

# Perancangan Aplikasi Pengecekan Nilai Ujian Mahasiswa melalui SMS

Hendra<sup>1</sup>.

## Intisari

*Pada era piranti bergerak telah menciptakan berbagai kemudahan bagi pemakainya dimana kemudahan mendapatkan informasi walaupun jauh dengan sumber informasi dengan syarat tersedianya jaringan komunikasi dari dan ke sumber informasi, salah satu masalah yang dihadapi mahasiswa adalah perlunya kehadiran dikampus untuk dapat mengetahui hasil ujian semester. Makalah ini menjelaskan perancangan suatu aplikasi perangkat lunak yang dapat digunakan untuk menerima dan mengirim SMS melalui komputer. Aplikasi ini dirancang untuk digunakan diperguruan tinggi sehingga para mahasiswa dapat melihat nilai hasil ujian semester melalui SMS dengan tanpa kehadiran di kampus.*

## 1. Pendahuluan

Melihat pengumuman nilai hasil ujian merupakan suatu event yang paling dinantikan mahasiswa begitu masa ujian semester berlalu, setiap mahasiswa rasanya tidak sabar untuk segera mengetahui nilai prestasi mereka dan rela melakukan perjalanan pulang pergi ke kampus. Beberapa perguruan tinggi telah menyediakan fasilitas pengumuman nilai ujian melalui web, tetapi masih banyak daerah yang belum dicapai oleh jaringan internet dibandingkan dengan jaringan GSM. Jika ditinjau dari segi perkembangan ICT dewasa ini yang sedemikian cepat dimana bertujuan untuk mengurangi pergerakan manusia untuk mendapatkan informasi, tentu saja kehadiran fisik dikampus untuk sekedar melihat nilai hasil ujian adalah tidak efisien. Apalagi dengan tingginya biaya transportasi dan membengkaknya biaya subsidi pemerintah dewasa ini yang sebagai akibat tingginya harga minyak dunia.

Dewasa ini kepemilikan telepon bergerak tentu saja bukanlah hal yang mewah lagi, dan diyakini bahwa setiap mahasiswa telah memiliki sarana komunikasi tersebut, dan jaringan GSM telah mencapai daerah pelosok.

## 2. Permasalahan

Berdasarkan uraian tersebut diatas, maka permasalahan pada penelitian ini adalah bagaimana hasil ujian mahasiswa dapat disampaikan ke telepon bergerak yang dimiliki oleh mahasiswa melalui komputer.

## 3. Tujuan

Penelitian ini bertujuan untuk mendapat suatu rancangan aplikasi yang mampu membaca data hasil ujian mahasiswa dari sistim informasi akademik dan mengirimkan informasi tersebut dalam bentuk SMS.

## 4. Asumsi

Penelitian ini menggunakan asumsi bahwa GSM modem yang digunakan mendukung AT Command dan SMS dalam format PDU, serta tersedianya fasilitas kabel data yang menghubungkan Komputer dengan modem.

---

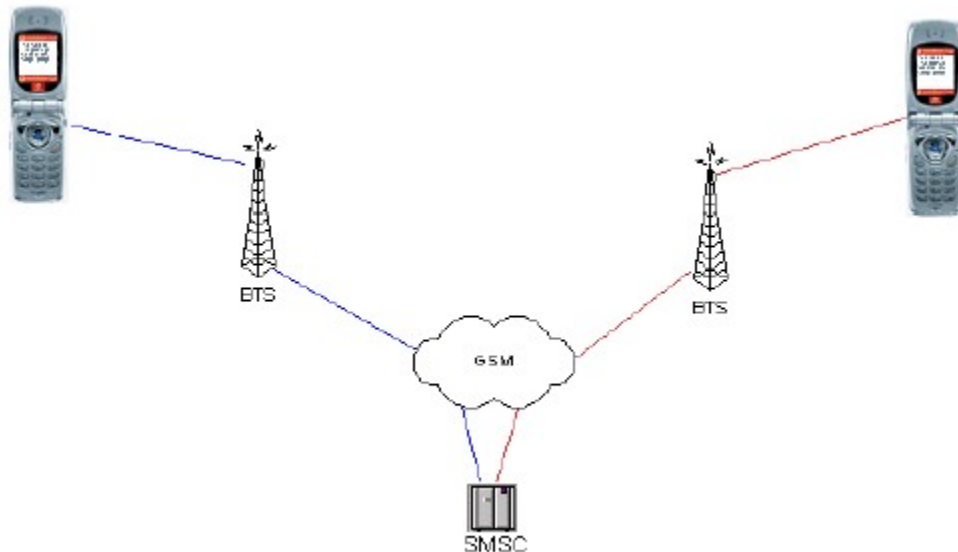
<sup>1</sup> Dosen tetap IBBI

## 5. Tinjauan Teoritis

### 5.1. SMS

SMS merupakan singkatan dari Short Message Service. Pesan SMS, sebagaimana yang dispesifikasikan oleh organisasi ETSI<sup>2</sup> (documents GSM 03.40 and GSM 03.38), dapat mencapai panjang 160 karakter, dimana masing-masing karakter adalah 7 bit.

Pesan singkat normalnya dimasukan melalui keypad pada handphone dan disimpan kedalam kartu SIM. Kemudian dikirim melalui kanal sinyal ke SMSC<sup>3</sup> seperti yang ditunjukkan pada Gambar 1.



Gambar 1. Proses pengiriman SMS melalui SMSC  
(Sumber: Stefen Fringe, 2005)

SMSC menyimpan pesan tersebut (max 2 hari) dan mengirimkannya melalui kanal sinyal ke penerima. Jika pengirim meminta status report, maka SMSC akan mengirim balik suatu status report tentang status pesan tersebut.

### 5.2. Perintah Modem GSM

Untuk aplikasi SMS anda perlu membuat koneksi ke SMSC dengan Modem GSM yang dapat mengirim dan menerima pesan singkat melalui kanal sinyal yang tersedia ketika modem sedang tercatat pada network operator.

Semua perintah GSM Modem sebagaimana yang dispesifikasikan pada ETSI GSM 07.07 [1] and GSM 07.05 [2] dimulai dengan AT atau dikenal dengan istilah AT-Command, dan setelah memberikan perintah anda perlu mengakhirinya dengan enter "<CR>" (= 0x0D). Beberapa perintah dapat diketik pada satu baris dengan suatu semicolon (;) sebagai pemisah antar perintah. Hasil input akan direpson oleh "OK" atau "ERROR". Perintah yang sedang diproses dapat terhenti oleh karakter yang dimasukan berikutnya, hal ini berarti anda tidak diperbolehkan memasukkan perintah selanjutnya sampai dengan anda menerima respon tersebut.

<sup>2</sup> European Telecommunications Standards Institute (**ETSI**)

<sup>3</sup> SMSC is short for "Short Message Service Center," and is the machine(s) within a wireless service provider's network that provides the routing of all SMS or text messages. Much like an email server, the SMSC handles large volumes of messages sent between two mobile phones or a mobile phone and a software application.

