

LEMBAR KERJA-4

PRAKTIK KOMUNIKASI DATA

I. Pokok Bahasan

PENGEMBANGAN APLIKASI MODEM

II. Judul

KOMUNIKASI DATA ANTAR DTE MENGGUNAKAN HANDPHONE

III. Tujuan

Setelah selesai praktek diharapkan mahasiswa dapat :

1. Memahami tentang cara kerja modem
2. Memahami cara kerja komunikasi data antara DTE dan Wireless Modem (Handphone)
3. Mengaplikasikan instruksi *AT Command* untuk membangun komunikasi data antar DTE (komputer) menggunakan pesawat handphone

IV. Teori Singkat

Modem adalah suatu piranti yang digunakan untuk mengubah isyarat digital menjadi isyarat analog (pada frekuensi suara) dan sebaliknya sehingga dapat dilewatkan pada jaringan PSTN (telpn publik).

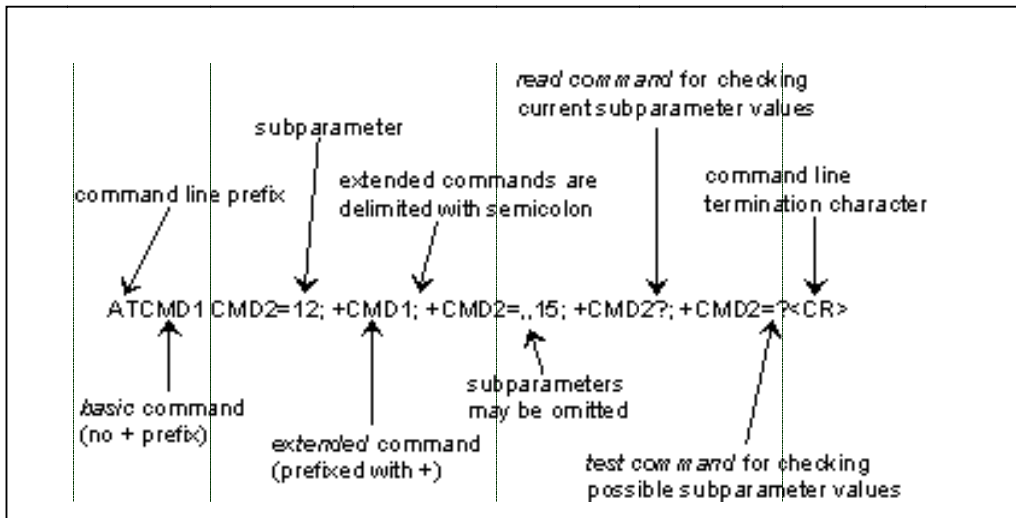
Modem singkatan dari modulasi-demodulasi. Modulasi adalah mengubah isyarat digital (keluaran dari port serial PC) menjadi isyarat analog. Isyarat digital ditumpangkan pada isyarat analog pada frekuensi suara. Sedangkan demodulasi adalah proses kebalikan dari modulasi.

PC berkomunikasi dengan perangkat modem menggunakan instruksi AT Command. Sementara itu pesawat handphone dalam perkembangannya juga mendukung atau ada fitur wireless modem di dalamnya, sehingga pesawat handphone dapat "bercakap-cakap" dengan PC menggunakan instruksi *AT Command* ini. Instruksi *AT Command* yang lengkap dan mudah didownload secara bebas diinternet adalah *AT Command* untuk handphone tipe Nokia dan Siemens. Handphone Nokia yang mendukung instruksi *AT Command* harus ada fitur *wireless modem* di dalamnya (misal: Nokia 3220, 6610, 6100, 6020). Sementara itu untuk handphone merk Siemens (pelopor fitur *AT Command* pada handphone) relatif lebih banyak seri yang mendukung hampir semua seri Cxx, Mxx dan Sxx mendukung instruksi *AT*.

Pemrograman *AT Command* berbasis PC

AT singkatan dari *Attention* (perhatian). Sebenarnya komputer menggunakan perintah AT ini untuk berkomunikasi dengan modem, sehingga ketika dilakukan pemrograman handphone menggunakan *AT Command* maka komputer menganggap handphone sebagai modem (GSM modem). Inilah sebabnya mengapa format data selalu menggunakan standar RS232 (serial asinkron), yang memang port serial PC dirancang untuk berkomunikasi dengan modem.

Jenis perintah AT ada yang tidak sama atau tidak kompatibel antara handphone jenis satu dengan yang lain. Berikut sintak umum dari perintah *AT command*.



Gambar 1. Sintak umum instruksi AT pada pesawat handphone nokia

Instruksi-instruksi AT yang penting untuk diketahui seperti dalam tabel 2 berikut.

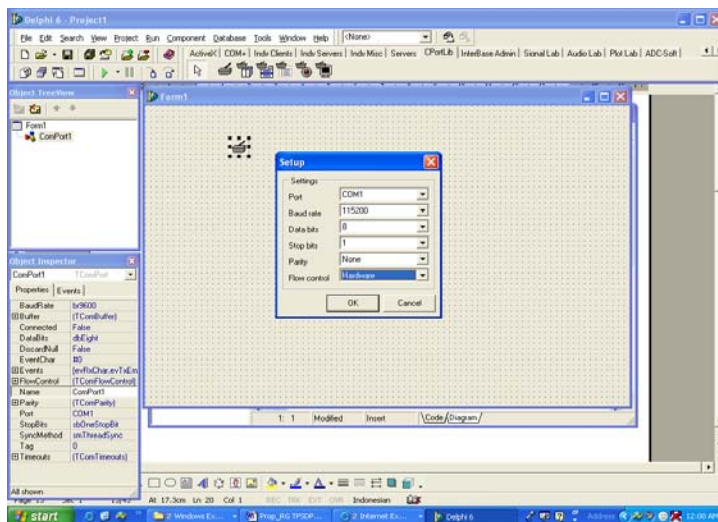
Tabel 1. Daftar instruksi AT yang penting diketahui.

