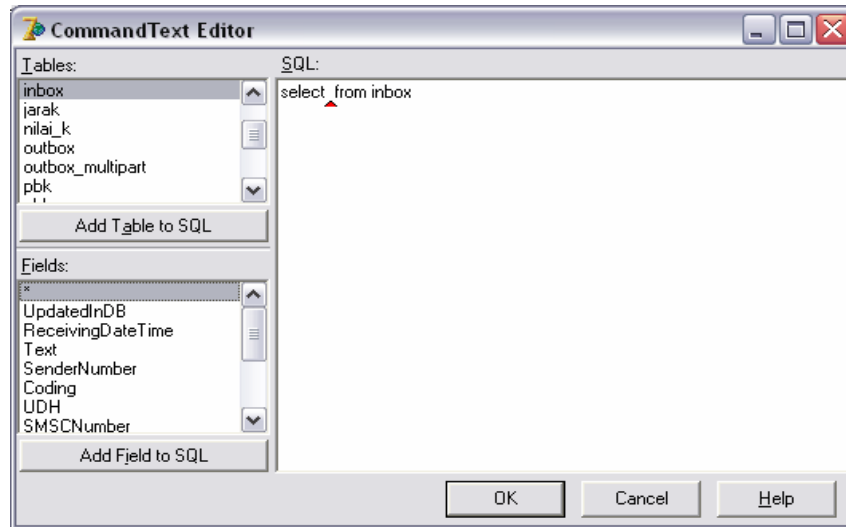
- Dari gambar di atas klik tombol Test Connection untuk menguji koneksi, jika koneksi sukses dilakukan maka akan muncul tampilan seperti di bawah ini:



2. Setting properties pada komponen ADODataset:



Dari gambar di atas, klik ... dari CommandText, kemudian muncul tampilan seperti di bawah ini. Kemudian lakukan seleksi tabel inbox pada Tables lalu tekan tombol Add table to SQL. Seleksi * pada Field yang menandakan kita akan menyeleksi semua field dari tabel inbox lalu tekan tombol Add field to SQL



b. Kode Program

Setelah desain dan pengesetan properti dari masing-masing komponen pada desain yang telah dibuat, maka langkah berikutnya adalah mengetikkan kode-kode program seperti dibawah ini:

Isikan kode program di bawah ini, dengan cara doble klik pada form:

```
procedure TForm1.FormShow(Sender: TObject);
begin
    txtSebelum.Text := IntToStr(TableInbox.RecordCount);
    txtSetelah.Text := '0';
    txtPerubahan.Text := '0';
    Timer1.Enabled := True;
end;
```

Fungsi ini digunakan untuk melakukan proses split(pemecahan) kalimat (isi dari sms yang masuk) menjadi kata, proses pemecahan kalimat isi dari sms yang masuk ditandai oleh spasi

```
procedure Split(const pembatas: char; // karakter pembatas
                input: string; // input isi SMS inbox
                const hasil: TStrings); // hasil parsing
begin
    Assert(Assigned(hasil));
    hasil.Clear;
    hasil.Delimiter := pembatas;
    hasil.DelimitedText := input;
end;
```

Dobel klik Timer1, kemudian ketikkan kode program dibawah ini:

```
procedure TForm1.Timer1Timer(Sender: TObject);
var
coba,cobal:String;
a : TStringList;
  reg, jk, tb, bb, lpe, lpa, lemak : string;
begin
  DBGrid1.Update;
  txtSetelah.Text := InttoStr(TableInbox.RecordCount);
  if (StrToInt(txtSebelum.Text) < StrToInt(txtSetelah.Text)) then
    begin
      a := TStringList.Create;
      tableinbox.Close;
      tableinbox.Open;
      tableinbox.Last;
      coba:=TableInbox.FieldName('SenderNumber').AsString;
      cobal:=TableInbox.FieldName('TextDecoded').AsString;
      try
        Split(' ', cobal, a);
        reg := (a[0]);
        jk := (a[1]);
        tb := (a[2]);
        bb := (a[3]);
        lpe := (a[4]);
        lpa := (a[5]);
        lemak := (a[6]);
        finally
          a.Free;
        end;
      txtNo.Text:=coba;
      txtKata1.Text:=reg;
      txtKata2.Text:=jk;
      txtKata3.Text:=tb;
      txtKata4.Text:=bb;
      txtKata5.Text:=lpe;
      txtKata6.Text:=lpa;
      txtKata7.Text:=lemak;
      if (jk='L') then
```

```

begin
    Query2.SQL.Clear;
    Query2.SQL.Add('INSERT INTO
sampel_data_l(tinggi_badan,berat_badan,lingkar_perut,' +
        'lingkar_panggul,lemak) values (''+ tb +',''+ bb +',''+ lpe
+',''+ lpa +',''+ lemak +'')');
    Query2.ExecSQL;
end
else if (jk='P') then
begin
    Query2.SQL.Clear;
    Query2.SQL.Add('INSERT INTO
sampel_data_p(tinggi_badan,berat_badan,lingkar_perut,' +
        'lingkar_panggul,lemak) values (''+ tb +',''+ bb +',''+ lpe
+',''+ lpa +',''+ lemak +'')');
    Query2.ExecSQL;
end;

    txtPerubahan.Text := InttoStr(StrtoInt(txtSetelah.Text) -
StrtoInt(txtSebelum.Text));
    Query1.SQL.Clear;
    Query1.SQL.Add(' INSERT INTO outbox(DestinationNumber,
TextDecoded) ' +
        ' values(''+coba+'', "nyobain SMS getwey nieh...")');
    Query1.ExecSQL;

    txtSebelum.Text := InttoStr(TableInbox.RecordCount);
    tableinbox.Next;
    txtNo.Text:=coba;
end else
    txtPerubahan.Text := 'Data tetap';

end;

```

Dobel klik cmdTutup (Tutup), kemudian ketikkan kode program di bawah ini:

```

procedure TForm1.Button1Click(Sender: TObject);
begin
    Application.Terminate;
end;

```