### 5.2.1. Mengirim dan menerima SMS dengan PC

Untuk dapat mengirim SMS melalui komputer, anda membutuhkan handphone yang memiliki fasilitas build-in GSM modem yang dapat dioperasikan melalui suatu antarmuka serial (data cable ataupun koneksi infra merah).

Terdapat dua cara untuk mengirim pesan SMS: dengan modus text dan modus PDU (protocol description unit). Modus text (tidak tersedia pada beberapa handphone).

### 5.2.2. Mengirim dan menerima SMS dengan PC

Berikut ini adalah beberapa perintah AT yang dapat digunakan untuk mendeteksi kesiapan GSM modem<sup>4</sup> seperti ditunjukkan pada Tabel 1.

AT	Mengembalikan "OK" sebagai konfirmasi modem berfungsi.
AT+CPIN="xxxx"	Memasukan PIN untuk SIM ( jika diaktifkan )
AT+CREG?	Mengembalikan "0,1" sebagai konfirmasi modem terkoneksi ke jaringan GSM.
AT+CSQ	Mendapatkan info kualitas sinyal (berkisar antara 2 s/d maksimum 31.99 untuk memungkinkan pengiriman dan penerimaan sms dari SMSC)

Tabel 1. Perintah AT untuk inisialiasi GSM Modem

## 5.3. Format PDU

Ada dua cara untuk mengirim dan menerima pesan SMS; yaitu dengan modus text dan dengan modus PDU (protocol description unit). Modus text (tidak tersedia untuk beberapa handphone) hanya berupa hasil encoding dari urutan bit yang dinyatakan oleh modus PDU.

### 5.3.1. Menerima pesan SMS dengan Modus PDU

String PDU bukan hanya terdiri dari pesan saja, tetapi juga mengandung sejumlah informasi mengenai pengirim, SMSC yang digunakan, stempel waktu, dll. Semuanya dalam bentuk hexadecimal octet atau decimal *semi-octets*. String berikut diterima dari handphone Nokia 6610 ketika menerima pesan "Indoprogram".

```
AT+CMGF=0
OK
AT+CMGR= 1
+CMGR: 1,,26
0791261203797694240C9126160380028200006070107103631C084937F90D97BFCF
OK
```

**07 91261203797694** 240C9126160380028200006070107103631C084937F90D97BFCF

Barisan octet ini terdiri dari tiga bagian : octet inisial yang menyatakan panjang dari SMSC ("07"), SMSC dalam hal ini adalah ("91261203797694"), dan bagian SMS\_DELIVER (ditentukan oleh ETSI pada GSM 03.40).

---

<sup>4</sup> A GSM modem can be an external modem device, such as the Wavecom FASTRACK Modem. Insert a GSM SIM card into this modem, and connect the modem to an available serial port on your computer. A GSM modem can be a PC Card installed in a notebook computer, such as the Nokia Card Phone. A GSM modem could also be a standard GSM mobile phone with the appropriate cable and software driver to connect to a serial port on your computer. Phones such as the Nokia 7110 with a DLR-3 cable, or various Ericsson phones, are often used for this purpose.

Octet	Keterangan
07	Panjang dari informasi SMSC (dalam hal ini 7 octet)
91	Jenis alamat SMSC. (91 menyatakan format internasional dari nomor telepon)
261203797694	Nomor service center (dalam decimal semi-octet). Panjang dari nomor telepon adalah ganjil (11), sehingga digenapi dengan F untuk membentuk octet yang sesialo. Nomor telepon SMSC dalam hal ini adalah "+27381000015"..
24	Octet pertama dari SMS-Delivery Octet.
0C	Panjang dari alamat pengirim (0B hex = 11 dec)
91	Jenis alamat dari nomor pengirim
261603800282	Nomor pengirim
00	TP-PID. Protocol identifier.
00	TP-DCS Data coding scheme
60 70 10 71 03 63 1C	TP-SCTS. Stempel waktu (semi-octets)
08	TP-UDL. Panjang data, ukuran panjang dari message. TP-DCS field menyatakan data 7-bit, jadi dalam hal ini adalah jumlah bilangan septets (10). Jika TP-DCS field diset untuk menyatakan 90bit data atau Unicode, ukuran panjang adalah bilangan octet (9).
4937F90D97BFCF	TP-UD. Pesan "Indoprog", 8 bit octet yang dinyatakan dalam 7-bit data.

Tabel 2. Bagian-bagian dari sms octet yang diterima.  
(Sumber: ETSI)

Semua octet diatas adalah hexa-desimal 8-bit, kecuali pada nomor SMSC, nomor pengirim dan stempel waktu; yang dinyatakan sebagai decimal semi-octet. Bagian dari message part pada akhir dari string PDU terdiri dari hexa-desimal 8-bit octet, tetapi octet ini menyatakan data 7-bit..

Semi octet adalah desimal, dengan contoh nomor pengirim dapat diperoleh dengan pengantian antar semi octet dari "261603800282" menjadi "626130082028". Pada bagian stempel waktu, adalah sama dengan "60 70 10 71 03 63 1C", dimana 6 karakter pertama mewakili tanggal, enam berikutnya adalah waktu, dan dua karakter akhir mewakili zona waktu dalam kaitan dengan GMT.

### 5.3.2. Mengirim pesan dalam modus PDU

Pada contoh berikut menunjukkan bagaimana mengirim pesan "Indoprog" pada modus PDU pada Nokia 6110.

```
AT+CMGF=0
OK
AT+CMGS=21
>0011000C912616038002820000AA084937F90D97BFCF<ctrl-z>
+CMGS: 210
OK
```

Ada 21 octet pada pesan ini (42 karakter). Octet pertama ("00") tidak dihitung, hanya berupa indicator dari panjang SMSC yang disediakan. String PDU terdiri dari format yang ditunjukkan pada Tabel 3.

Octet(s)	Description
00	Panjang dari informasi SMSC. Dalam hal ini 0 menyatakan bahwa SMSC yang tersimpan pada handphone akan digunakan. Catatan : Octet ini sifatnya optional.
11	Octet pertama dari
00	TP-Message-Reference. Nilai 00 dalam hal ini menyatakan referensi ke nomor telepon itu sendiri.
0C	Address-Length. Panjang dari nomor telepon (12)
91	<a href="#">Type-of-Address</a> . (91 menyatakan menggunakan format internasional untuk nomor telepon).
1603800282	Nomor telepon dalam semi octet (1603800282).
00	TP-PID. <a href="#">Protocol identifier</a>
00	TP-DCS. <a href="#">Data coding scheme</a> .
AA	<a href="#">TP-Validity-Period</a> . "AA" berarti 4 days
08	TP-User-Data-Length. Panjang dari pesan.
4937F90D97BFCF	TP-User-Data. Octet ini mewakili pesan "Indoprogram".

Tabel 3. Bagian-bagian dari sms octet untuk pengiriman sms.

### 5.3.3. Konversi septet ke octet.

Pesan "Indoprogram" terdiri dari 8 karakter, yang disebut sebagai septet ketiga direpresentasikan sebagai 7 bit. Septet ini perlu ditransformasi menjadi octet agar dapat ditransfer seperti yang ditunjukkan pada Tabel 4.