No	Syntax	Arti
1.	AT+CGMM	Baca seri dari Hp
2.	AT+CPBR=3	Baca data Phone book nomor 3
3.	AT+CGSN	Tampilkan IMEI / serial number
4.	AT+CSCS="GSM"; +CMGR= 2	Membaca sms nomor <index>, CSCS= pilih tipe karakter
5.	Nokia: ATD[02723113931]; +&c[0] siemen: ATD[02723113931]; &c0	Call pesawat nomor (0272)3113931. +&c[0] : artinya carrier selalu ON
6.	AT+CHUP	Hung up (menghentikan call)
7.	AT+CNMI=1,1,2,1,1	Contoh perintah utk setting "New Message Indicator" , jika ada SMS baru ditampilkan di komputer:
	AT+CNMI= 1,2,0,0,1	Jika ada SMS baru, isinya ditampilkan langsung di komputer, tetapi sebelumnya harus dilakukan instruksi : AT+CSMS=1
8.	AT+CPMS="SM" / "ME"	Memilih memori yang aktif digunakan: SM = Mem SIM Card, ME = Memori handphone
9.	Nokia: AT+CMGL Siemen: AT+CMGL=4	Untuk menampilkan sms seluruhnya dalam HP
10.	AT+CMGD=1	Untuk menghapus SMS nomor 1
11.	AT+CMGR=1	Membaca SMS indek 1 di memori Hp
12.	Nokia: AT+CMGS="02723113931" Siemen: lihat langkah praktikum ini	Nokia: mulai kirim SMS ke nomor 02723113931. Muncul prompt >. Lalu ketikkan pesan anda. Akhiri dengan CTRL+Z

Setelah mengetahui instruksi-instruksi AT, lalu untuk mengimplementasikan atau memanfaatkan instruksi-instruksi tersebut adalah dengan mengirimkannya secara berurutan sesuai aturan atau sintak ke pesawat handphone melalui suatu program aplikasi. Dalam Windows Xp sudah disediakan suatu program aplikasi untuk

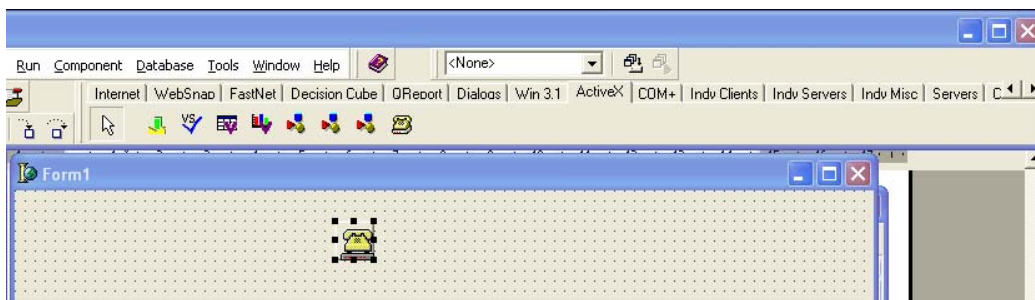
komunikasi dengan port serial (COMn, di mana n=1,2,3...) yakni program *Hyperterminal*. Apabila kabel data handphone sudah terpasang dan berjalan dengan baik di PC, maka sebagai contoh misalnya, dalam jendela *hyperterminal* ketikkan: "AT+CGMM" lalu Enter, maka dibawah instruksi itu akan muncul tulisan tentang jenis dan nomor seri handphone, selain itu juga menandakan bahwa handphone telah siap melakukan komunikasi data lagi dengan PC.

Sementara itu dalam kompilier Borland Delphi 6 ada suatu komponen tambahan yang dibuat khusus untuk komunikasi data dengan port serial asinkron (RS232). Komponen ini disebut dengan *Comport* dan dapat diinstal ke IDE Delphi versi 5, 6 dan 7 dengan mudah. Komponen ini dapat diperoleh di internet secara bebas termasuk *source code* programnya. Gambar 2 berikut ini menampilkan contoh pemakaian komponen *Comport* yang telah diinstal dalam Delphi 6. Perhatikan kotak dialog untuk setup *Comport*, hampir sama jika dibandingkan dengan kotak dialog pada setting *hyperterminalnya* Windows Xp.



Gambar 2. Tampilan dalam Delphi 6 yang sudah terdapat komponen *Comport*

Sebenarnya dalam Compiler Borland Delphi versi 6 ke atas sudah tersedia fasilitas atau komponen tambahan (ActiveX Control) untuk pemrograman komunikasi data lewat port serial (COM). Komponen ini disebut dengan *TMSComm*. Komponen tambahan *TMSComm* harus diinstall sendiri lewat menu: *Component | Import ActiveX Control*. Penggunaan komponen *TMSComm* relatif lebih rumit karena dibutuhkan rutin sendiri untuk kendali penerimaan byte-byte data yang masuk dari port serial.



Gambar 3. Cuplikan IDE Delphi 6 dan form aplikasi yang sudah terinstall komponen *TMSComm*

Protocol Data Unit (PDU) Format

Hal penting yang harus diketahui dalam pemrograman komunikasi data DTE-Hp menggunakan *AT Command* adalah ketika akan melakukan instruksi-instruksi untuk pengiriman dan pembacaan SMS (*Short Message Service*). Setiap pengiriman SMS, baik dari HP 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 adalah contoh PDU yang diterima oleh HP (New SMS atau Inbox):

```
06 91 2618010000 24 0C 91 261822178527 00 00 704020 400524 82
07 4576794D97BF01
```

Dengan keterangan sebagai berikut:

Oktet / Digit Hexa	Keterangan
06	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.
2618010000	Nomor SMSC yang digunakan. Jika jumlah digit nomor SMS adalah ganjil, maka digit paling belakang dipasangkan dengan huruf F. Kalau diterjemahkan, nomor SMSC yang digunakan adalah: +6281100000 (Telkomsel), contoh yg lain: +62855000000 (IM3)
24	Oktet pertama untuk pesan SMS yang diterima
0C	Panjang digit dari nomor pengirim (0C hex = 12 desimal)
91	Jenis nomor pengirim (sama dengan jenis nomor SMSC)
261822178527	Nomor pengirim SMS, yang jika diterjemahkan adalah +628563619400
00	Pengenal protokol, dalam hal ini adalah 0
00	Skema pengkodean SMS, juga bernilai 0
704020 400524 82	Waktu pengiriman, yang berarti 07-04-02 (2 April 2007), dan jam 04:50:42. Sedangkan 82 adalah Timezone yang digunakan.
07	Panjang dari pesan SMS, dalam hal ini adalah 7 huruf (dalam mode 7 bit).
45 76 79 4D 97 BF 01	Pesan SMS dalam mode 7 bit. Jika diterjemahkan kedalam 8 bit, lalu dirubah ke ASCII, maka didapat pesan 'Elektro'