I	n	d	o	p	r	o	g
73	110	100	111	112	114	111	103
1001001	1101110	1100100	1101111	1110000	1110010	1101111	1101100
1101000	110111 0	11001 00	1101 111	111 0000	11 10010	1 101111	1101100

Tabel 4. Konversi septet menjadi octet.

Septet pertama diubah menjadi octet dengan menambah bit yang paling kanan dari septet kedua. Bit ini disisipkan pada sebelah kiri sehingga menjadi 0 + 1001001 = 01001001 ("E8"). Bit sebelah kanan untuk septet kedua telah digunakan, sehingga dibutuhkan dua bit dari septet berikutnya, dan hal ini dilakukan untuk kekurangan pada septet-septet berikutnya, sehingga akhirnya menghasilkan octet seperti yang ditunjukkan oleh Tabel 5.

0 1101000	00 110111	111 11001	0000 1101	10010 111	101111 11	1101100 0
49	37	F9	0D	97	BF	CF

Tabel 5. Konversi octet binary menjadi hexadesimal.

Sehingga octet dari "Indoprogram" adalah 49 37 F9 0D 97 BF CF

### 5.4. Komunikasi data serial

Setiap komputer tersedia satu atau lebih serial port, yang dinamai sebagai COM1, COM2, dan seterusnya untuk komunikasi data serial dengan perangkat eksternal. Serial port berfungsi sebagai interpreter antara CPU dengan perangkat serial, dimana data byte dari CPU dikonversi menjadi bit-bit, dan ketika data diterima dikonversi kembali menjadi byte.

### 5.4.1. Membuat koneksi serial

Langkah pertama dari komunikasi serial adalah membentuk koneksi ke serial port. dengan melakukan setting pada property-property seperti yang ditunjukkan pada Tabel 6.

Properties	Description
CommPort	Menentukan nomor port yang akan digunakan
Settings	Menentukan baud rate, parity, data bits, dan stop bit sebagai suatu string.
PortOpen	Membuka koneksi serial port

Tabel 6. Properti membuat koneksi serial

Contoh :

```
MSComm1.Settings = "9600,n,8,1"  
MSComm1.CommPort = 1  
MSComm1.PortOpen = True
```

### 5.4.2. Interaksi data dengan Modem

Sesaat setelah port serial terkoneksi, maka program dapat melakukan interaksi dengan Modem melalui property output, dan membaca data melalui property input.

Contoh :

```
MSComm1.Output = "AT+CMGF=0" & vbCr  
rec = MSComm1.Input
```

### 5.4.3. Handshaking

Suatu bagian untuk menangani proses penerimaan dan pengiriman data berhasil – dimana kecepatan pengiriman data tidak menyebabkan buffer overflow, diperlukan suatu protocol handshaking.

Anda dapat menentukan protocol handshaking yang digunakan, sebagai defaultnya adalah none (comNone) yang ditunjukkan pada Tabel 7.

Setting	Nilai	Keterangan
comNone	0	Tidak ada handshaking (Default).
comXOnXOff	1	XOn/XOff handshaking.
comRTS	2	RTS/CTS (Request To Send/Clear To Send) handshaking.
comRTSXOnXOff	3	Menggunakan Request To Send dan XOn/XOff handshaking.

Tabel 7. Properti protocol handshaking

Protokol yang anda gunakan sangat tergantung kepada peralatan yang digunakan, dan berikut ini ditunjukkan setting COM Port pada visual basic.

```
MSComm1.Settings = "9600,n,8,1"  
MSComm1.CommPort = xComPort  
MSComm1.Handshaking = comRTS  
MSComm1.PortOpen = True
```

## 6. Metodologi penelitian

Penelitian ini dilakukan dengan :

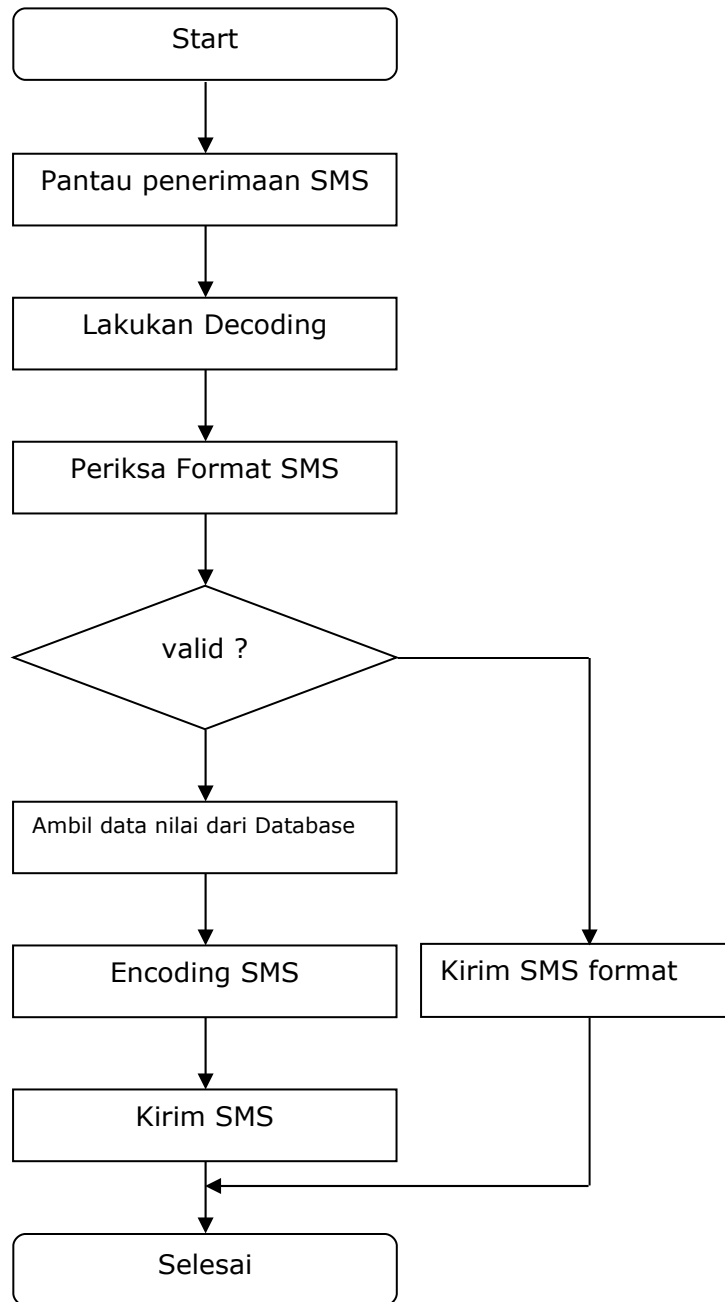
1. Melakukan studi pustaka dengan mengambil referensi dari beberapa situs yang relevan yang membahas tentang teknologi SMS, GSM Modem, AT-Command, format PDU, Komunikasi data melalui port COM.
2. Melakukan pengujian terhadap masing-masing komponen terkait dengan proses AT-Command, dan kodefikasi SMS text menjadi format PDU dan sebaliknya.
3. Merancang algoritma, struktur database dan user interface, membuat prototipe perangkat lunak dengan mengabungkan berbagai komponen tersebut diatas.
4. Melakukan pengujian terhadap prototipe perangkat lunak atas efektifitas hasil rancangan untuk suatu kesimpulan.