Frame data dengan format seperti diatas adalah untuk pembacaan SMS yang diterima (inbox). Sementara itu PDU yang diterima dari HP berbeda dengan PDU yang untuk dikirim, namun tetap menggunakan aturan yang sama. Perbedaannya terutama pada field yang dikandung, baik nilai maupun fungsinya.

Berikut ini contoh struktur PDU untuk dikirim:

06 91 2618010000 11 00 0B 81 8021725178F2 00 00 FF 0A
CCB018B47CB7C9613A08

Keterangan:

Oktet / Digit Hexa	Keterangan
06	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.
2618010000	Nomor SMSC yang digunakan. Jika jumlah digit nomor SMS adalah ganjil, maka digit paling belakang dipasangkan dengan huruf F. Kalau diterjemahkan, nomor SMSC yang digunakan adalah : 081100000, Contoh lain : +62855000000 (IM3) dikodekan: 2658050000F0
11	Oktet pertama untuk PDU SMS untuk dikirim (SMS SUBMIT).
00	TP-Message-Reference. Diisi "00" agar diisi otomatis oleh handphone.
0B	Panjang digit dari nomor penerima. Disini 0C hex = 12 desimal, artinya panjang digit nomor tujuan 12 digit
81	Jenis nomor penerima (sama dengan jenis nomor SMSC)
8021725178F2	Nomor penerima SMS, yang jika diterjemahkan adalah 08122715872
00	Pengenal protokol, dalam hal ini adalah 0.
00	Skema pengkodean SMS, juga bernilai 0.
FF	Validitas waktu. FF berarti maksimum.
0A	Panjang dari pesan SMS, dalam hal ini adalah 10 huruf (dalam mode 7 bit).
CC B0 18 B4 7C B7 C9 61 3A 08	Pesan SMS dalam mode 7 bit. Jika diterjemahkan kedalam 8 bit, lalu dirubah ke ASCII, maka didapat pesan 'Lab Komdat'

Untuk handphone tertentu, pemberian nomor SMSC pada PDU dapat dihilangkan, untuk kemudian akan diisi oleh handphone sesuai dengan kartu yang sedang digunakan. Tapi untuk lebih amannya, nomor SMSC yang digunakan sebutkan saja.

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, penghitungan jumlah oktet mulai dari kode SMS SUBMIT (11). Untuk contoh di atas, berarti jumlah digit dalam PDU adalah 48, yang berarti terdiri dari 24 oktet. Sehingga perintah yang digunakan adalah:

AT+CMGS=24

Setelah itu, tunggu respon dari handphone. Kalau gagal, dengan berbagai alasan, maka yang dikembalikan adalah ERROR. Sedangkan jika perintah tersebut diijinkan, maka yang dikembalikan adalah karakter '>' (lebih besar) yang bisa diartikan sebagai Prompt untuk penulisan pesan yang akan dikirim.

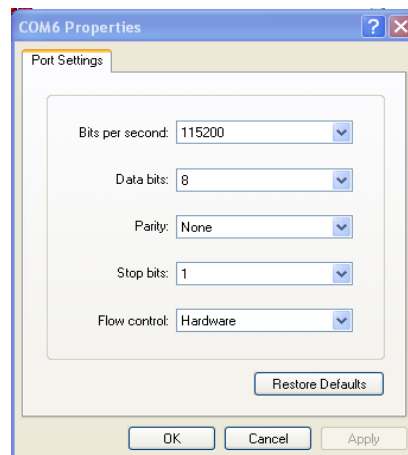
Jika perintah tersebut sukses, selanjutnya adalah menuliskan semua PDU tersebut ke handphone, dan diakhiri dengan penulisan karakter ASCII 26 (CTRL+Z). Jika sukses, maka yang dikembalikan adalah OK, dan SMS segera akan terkirim (jika pulsa masih mencukupi).

V. Langkah Praktikum

PRAKTIK A.

Pengenalan *AT Command* Menggunakan *Hyperterminal*

1. Tancapkan kabel data Siemen C55 ke port serial COM1. Pasangkan pada Hp siemen yang sesuai.
2. Bukalah program *Hyperterminal*
3. Set port pada COM1. Aturlah konfigurasi parameter seperti Gambar berikut:



Gambar 4.
Kotak dialog Setting
Parameter Port Serial di
Hyperterminal

Baut rate : 115000
Lebar kata : 8 bit
Parity : None
Stop bit : 1
Flow control : Hardware

Lalu klik OK

4. Klik tombol "Connect", tunggu sebentar. Pastikan kursor berkedip di jendela *Hyperterminal*, dan di status bar tertampil tulisan: "Connected".
5. Pada jendela *hyperterminal* ketikkan

AT (lalu enter)

6. Jika muncul tulisan OK berarti koneksi antara Hp dan Komputer telah benar dan Hp siap menerima instruksi-instruksi *AT Command* berikutnya.
Ketikan :

AT+CGMM (lalu Enter)

Amati apa yang tertampil di jadi selanjutnya

7. Cobalah anda lakukan beberapa instruksi AT seperti dalam tabel diatas. Amati dan catat hasilnya.
8. Coba anda lakukan Instruksi untuk pengiriman SMS seperti berikut ini:

AT+CMGS=18

setelah muncul tanda ">" (prompt) ketikkan pesan dalam format PDU berikut ini:
> 0011000B818021725178F20000A804E3B7380C (lalu tekan Ctrl+Z)
+CMGS: 204
OK

PENTING...!!!