Untuk mendukung proses penelitian, penulis menggunakan menggunakan beberapa tools pendukung seperti Apache web server, PHP untuk penrograman sisi server, dan bahasa pemrograman pemrograman Visual Basic untuk proses kodefikasi PDU, dan komunikasi OCX MSCOMM.

## 7. Perancangan Aplikasi

### 7.1. Algoritma

Secara umum aplikasi akan beroperasi dengan memantau keberadaan SMS baru dengan menggunakan fungsi Time, jika ditemukan SMS baru, maka aplikasi akan mengirim perintah AT untuk pembacaan SMS dalam modus PDU, kemudian melakukan decoding untuk mendapatkan nomor telepon pengirim dan pesan SMS (berisi NIM). Kemudian aplikasi akan melakukan pengambilan daftar nilai prestasi dari table nilai berdasarkan NIM. Daftar diformat menjadi pesan SMS dan dikirim kembali ke nomor telepon pengirim. Adapun algoritma yang digunakan pada aplikasi yang dirancang oleh peneliti ditunjukkan pada Gambar 2.



Gambar 2. Flowchart logika program SMS nilai

## 7.2. Rancangan Format pesan SMS

Aplikasi SMS membutuhkan pemakai mengirim SMS ke nomor telepon SMS Gateway dan melakukan respon yang sesuai dengan isi SMS yang diterima.

Adapun format pesan SMS yang digunakan adalah sebagai berikut :

<NIM>[,SEMESTER][,KODE MATAKULIAH]



dimana :

<NIM>, merupakan nomor NIM mahasiswa dan sifatnya Mandatory

[,SEMESTER>, merupakan nomor SEMESTER yang ingin dilihat daftar nilainya dan sifatnya Optional.

[,KODE MATAKULIAH], merupakan kode matakuliah yang sifatnya Optional.

Contoh :

Pesan SMS :  
920403024

Akan menerima respon daftar nilai ujian semester yang sedang diikuti mahasiswa.

Pesan SMS  
920403024,4

Akan menerima respon daftar nilai semester 4 (empat) menurut jurusan.

Pesan SMS  
920403024,,MKI456

Akan menerima respon nilai untuk matakuliah MKI456 Simulasi Komputer

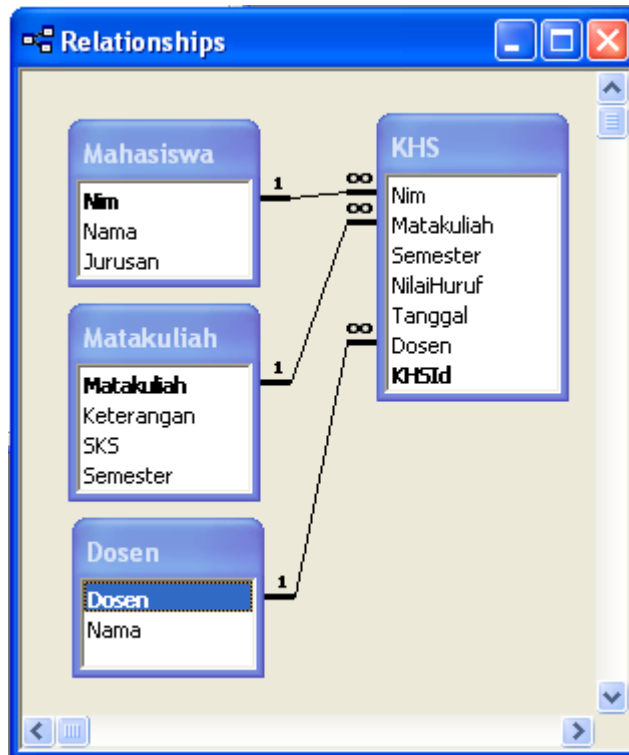
### **7.3. Pesan Kesalahan**

SMS yang dikirim pemakai tidak selamanya benar, sehingga perlu diberikan respon pemberitahuan jenis kesalahan sehingga dapat ditindaklanjuti oleh pemakai, berikut ini adalah pesan kesalahan :

001 Format SMS salah, ketik : <NIM>[,SEMESTER][MATAKULIAH], contoh ketik : 920403024 atau 920403024,4 atau 920403024,,MKI345

### **7.4. Rancangan Basis Data**

Sebagai sumber data sms, diperlukan suatu database untuk penyimpanan data akademik mahasiswa dengan skema yang ditunjukkan pada Gambar 3.



Gambar 3. Skema database akademik

### Struktur database akademik.

Tabel mahasiswa

Field	Tipe	Ukuran	Keterangan
Nim	Char	10	Primary key
Nama	Char	50	
Jurusan	Char	10	

Tabel matakuliah

Field	Tipe	Ukuran	Keterangan
Matakuliah	Char	5	Primary key
Keterangan	Char	50	
SKS	Integer		
Semester	Integer		