SERING-SERINGLAH MENYIMPAN PROJECT ANDA DALAM DIREKTORI PRIBADI ANDA. PASTIKAN FILE UNIT DAN PROJECT ANDA SUDAH TERSIMPAN DENGAN BENAR.

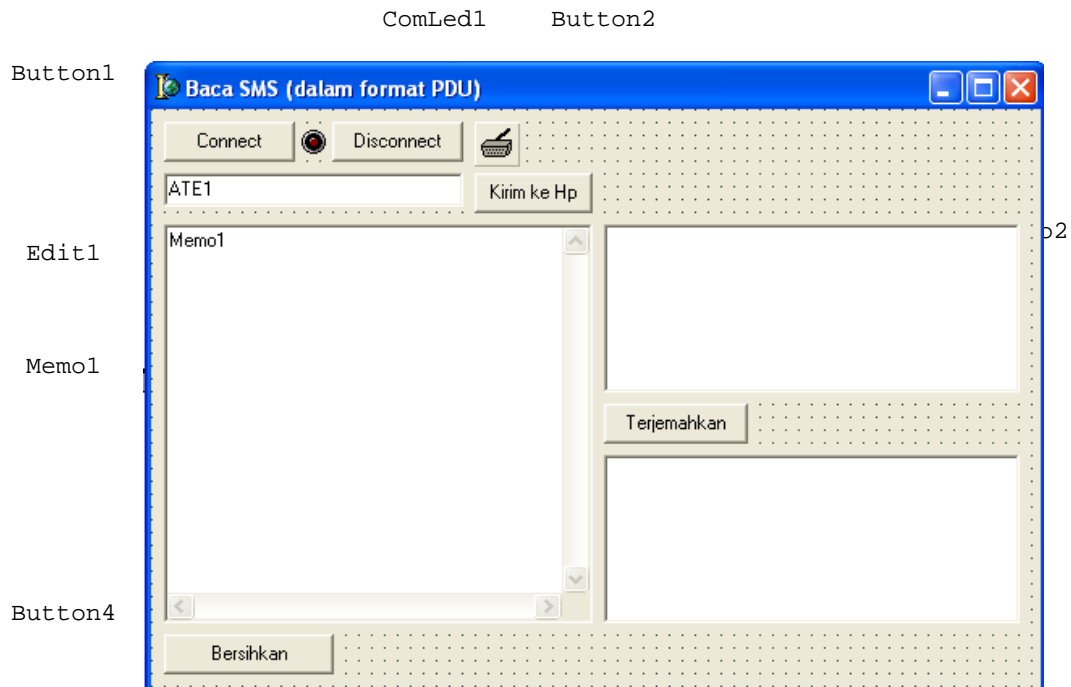
- ☞ **JIKA ANDA INGIN OPEN FILE PAKAILAH MENU *File - Open Project...* JANGAN PAKAI MENU *File - Open...***
- ☞ **BUKALAH FILE YANG BEREKSTENSI *.prj (file project).**
- ☞ **JIKA ANDA OPEN FILE *.pas MAKA PROGRAM ANDA KEMUNGKINAN TIDAK BISA DIJALANKAN.**

PRAKTIK B. PEMROGRAMAN AT COMMAND UNTUK KOMUNIKASI DATA ANTAR DTE (KOMPUTER)

B.1. Pembacaan SMS Diterima

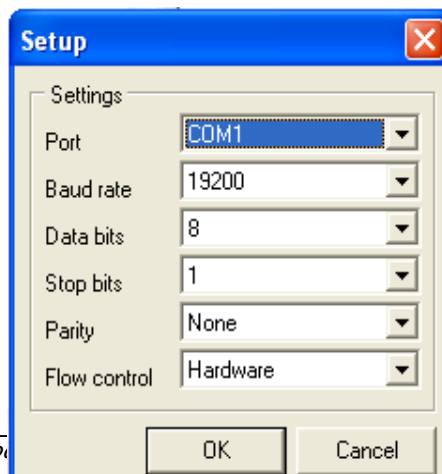
1. Buka atau jalankan program Borland Delphi 7 anda, lalu buat form aplikasi baru (default).
2. Gantilah nilai Caption form tersebut dengan teks:
"Baca SMS (dalam format PDU)"
3. Buatlah/Tambahkan ke dalam form tersebut komponen-komponen TButton (5 buah), TEdit (1 buah), TMemo (3 buah), TComLed (1 buah) dan TcomPort (1 buah).

- Selanjutnya aturlah besar, kecil dan letak/posisi tiap-tiap komponen dalam form tersebut seperti dalam Gambar dibawah ini.



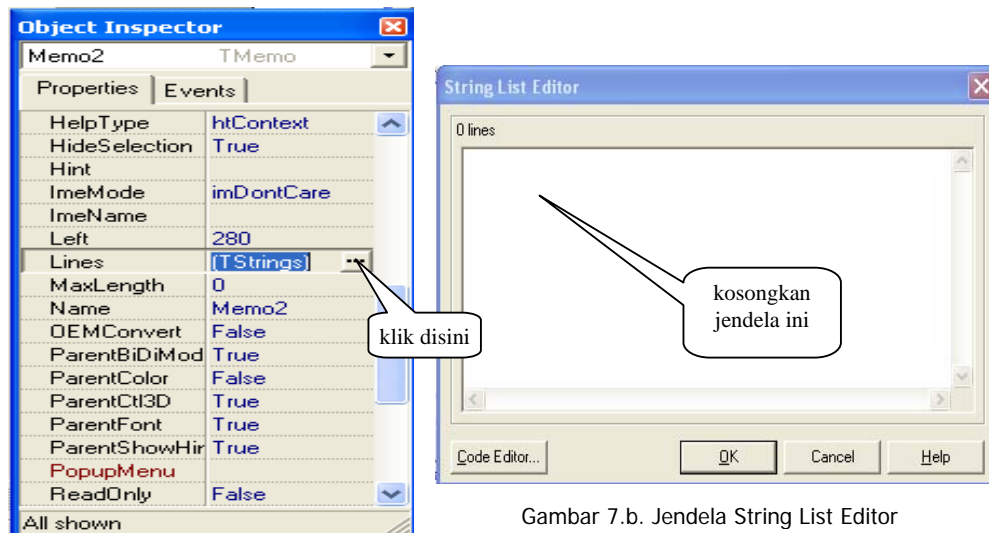
Gambar 5. Form aplikasi yang dibuat dalam praktikum ini

- Gantilah nilai properties Caption dari komponen TButton berikut:
 Button1 = "Connect", Button2 = "Disconnect",
 Button3 = "Kirim ke Hp", Button4 = "Bersihkan",
 Button5 = "Terjemahkan".
- Gantilah nilai property Text dari Edit1 dengan "ATE1".
- Aturlah nilai property ComPort dari ComLed1 = ComPort1.
- Dalam mode Form, Dobel klik komponen ComPort1, maka akan muncul kotak dialog *Setup ComPort*.
 Set dan aturlah nilai setiap parameter seperti Gambar 6 berikut.