Tabel Dosen

Field	Tipe	Ukuran	Keterangan
Dosen	Char	10	Primary key
Nama	Char	50	

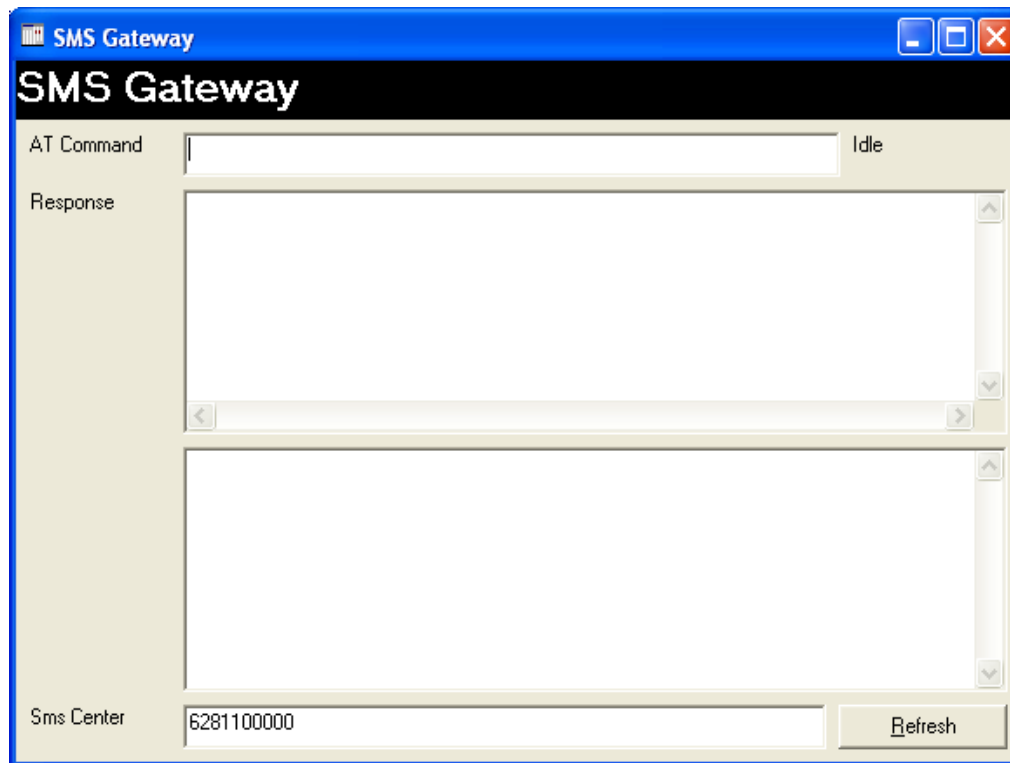
Tabel KHS

Field	Tipe	Ukuran	Keterangan
Nim	Char	10	Foreign key
Matakuliah	Char	5	Foreing key
Semester	Integer		Semester pada KHS
Nilai huruf	Char	1	
Tanggal	Date		Tanggal KHS
Dosen	Char	10	Dosen penguji
KHSid	Autonumber		Primary key

## 7.5. Rancangan antar muka

### 7.5.1. Sms Gateway

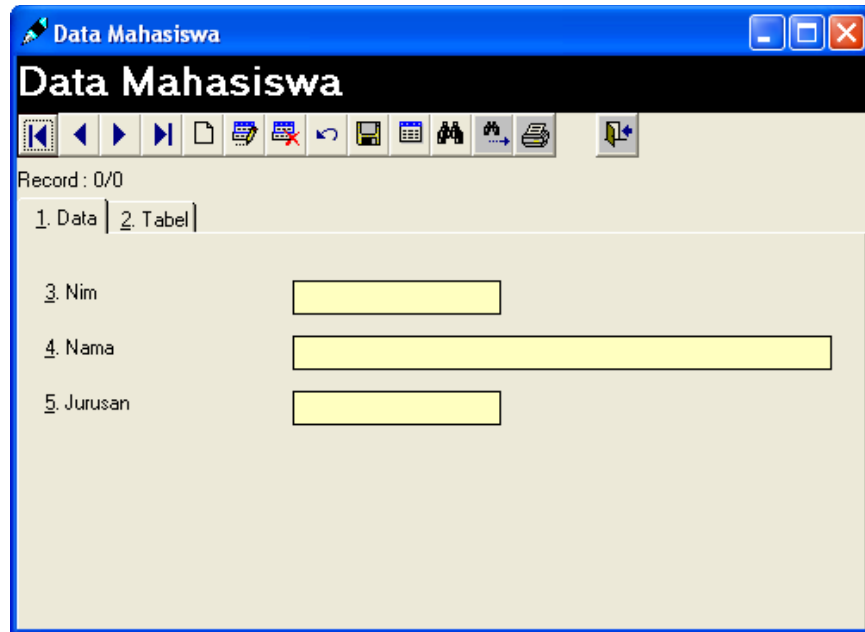
Merupakan aplikasi yang dijalankan untuk memantau penerimaan sms dan melakukan respon terhadap isi sms yang diterima. Aplikasi ini dijalankan pada komputer yang difungsikan sebagai sms Gateway yang ditunjukkan pada Gambar 4.



Gambar 4. Interface sms gateway

### 7.5.2. Entry data mahasiswa

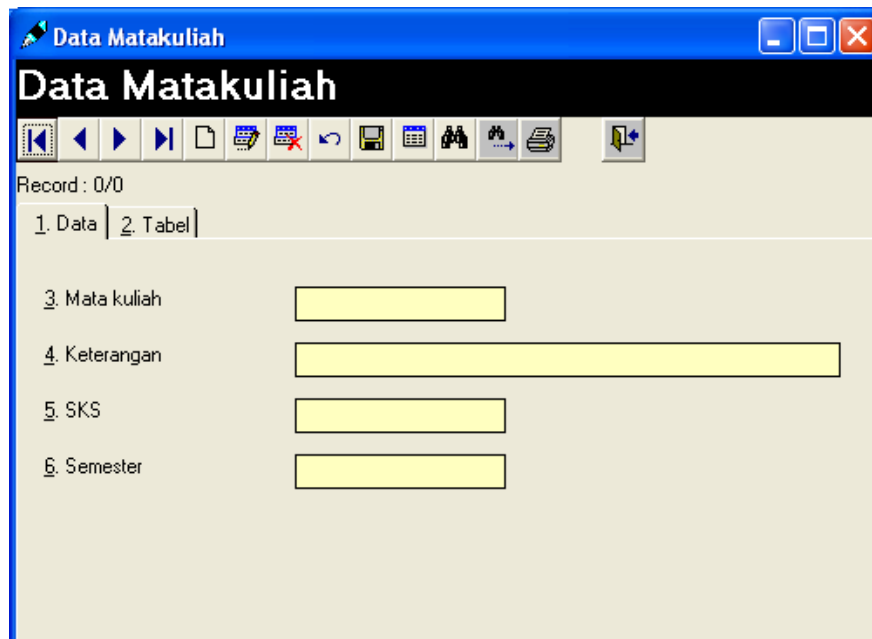
Form ini digunakan untuk entri data mahasiswa baru yang terdiri dari nim, nama, dan jurusan sebagaimana yang ditunjukkan pada Gambar 5.



Gambar 5. Interface entry data mahasiswa

### 7.5.3. Entry data matakuliah

Form ini digunakan untuk entri data matakuliah yang terdiri dari kode matakuliah, keterangan, jumlah sks, dan semester sebagaimana yang ditunjukkan pada Gambar 6.



Gambar 6. Interface entry data mata kuliah

### 7.5.4. Entry data Dosen

Form ini digunakan untuk entri data matakuliah yang terdiri dari kode matakuliah, keterangan, jumlah sks, dan semester sebagaimana yang ditunjukkan pada Gambar 7.

The screenshot shows a software window titled "Data Dosen". At the top, there is a menu bar with "1. Data" and "2. Tabel". Below the menu bar is a toolbar with icons for navigation (back, forward, home, end), editing (undo, redo, delete, insert), and other functions. The main area of the window displays "Record : 0/0" and two input fields: "3. Dosen" and "4. Nama".

Gambar 7. Interface entry data dosen

### 7.5.5. Entry Data KHS

Form ini digunakan untuk entri data KHS hasil ujian mahasiswa sebagaimana yang ditunjukkan pada Gambar 8.

The screenshot shows a software window titled "Data KHS". At the top, there is a menu bar with "1. Data" and "2. Tabel". Below the menu bar is a toolbar with icons for navigation and editing. The main area of the window displays "Record : 0/0" and several input fields: "3. Tanggal", "4. Dosen", "5. Mata Kuliah", "6. Nim", "7. Semester", and "8. Nilai huruf". There are also labels for "Nama" and "Jurusan".