Gambar 6.
Kotak dialog Setup dan nilainya dari komponen ComPort

9. Klik komponen Memo1. Pada jendela *Objec Inspector*, tab *Properties*, cari dan pilih properti *ScrollBars*, aturlah nilainya = *ssBoth*.
10. Klik komponen Memo2. Perhatikan Gambar 7.a. berikut. Pada jendela *Object Inspector*, pilih property *Lines*, double klik pada teks *TStrings* atau klik tombol [...] di sebelah kanannya.
11. Maka akan muncul jendela "String List Editor" dari Memo2 (Gambar 7.b.). Kosongkan atau hapus semua teks yang ada dalam jendela tersebut. Lalu klik OK



Gambar 7.a. Tampilan jendela Object Inspector dari Memo2

Gambar 7.b. Jendela String List Editor

12. Gambar 7.a. Klik tab *Events*. Double klik pada *event* *OnClick*-nya. Lalu pada prosedur *Event handler* *OnClick* tersebut isilah dengan 2 baris program berikut:

```

procedure TForm1.Memo2Click(Sender: TObject);
begin
    Memo2.Clear;
    Memo3.Clear;
end;

```

13. Ulangilah langkah **10 & 11** di atas, tetapi untuk komponen Memo3.
14. Double klik komponen Button1. Pada *event handler* *OnClick*-nya ketikkan:


```
ComPort1.Open;
```
15. Double klik komponen Button2. Pada *event handler* *OnClick*-nya ketikkan:


```
ComPort1.Close;
```
16. Double klik komponen Button3. Pada *event handler* *OnClick*-nya ketikkan:

```

procedure TForm1.Button3Click(Sender: TObject);
var str : string;
begin

```

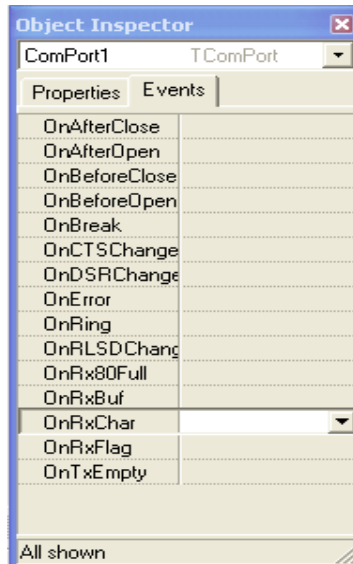
```

str := Edit1.Text+#13#10;
Comport1.WriteString(str);
end;

```

17. Double klik komponen Button4. Pada *event handler* OnClick-nya ketikkan:


```
Memol.Clear;
```
18. Kembali ke mode Form. Klik komponen ComPort1. Pada jendela *Object Inspector*, pilih tab *Events*. Pilih dan double klik pada event "OnRxChar" (Gambar 8).



Gambar 8.
Tampilan jendela *Object Inspector* dari ComPort1 dalam Tab *Events*. Tampak pilihan event : OnRxChar yang siap double klik

double klik
disini

19. Maka anda akan dibawa ke mode editor dalam prosedur *event handler* OnRxChar-nya Comport1. Lalu ketikkan program berikut dalam prosedur *event handler* tersebut.

```

procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
var str : string;
begin
  Comport1.ReadStr(str,Count);
  Memol.Text := Memol.Text + str;
end;

```

Sekarang mulai membuat program untuk mengubah atau menerjemahkan format data PDU dari SMS yang terbaca sehingga dapat dimengerti maknanya. Untuk keperluan ini dibuat sebuah fungsi PDU2Text dan sebuah prosedur ConvertMSG.

Fungsi PDU2Text mempunyai tugas mengkonversi data isi/teks SMS dari format PDU ke dalam bentuk kode ASCII (sehingga dapat direpresentasikan dalam karakter). Fungsi ini mempunyai masukan berupa data string bagian pesan (isi SMS) dari data PDU seluruhnya yang terbaca dan mempunyai keluaran teks isi SMS.

Sementara Prosedur ConvertMSG bertugas memarsing dari seluruh data dalam format PDU yang terbaca menjadi bagian-bagian kecil dan sekaligus

menerjemahkannya sesuai dengan tabel di atas. Prosedur ini mempunyai masukan data PDU terbaca dari Handphone.