Gambar 8. Interface entry data KHS

## 8. Implementasi

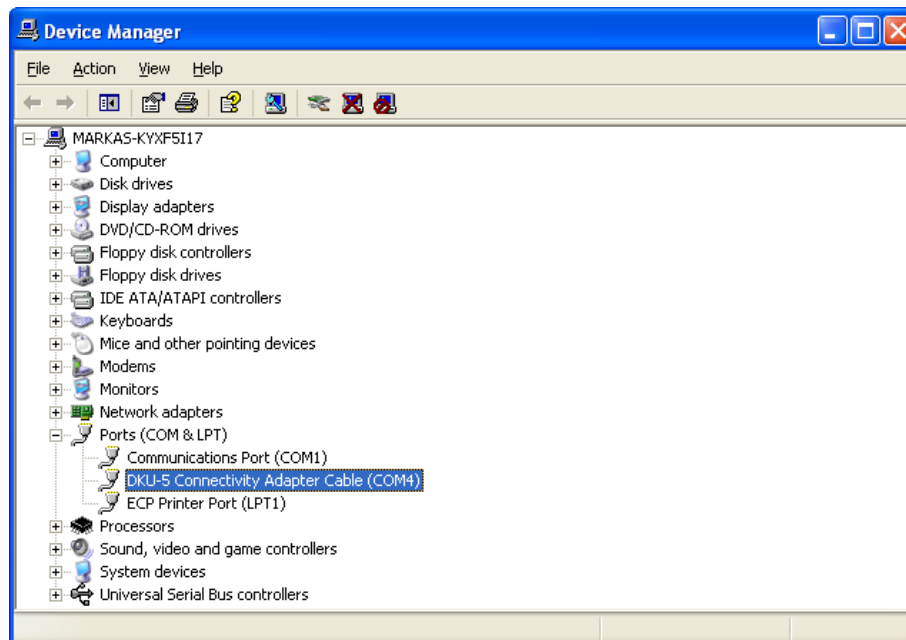
Untuk mengimplementasikan program penerimaan dan pengiriman sms nilai mahasiswa diperlukan beberapa perangkat keras pendukung yaitu :

1. Handphone yang build in GSM Modem, dalam hal ini penulis menggunakan Nokia 6610i.
2. Kabel DKU-5 sebagai interface antara CPU dengan GSM Modem



Gambar 9. Nokia DKU 5 - Cellular phone cable - USB - 4 pin USB

Selain perangkat keras, juga diperlukan perangkat lunak pendukung yaitu driver Nokia untuk DKU-5.



Gambar 10. Driver DKU-5 terinstalasi sebagai COM4.

Selanjutnya dapat dijalankan aplikasi smsGateway, dan melakukan setting terhadap nomor serial port yang digunakan (dalam hal ini COM4), dan proses penerimaan dan pengiriman sms dapat dilakukan.

## 9. Kesimpulan

1. Pengiriman dan penerimaan SMS melalui PC dapat dilakukan dengan program aplikasi SMS gateway.
2. Aplikasi SMS gateway bekerja dengan menggunakan komunikasi data serial.
3. Perintah pengiriman dan penerimaan sms menggunakan perintah AT sesuai dengan standard ETSI GSM.
4. Pengiriman SMS melalui PC dapat dilakukan dengan modus text dan modus PDU.
5. Modus text tidak tersedia pada semua handphone.
6. Untuk menjalankan SMS gateway diperlukan handphone yang memiliki fasilitas GSM Modem build in.
7. Berdasarkan hasil pengujian dilakukan, aplikasi hasil pengembangan peneliti dapat menerima dan mengirim SMS nilai mahasiswa sesuai dengan tujuan dari penelitian ini.

## 10. Saran

Untuk aplikasi SMS gateway dengan kapasitas yang tinggi sebaiknya menggunakan perangkat khusus GSM Modem.

## Lampiran :

Berikut ini adalah modul untuk penerimaan dan pengiriman SMS dengan perintah AT.

Option Explicit

Dim SendingAndReceiving As Boolean

Dim CheckingPort As Boolean

Dim Indications As String

```
Private Sub cmdRefresh_Click()  
txtSmsc.Text = GetSMSCNum()  
End Sub
```

```
Private Sub Form_Load()  
Call RefreshSetting  
MSComm1.Settings = "9600,n,8,1"  
MSComm1.CommPort = xComPort  
MSComm1.Handshaking = comRTS  
MSComm1.SThreshold = 0  
MSComm1.PortOpen = True  
MSComm1.InBufferCount = 0  
'Test Modem  
Call SendAndResponse("AT")  
'Get Identity  
Call SendAndResponse("AT+GMI")  
Call SendAndResponse("AT+GMM")  
Call SendAndResponse("AT+GMR")  
Call SendAndResponse("AT+GSN")  
'Setting PDU Mode  
Call SendAndResponse(xPduMode)  
'Setting SMS Memory  
Call SendAndResponse(xSMSMemory)  
Indications = ""  
SendingAndReceiving = False  
CheckingPort = False  
Timer1.Interval = xCheckDuration  
Timer1.Enabled = True  
End Sub
```

```

Private Sub Form_Resize()
On Error Resume Next
lblTitle.Move 0, 0
lblTitle.Width = Me.ScaleWidth
End Sub

Private Sub Form_Unload(Cancel As Integer)
If MsgBox("Apakah anda yakin tutup SMS Gateway", vbYesNo + vbQuestion, "Sms gateway")
= vbYes Then
    MSComm1.PortOpen = False
Else
    Cancel = True
End If
End Sub

'Periksa indication received sms
Private Sub Timer1_Timer()
Dim rec As String
If Not CheckingPort Then
    CheckingPort = True
    If Not SendingAndReceiving Then
        Do While MSComm1.InBufferCount > 0
            rec = MSComm1.Input
            Indications = Indications + rec
            txtResponded.Text = txtResponded.Text + rec
        Loop
        If Right(Indications, 2) = vbCrLf Then
            Call Analisa(Indications)
            Indications = ""
        End If
    End If
    CheckingPort = False
End If
End Sub

Private Sub Analisa(iStr As String)
Dim Mem3 As String
Dim Comma As Integer
Dim Index As Integer
Dim sSMS As String
Dim sPDU As String
If Left(iStr, 9) = vbCrLf + "+CMTI: " Then
    Comma = InStr(iStr, ",")
    Mem3 = Mid(iStr, 10, Comma - 10)
    Index = Val(Mid(iStr, Comma + 1))
    If Index > 0 Then
        'Set memory location
        Call SendAndResponse("AT+CPMS=" & Mem3)
        'Get sms
        sSMS = SendAndResponse("AT+CMGR=" & Index)
        sPDU = ExtractPDU(sSMS)
        Call AnalisaPDU(sPDU)
        'Delete sms
        Call SendAndResponse("AT+CMGD=" & Index)
    End If
End If
End Sub

Private Sub AnalisaPDU(sPDU As String)