20. Ketikkan cuplikan program berikut ini di bagian implementasi (paling atas).

```
var
  Form1: TForm1;

implementation

{$R *.dfm}

//=====
// Fungsi dan Prosedur untuk keperluan menerjemahkan
// data SMS terbaca dari handphone dalam format PDU.
// Penggunaan:
//
// ConvertMSG(<sms terbaca dalam PDU>, <No. SMS Center>, <tipe>,
//           <no pengirim>, <bentuk>, <skema>, <tanggal>, <batas>,
//           <isi>: string);
//
// parameter input hanyalah : <sms terbaca dalam PDU> yang lain adalah
// parameter keluaran.
//
// Sumber : Aryo Sanjaya @ BengkelProgram.Com
//=====

const  sOK = #13#10'OK';
       sERROR = #13#10'ERROR';

function PDU2Text(pdudata: string): string;
var pdu,isi,hasilteks,huruf: string;
    i: integer;
    m,n,vgeser,sisa,c,d,e,f,panjang: byte;
    hasil,dbiner: array[1..9000] of byte;

begin
  if length(pdudata)=0 then begin
    Result := '';
    exit;
  end;

  pdu := copy(pdudata,3,length(pdudata));
  isi:= '';
  panjang := length(pdu) div 2;

  for i := 1 to panjang do begin
    huruf := copy(pdu, i*2 - 1, 2);
    dbiner[i] := StrToInt('$' + huruf);
  end;

  m := 1;
  vgeser := 0;
  sisa := 0;
  n := 1;
  while n <= panjang do begin
    c := dbiner[n];
    d := c shl vgeser;
    e := d or sisa;
    f := e and $7F;
    hasil[m] := f;
  end;
end;
```

```

        Inc(vgeser);
        c := dbiner[n];
        d := c shr (8-vgeser);
        sisa := d;
        inc(m);
        inc(n);
        if vgeser >= 7 then begin
            hasil[m] := sisa and $7F;
            inc(m);
            sisa := 0;
            vgeser := 0;
        end;
    end;
    hasilteks := '';
    for i := 1 to m - 1 do
        hasilteks := hasilteks + chr(hasil[i]);
    Result := hasilteks;
end;

```

```

procedure ConvertMSG(var datasms,smc,tipe,pengirim,bentuk,
                    skema,tanggal,batas,isi: string);

```

```

var pdu: string;
    p,i: integer;

```

```

begin

```

```

    pdu := datasms;
    smc := '';
    p := StrToInt('$' + copy(pdu, 1, 2)) - 1;
    pdu := copy(pdu,5,length(pdu)-4);
    for i := 1 to p do begin
        smc := smc + pdu[i*2];
        smc := smc + pdu[i*2-1];
    end;
    if smc[length(smc)] = 'F' then
        smc := copy(smc, 1, length(smc) - 1);
    pdu := copy(pdu, p*2+1,length(pdu)-p*2);

```

```

    tipe := copy(pdu, 1, 2);
    pdu := copy(pdu, 3, length(pdu)-2);

```

```

    pengirim := '';
    p := StrToInt('$'+copy(pdu,1,2));
    if p mod 2 = 1 then inc(p);
    pdu := copy(pdu,5,length(pdu)-4);
    for i := 1 to p div 2 do begin
        pengirim := pengirim + pdu[i*2];
        pengirim := pengirim + pdu[i*2-1];
    end;
    if pengirim[length(pengirim)] = 'F' then
        pengirim := copy(pengirim, 1, length(pengirim) - 1);

```

```

    pdu := copy(pdu,p+1,length(pdu)-p);

```

```

    bentuk := copy(pdu,1,2);
    pdu := copy(pdu, 3, length(pdu)-2);

```

```

    skema := copy(pdu,1,2);
    pdu := copy(pdu, 3, length(pdu)-2);

```

```

    tanggal := pdu[6]+pdu[5] + '-' + pdu[4]+pdu[3] + '-' +
        pdu[2]+pdu[1] + ' ' +

```

```

        pdu[8]+pdu[7] + ':' + pdu[10]+pdu[9] + ':' +
        pdu[12]+pdu[11];
    pdu := copy(pdu, 13, length(pdu)-12);

    batas := copy(pdu,1,2);
    pdu := copy(pdu, 3, length(pdu)-2);
    isi := PDU2Text(pdu);
end;
```

21. Kembali ke modus Form (F12). Doppel klik komponen Button5. Pada prosedur *event handler* OnClick-nya isilah dengan program berikut ini.

```

procedure TForm1.Button5Click(Sender: TObject);
var smsc, tipe, pengirim, bentuk, skema, tanggal, batas,
    isi, s : string;
begin
    s := copy(memo2.Text,1,Length(Memo2.Text)-0);
    ConvertMSG( s, smsc, tipe, pengirim, bentuk, skema,
        tanggal, batas, isi);
    Memo3.Lines.Add(tanggal+';dari '+pengirim+' : '+#13#10+ isi);
end;
```

22. Tancapkan kabel data Siemen C55 ke port serial COM1. Pasangkan pada Hp siemen yang sesuai.
23. Jalankan program anda. Klik tombol "Connect". Jika led menyala berarti port COM telah terbuka dan siap dilakukan komunikasi data.
24. Klik tombol "Kirim ke Hp". Jika muncul tulisan ATE1 dan OK dibawahnya berarti komunikasi antara PC dengan Handphone sudah terjalin dengan benar dan baik.
25. Selanjutnya anda bisa mencoba memberikan instruksi-instruksi AT yang lain seperti dalam tabel di atas (dalam Teori Singkat) dengan cara mengetikkannya di Edit1 lalu mengklik tombol "Kirim ke Hp".
26. Amati, catat dan analisislah hasil atau respon yang diberikan handphone atas instruksi-instruksi AT yang telah anda berikan tersebut.
27. Sekarang cobalah untuk membaca semua SMS yang tersimpan di handphone dengan memberikan/mengetikkan *AT Command* :
AT+CMGL=4 di Edit1
28. Kopikan salah satu data dari SMS yang terbaca dalam format PDU (blok dengan klik kiri mouse, lalu klik kanan, pilih copy), masukkan ke dalam Memo2 (klik kanan paste).
29. Klik tombol "Terjemahkan". Perhatikan, jika program anda benar maka isi SMS akan tertampil di jendela Memo3.
30. Untuk menerjemahkan SMS-SMS terbaca yang lain lakukan dengan cara yang sama seperti di atas.

Namun jangan lupa klik dulu Memo2 untuk membersihkan jendela Memo2 dan Memo3.

ADVANCE

Program B.1 di atas penerjemahan SMS terbaca dalam format PDU dilakukan secara manual satu per satu SMS (copy-paste). Ini tentu tidak efektif jika SMS yang terbaca jumlahnya banyak dan kita ingin tahu isi semua SMS. Untuk keperluan ini program diatas harus diubah sedikit sehingga penerjemahan SMS dilakukan secara menyeluruh dalam satu memo1 tersebut.

Banyak trik dapat dilakukan untuk keperluan ini, salah satunya adalah dengan mendeteksi adanya string +CMGL (hasil baca atau respon dari hp yang menunjukkan bahwa pembacaan SMS berhasil). Prinsipnya adalah dilakukan pengecekan string, mulai dari baris pertama hingga baris terakhir dalam Memo1, jika ditemukan string (sub-string) '+CMGL' maka program akan membaca string yang ada di baris dibawahnya (SMS PDU), kemudian langsung diterjemahkan dan hasil terjemahan langsung ditampilkan di Memo3.

Rutin untuk mengimplementasikan trik ini cukup sederhana, sebenarnya tinggal merubah dan memodifikasi sedikit program yang ada dalam prosedur *event handler* Button5 dalam program B1 di atas. Berikut cuplikan programnya.

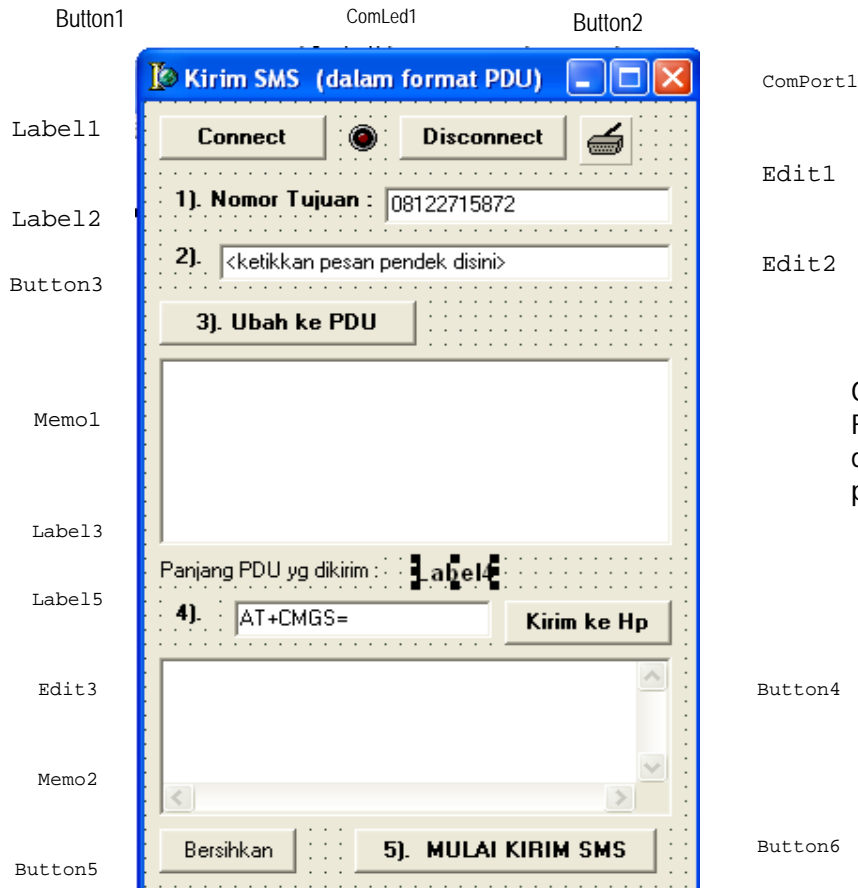
```
procedure TForm1.Button5Click(Sender: TObject);
var smsc, tipe, pengirim, bentuk, skema, tanggal, batas,
    isi, s, str : string;
    jmlbaris, i : integer;
begin
Memo3.Lines.Add('Hasil Terjemahan SMS format PDU');
Memo3.Lines.Add('====='+#13#10);
jmlbaris := 0;
for i:=1 to Memo1.Lines.Count-1 do
begin
str := Memo1.Lines[i];
str := copy(str,1,5);
if (str = '+CMGL') then
begin
inc(jmlbaris);
s := copy(memo1.Lines[i+1],1,Length(Memo1.Lines[i+1])-0);
ConvertMSG(s, smsc, tipe, pengirim, bentuk, skema,
    tanggal, batas, isi);
Memo3.Lines.Add('SMS ke:'+inttostr(jmlbaris)+' Pengirim:
    '+pengirim + '; waktu: '+tanggal+#13#10+ isi);
Memo3.Lines.Add(#13#10);
end;
end;
end;
```

B.2. Pengiriman SMS

1. Tancapkan kabel data Siemen C55 ke port serial COM1. Pasangkan pada Hp siemen yang sesuai.
2. Buka atau jalankan program Borland Delphi 7 anda, lalu buat form aplikasi baru (default).
3. Gantilah nilai Caption form tersebut dengan teks:

"Kirim SMS (dalam format PDU)"

4. Buatlah/Tambahkan ke dalam form tersebut komponen-komponen TButton (6 buah), TEdit (3 buah), TLabel (5 buah), TMemo (2 buah), TComLed (1 buah) dan TcomPort (1 buah).
5. Selanjutnya aturlah besar, kecil dan letak/posisi tiap-tiap komponen dalam form tersebut seperti dalam Gambar dibawah ini.



Gambar 9.
Form aplikasi yang
dibuat untuk
praktikum B.2.

6. Aturlah nilai properties dan prosedur event-handler OnClick untuk **ComPort1**, **Button1**, **Button2**, dan **ComLed1** sama dengan praktikum B.1.
7. Setup parameter untuk komponen ComPort1 seperti Gambar 6.
8. Buatlah prosedur *Event handler* OnRxChar dari ComPort1 seperti berikut ini:

```
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
```

```

var str : string;
begin
  Comport1.ReadStr(str,Count);
  Memo2.Text := Memo2.Text + str;
end;