```



```

Dim sSMS As New dsSMS
Dim rSMS As New dsSMS
Dim oPDU As New clsPDU
Dim sSQL As String
Dim kdm As String
Dim kd1 As String
Dim kd2 As String
Set sSMS = oPDU.PDUToSMS(sPDU)
ParseSMS(sSMS,nim, semester,matakuliah)
rSMS.tSmsc = txtSmsc.Text
rSMS.tSmsn = sSMS.tSmsn
kdm = SmsNilai(nim, semester)
If kdm <> "" Then
    Do
        kd1 = split160(kdm, kd2)          'potong 16 character
        rSMS.tMessage = kd1
        Call SendSMS(rSMS)              'jawab sms motor type
        kdm = kd2
    Loop Until kdm = ""
End If
Set sSMS = Nothing
Set rSMS = Nothing
Set oPDU = Nothing
End Sub

'function buat sms nilai
Private Function SmsNilai(nim as string, semester As integer) As String
Dim rs As New Recordset
Dim sSQL As String
Dim SMS As String

sSQL = "Select * From KHS Where nim='" & Nim & "' And Semester='" & semester &';"

rs.Open sSQL, MyConn, adOpenStatic, adLockReadOnly, adCmdText

SMS = ""

Do While Not rs.EOF
    If SMS = "" Then
        SMS = rs("Matakuliah") & ":" & rs("NilaiHuruf")
    Else
        SMS = SMS & "," & rs("Matakuliah") & ":" & rs("NilaiHuruf")
    End If
    rs.MoveNext
Loop
rs.Close

SmsNilai = SMS
End Function

Private Function split160(ByVal SMS As String, ByRef sisa As String) As String
Dim lkoma As Integer
Dim I As Integer
Dim result As String
lkoma = 0
result = ""
If Len(SMS) > 160 Then
    For I = 1 To 160
        If Mid(SMS, I, 1) = "," Then

```

```

        lkoma = I
    End If
Next I
result = Left(SMS, lkoma - 1)
sisa = Mid(SMS, lkoma + 1)
Else
    result = SMS
    sisa = ""
End If
split160 = result
End Function

'function kirim sms
Private Sub SendSMS(tSms As dsSMS)
Dim sPDU As String
Dim oPDU As New clsPDU
Dim tPDULen As Integer
Dim sResponse As String
sPDU = oPDU.SMSToPDU(tSms, tPDULen)
sResponse = SendAndResponse("AT+CMGS=" & tPDULen)
If InStr(sResponse, ">") <> 0 Then
    sResponse = SendAndResponse(sPDU & Chr(26))
End If
Set oPDU = Nothing
End Sub

'proses pdu menjadi info sms
Private Function ExtractPDU(sSMS As String)
Dim nATC As Integer
Dim nATR As Integer
Dim nPDU As Integer
Dim cAT As String
Dim rAT As String
nATC = InStr(sSMS, vbCrLf)
nATR = InStr(nATC + 3, sSMS, vbCrLf)
nPDU = InStr(nATR + 2, sSMS, vbCrLf + vbCrLf)
cAT = Mid(sSMS, 1, nATC - 1)
rAT = Mid(sSMS, nATC + 3, nATR - (nATC + 3))
If nPDU = 0 Then
    ExtractPDU = ""
Else
    ExtractPDU = Mid(sSMS, nATR + 2, nPDU - (nATR + 2))
End If
End Function

'user at command
Private Sub txtAT_KeyPress(KeyAscii As Integer)
Dim X As Long
If KeyAscii = 13 Then
    If Not CheckingPort Then
        txtResponded.Locked = True
        Call SendAndResponse(txtAT.Text)
        txtAT.Text = ""
        txtResponded.Locked = False
    End If
Else
    KeyAscii = Asc(UCCase(Chr(KeyAscii)))
End If

```

```
End Sub
```

```
'buat textbox tetap menampilkan data terakhir  
Private Sub txtResponded_Change()  
txtResponded.SelStart = Len(txtResponded.Text)  
End Sub
```

```
Private Function GetSMSCNum() As String  
Dim Ret As String  
Ret = SendAndResponse("AT+CSCA?")  
GetSMSCNum = Mid(Ret, InStr(Ret, ":") + 1, InStr(Ret, ",") - InStr(Ret, ":") + 1)  
End Function
```

```
'function mengirim data ke TA  
Private Function SendAndResponse(sSend As String) As String  
Dim X As Long  
Dim cTime As Date  
Dim sResponse As String  
Dim lTimeout As Boolean  
lTimeout = False  
cTime = Now  
Do While MSComm1.InBufferCount > 0  
If (Now - cTime) * 24 * 3600 > 60 Then 'Timeout 60 detik  
lTimeout = True  
Exit Do  
End If  
X = DoEvents  
Loop  
txtResponded.Text = txtResponded.Text & sSend & vbCrLf  
If Not lTimeout Then  
SendingAndReceiving = True  
MSComm1.InBufferCount = 0  
MSComm1.Output = sSend & vbCrLf  
sResponse = ""  
cTime = Now  
Do While (Now - cTime) * 24 * 3600 < 10 And _  
InStr(sResponse, "OK") = 0 And _  
InStr(sResponse, "ERROR") = 0  
X = DoEvents  
Do While MSComm1.InBufferCount > 0  
sResponse = sResponse + MSComm1.Input  
Loop  
Loop  
MSComm1.InBufferCount = 0  
If sResponse = "" Then sResponse = "NR" & vbCrLf  
SendingAndReceiving = False  
txtResponded.Text = txtResponded.Text & sResponse  
SendAndResponse = sResponse  
Else  
txtResponded.Text = txtResponded.Text & "TIMEOUT" & vbCrLf  
SendAndResponse = "TIMEOUT" & vbCrLf  
End If  
End Function
```

Berikut ini adalah class sms untuk konversi septet ke octet dan sebaliknya.

```
Option Explicit
```

```
'konversi Sms ke Pdu
```

```

Public Function SMSToPDU(tSms As dsSMS, ByRef tPDULen As Integer) As String
Dim otSMSC As String
Dim otSMSN As String
Dim otMessage As String
Dim oPDU As String
If Left(tSms.tSmsc, 2) <> "91" Then
    otSMSC = "91" & num2octets(tSms.tSmsc)
Else
    otSMSC = "91" & num2octets(Mid(tSms.tSmsc, 3))
End If
otSMSN = num2octets(tSms.tSmsn)
otMessage = sms2octets(tSms.tMessage)
oPDU = "11" & "00" & dec2hex(Len(tSms.tSmsn)) & "91" & otSMSN & "00" & "00" & "A7" &
dec2hex(Len(tSms.tMessage)) & otMessage
tPDULen = Len(oPDU) / 2
SMSToPDU = dec2hex(Len(otSMSC) / 2) & otSMSC & oPDU
End Function

```

'konversi Pdu ke Sms

```

Public Function PDUtoSMS(X As String) As dsSMS
Dim tSms As New dsSMS
tSms.tSmsc = PduSmscNum(X)
tSms.tSmsn = PduSmsnNum(X)
tSms.tDate = PduSmsDate(X)
tSms.tTime = PduSmsTime(X)
tSms.tMessage = Bin82Str(X)
Set PDUtoSMS = tSms
End Function

```

```

Public Function Bin82Str(X)
Dim I As Integer
Dim b As String
Dim s As String
b = SmsOctets2Bin8(X)
s = ""
For I = Len(b) To 1 Step -7
    If I >= 7 Then
        s = s & Chr(bin2dec(Mid(b, I - 7 + 1, 7)))
    End If
Next I
Bin82Str = s
End Function