```

9. Ketikkan cuplikan program berikut ini di bagian implementasi (paling atas).

```

.....
var
  Form1: TForm1;

implementation

{$R *.dfm}

//=====
// Fungsi - fungsi untuk keperluan mengkonversi suatu pesan (teks)
// SMS ke dalam format PDU yang siap untuk dikirim dengan instruksi
// AT+CMGS.
// Contoh Penggunaan:
//
// PDU := ConvertText(Simpatismsc,'','','08122715872','','','pesan');
//
// PDU adalah variabel bertipe string untuk menampung keluaran fungsi
// yang berupa string data PDU siap kirim.
//
// Sumber : Aryo Sanjaya @ BengkelProgram.Com
//=====

const
  sOK = 'OK';
  sERROR = 'ERROR';
  ProXLsmc = '62818445009';
  Simpatismc = '6281100000';
  Mentarismc = '62816124';
  IM3smc = '62855000000';

var
  lkirim : integer;

function text2PDU(text:string):string;
var PDU : string;
    geser,panjang,tmp,tmp2,tmp3,n:byte;
begin
  PDU := '';
  panjang :=length(text);
  PDU :=PDU+inttohex(panjang,2);

  geser :=0;
  for n :=1 to panjang-1 do
  begin
    tmp2 :=ord(text[n]);
    if geser<>0 then tmp2 :=tmp2 shr geser;
    tmp :=ord(text[n+1]);
    if geser=7 then
    begin
      geser :=0;
    end else
    begin

```



```

        tmp3 :=8-(geser+1);
        if tmp3<>0 then tmp:=tmp shl tmp3;
        PDU :=PDU+inttohex((tmp or tmp2),2);
        inc(geser);
    end;
end;
if geser<7 then
begin
    tmp2:=ord(text[panjang]);
    if(geser<>0)then tmp2:=tmp2 shr geser;
    PDU:=PDU+inttohex(tmp2,2);
end;
result:=PDU;
end;

```

```

function ConvertText(smsc,tipe,ref,tujuan,bentuk,skema,
    validitas,isi:string):string;

```

```

var
    PDU,tmp :string;
    p,
    i :byte;
begin
    PDU := '';
    If length(smsc)=0 then begin result :=''; exit; end;
    if length(tipe)=0 then tipe :='11';
    if length(ref)=0 then ref :='00';
    if length(bentuk)=0 then bentuk :='00';
    if length(skema)=0 then skema :='00';
    if length(validitas)=0 then validitas :='FF';
    If length(isi)=0 then begin result :=''; exit; end;

    if smsc[1]='0' then tmp :='81'+smsc else tmp :='91'+smsc;
    if(length(tmp)mod 2)<>0 then tmp :=tmp+'F';
    p := length(tmp);
    PDU :=PDU + inttohex(p div 2,2) + tmp[1] + tmp[2];
    for i:= 2 to length(tmp)div 2 do
    begin
        PDU :=PDU+tmp[i*2];
        PDU :=PDU+tmp[(i*2)-1];
    end;
    PDU :=PDU+tipe;
    PDU :=PDU+ref;
    if tujuan[1]='+' then tujuan:=copy(tujuan,2,length(tujuan)-1);
    PDU :=PDU+inttohex(length(tujuan),2);
    if(length(tujuan)mod 2)<>0 then tujuan:=tujuan+'F';
    if tujuan[1]='0' then PDU :=PDU+'81' else begin
        PDU :=PDU+'91';
    end;
    for i :=1 to length(tujuan)div 2 do
    begin
        PDU :=PDU+tujuan[i*2];
        PDU :=PDU+tujuan[(i*2)-1];
    end;
    PDU :=PDU+bentuk;
    PDU :=PDU+skema;
    PDU :=PDU+validitas;
    tmp := Text2PDU(isi);
    PDU :=PDU+tmp;
    i := length(tmp);
    lkirim := (length(PDU) - p) div 2;

```

```

    result :=PDU;
end;

```

10. Buatlah variabel global bertipe string dengan nama PDU

```

.....
    public
        { Public declarations }
    end;

var
    Form1: TForm1;
    PDU : string;

implementation

    {$R *.dfm}
.....

```

11. Double klik Button3. Isilah prosedur event handler OnClicknya dengan:

```

procedure TForm1.Button3Click(Sender: TObject);
var tujuan, isi :string;
begin
    isi := Edit3.Text;
    tujuan := Edit2.Text;
    PDU := ConvertText('Simpatismsc',',',',',tujuan,',',',',',',isi);
    PDU := PDU+#$1A;
    memo1.Text := PDU;
    Label4.Caption := inttostr(lkirim-1);
end;

```

12. Sesuaikan nomor **SMS service center** sesuai dengan provider kartu handphone anda (dalam contoh ini telkomsel).
Lihat di deklarasi Constanta di bagian implementation.

13. Double klik Button4. Isilah prosedur *event handler* OnClicknya dengan:

```

procedure TForm1.Button4Click(Sender: TObject);
var str : string;
begin
    str := Edit1.Text+#13#10;
    Comport1.WriteStr(str);
end;

```

14. Double klik komponen Button6. Isilah prosedur *event handler* OnClicknya dengan:

```

procedure TForm1.Button6Click(Sender: TObject);
var str : string;
begin
    str := PDU+#13#10;
    Comport1.WriteStr(str);

```

end;

15. Jalankan program anda. Jika belum berhasil perbaiki program anda (*debugging*) hingga berhasil dijalankan.
16. Klik tombol "Connect", pastikan led menyala.
17. Ketikkan pesan yang akan dikirim di Edit1.
18. Klik tombol Ubah ke PDU. Perhatikan berapa panjang SMS yang terhitung
19. Ketikkan:
AT+CMGS= <diisi dengan panjang PDU hasil perhitungan>
Lali klik tombol "Kirim ke Hp".
20. Klik tombol "Mulai Kirim SMS".
21. Jika muncul pesan OK di Memo2 berarti SMS terkirim.
22. Cek di hanphone tujuan.

Pertanyaan dan tugas:

1. Rekam semua hasil tiap langkah praktik diatas di lembar kerja
2. Apa fungsi dari modem itu? Jelaskan cara kerja diall-up modem!
3. Bagaimana cara komunikasi data antara PC-hanphone dapat terbangun?
4. Dalam program di atas buatlah fitur tambahan untuk membuat handphone "membuka percakapan" sendiri (otomatis) ketika ada panggilan masuk tiga kali!
Catatan: AT Command untuk membuka (Hung-up) adalah: ATA
5. Sempurnakan program diatas sehingga dapat mengintegrasikan program pengiriman dan program pembacaan SMS (dalam satu project delphi).
6. Buatlah kesimpulan dari hasil praktikum kali ini.