```

```

Public Function SmsOctets2Bin8(X)
Dim I As Integer
Dim o As String
Dim b As String
o = PduSmsOctets(X)
b = ""
For I = 1 To Len(o) Step 2
    b = hex2bin8(Mid(o, I, 2)) & b
Next I
SmsOctets2Bin8 = b
End Function

```

```

Public Function PduSmsOctets(X)
PduSmsOctets = Mid(X, 5 + PduSmscLen(X) * 2 + 4 + PduSmsnLenF(X) + 2 + 2 + 6 + 6 + 2
+ 2)
End Function

```

```

Public Function PduSmsLen(X)
PduSmsLen = Val("&H" & Mid(X, 5 + PduSmscLen(X) * 2 + 4 + PduSmsnLenF(X) + 2 + 2 + 6
+ 6 + 2, 2))
End Function

```

```

Public Function PduSmsTimeZone(X)
PduSmsTimeZone = Mid(X, 5 + PduSmscLen(X) * 2 + 4 + PduSmsnLenF(X) + 2 + 2 + 6 + 6,
2)
End Function

```

```

Public Function PduSmsTime(X)
Dim o As String
o = num2octets(PduSmsTimeOctets(X))
PduSmsTime = TimeSerial(Val(Left(o, 2)), Val(Mid(o, 3, 2)), Val(Mid(o, 5, 2)))
End Function

```

```

Public Function PduSmsTimeOctets(X)
PduSmsTimeOctets = Mid(X, 5 + PduSmscLen(X) * 2 + 4 + PduSmsnLenF(X) + 2 + 2 + 6, 3
* 2)
End Function

```

```

Public Function PduSmsDate(X)
Dim o As String
o = num2octets(PduSmsDateOctets(X))
PduSmsDate = DateSerial(Val(Left(o, 2)), Val(Mid(o, 3, 2)), Val(Mid(o, 5, 2)))
End Function

```

```

Public Function PduSmsDateOctets(X)
PduSmsDateOctets = Mid(X, 5 + PduSmscLen(X) * 2 + 4 + PduSmsnLenF(X) + 2 + 2, 3 * 2)
End Function

```

```

Public Function PduSmsnNum(X)
Dim n As String
n = num2octets(PduSmsnOctets(X))
If Right(n, 1) = "F" Then
    PduSmsnNum = Left(n, Len(n) - 1)
Else
    PduSmsnNum = n
End If
End Function

```

```

Public Function PduSmsnOctets(X)
PduSmsnOctets = Mid(X, 5 + PduSmscLen(X) * 2 + 4, PduSmsnLenF(X))
End Function

```

```

Public Function PduSmsnType(X)
PduSmsnType = Mid(X, 5 + PduSmscLen(X) * 2 + 2, 2)
End Function

```

```

Public Function PduSmsnLen(X)
PduSmsnLen = Val("&H" & Mid(X, 5 + PduSmscLen(X) * 2, 2))
End Function

```

```

Public Function PduSmsnLenF(X)
Dim SMSnLen As Integer
SMSnLen = PduSmsnLen(X)
If SMSnLen Mod 2 <> 0 Then SMSnLen = SMSnLen + 1
PduSmsnLenF = SMSnLen

```

End Function

```
Public Function PduSmscNum(X)
Dim n As String
n = num2octets(PduSmscOctets(X))
If Right(n, 1) = "F" Then
    PduSmscNum = Left(n, Len(n) - 1)
Else
    PduSmscNum = n
End If
End Function
```

```
Public Function PduSmscOctets(X)
PduSmscOctets = Mid(X, 5, PduSmscLen(X) * 2 - 2)
End Function
```

```
Public Function PduSmscType(X)
PduSmscType = Mid(X, 3, 2)
End Function
```

```
Public Function PduSmscLen(X)
PduSmscLen = Val(Left(X, 2))
End Function
```

```
Public Function num2octets(X)
Dim I As Integer
Dim n As String
Dim o As String
n = X
If Len(X) Mod 2 = 1 Then
    n = n + "F"
End If
o = ""
For I = 1 To Len(n) Step 2
    o = o + Mid(n, I + 1, 1) + Mid(n, I, 1)
Next I
num2octets = o
End Function
```

```
Public Function sms2octets(X)
Dim I As Integer
Dim b7 As String
Dim o As String
b7 = sms2bin7(X)
For I = Len(b7) To 1 Step -8
    If (I - 8) > 1 Then
        o = o & dec2hex(bin2dec(Mid(b7, I - 8 + 1, 8)))
    Else
        o = o & dec2hex(bin2dec(Mid(b7, 1, I)))
    End If
Next I
sms2octets = o
End Function
```

```
Public Function sms2bin7(X)
Dim I As Integer
Dim b As String
For I = 1 To Len(X)
    b = chr2bin7(Mid(X, I, 1)) & b
Next I
End Function
```

```

Next I
sms2bin7 = b
End Function

Public Function dec2hex(X)
dec2hex = Right("0" & Hex(X), 2)
End Function

Public Function bin2dec(X)
Dim I As Integer
Dim k As Integer
Dim d As Integer
k = 1
d = 0
For I = Len(X) To 1 Step -1
    d = d + Val(Mid(X, I, 1)) * k
    k = k * 2
Next I
bin2dec = d
End Function

Public Function chr2bin7(X)
chr2bin7 = Right("0000000" & chr2bin(X), 7)
End Function

Public Function chr2bin(X)
chr2bin = dec2bin(Asc(X))
End Function

Public Function hex2bin8(X)
hex2bin8 = Right("00000000" & hex2bin(X), 8)
End Function

Public Function hex2bin(X)
Dim d As Integer
d = Val("&H" & X)
hex2bin = dec2bin(d)
End Function

Public Function dec2bin(X)
Dim r As Integer
Dim b As String
r = X
Do While r > 0
    b = (r Mod 2) & b
    r = r \ 2
Loop
dec2bin = b
End Function

```

## Daftar Pustaka

Lars Pettersson, SMS and the PDU format, URL : <http://www.dreamfabric.com/sms>

Nokia Mobile Phone (2000), AT Command Set for Nokia GSM Product, URL :  
[http://www.dipol.com.ua/NOKIA\\_AT-help.pdf](http://www.dipol.com.ua/NOKIA_AT-help.pdf)

Stefen Fringe (2005), SMS Applications, URL : <http://smstools.meinemullemaus.de/book.html>

Prem (2004), serial port communication using visual basic, URL :  
[http://www.control.com/1026198113/index\\_html](http://www.control.com/1026198113/index_html)

### Biodata Penulis :

Nama Lengkap : Hendra

Gelar yang diperoleh : ST. (Jurusan Teknik Industri, USU-Medan)  
MT. (Jurusan Teknik Industri, USU-Medan)

Afiliasi : Teknik Informasi, STMIK IBBI

Jabatan : Dosen Tetap

Mata kuliah yang diajar : Bahasa Rakitan, Keamanan Komputer, Database Lanjut