

LAMPIRAN A
PROGRAM PADA SISTEM PENGIRIM PESAN KODE
IDENTITAS SEL BTS (SONYERICSSON K800i)

```

import java.util.Calendar;
import java.util.Date;
import java.util.Timer;
import java.util.TimerTask;
import javax.microedition.io.Connector;
import javax.microedition.lcdui.Canvas;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Graphics;
import javax.microedition.midlet.MIDlet;
import javax.microedition.midlet.MIDletStateChangeException;
import javax.wireless.messaging.MessageConnection;
import javax.wireless.messaging.TextMessage;

public class Proj0422051 extends MIDlet {
    private String KodeIdentitasSel;
    private String NomorTelepon;
    private String SkrString = " ";
    private Timer Pewaktu;
    private ClockTimerTask PewaktuTask;
    private Date Tanggal = new Date();
    private Calendar skr = Calendar.getInstance();
    Displayable d = new ClockCanvas();
    public Boolean isPaused;

    public Proj0422051() {
        // TODO Auto-generated constructor stub
    }

    protected void destroyApp(boolean arg0) throws
MIDletStateChangeException {
        // TODO Auto-generated method stub
    }

    protected void pauseApp() {
        // TODO Auto-generated method stub
    }

    protected void startApp() throws MIDletStateChangeException
{
        // TODO Auto-generated method stub

        d.addCommand(new Command("Keluar", Command.EXIT, 0));
        d.setCommandListener(new CommandListener(){

            public void commandAction(Command c, Displayable s) {
                notifyDestroyed();
            }
        });
        skr.setTime(Tanggal);

```

```

        SkrString = skr.get(Calendar.HOUR) + " : " +
skr.get(Calendar.MINUTE) + " : " + skr.get(Calendar.SECOND);
        Pewaktu = new Timer();
        PewaktuTask = new ClockTimerTask();
        Pewaktu.scheduleAtFixedRate(PewaktuTask, 10, 180000);

        Display.getDisplay(this).setCurrent(d);

        // kirim sms
        KirimSMS();
    }

```

```

class ClockTimerTask extends TimerTask {

    public void run() {
        Tanggal = new Date();
        skr = Calendar.getInstance();
        skr.setTime(Tanggal);

        SkrString = skr.get(Calendar.HOUR) + " : " +
skr.get(Calendar.MINUTE) + " : " + skr.get(Calendar.SECOND);
        ((ClockCanvas) d).repaint();

        // kirim sms
        KirimSMS();
    }
}

```

```

class ClockCanvas extends Canvas{

    protected void paint(Graphics g) {
        // TODO Auto-generated method stub
        int width = getWidth();
        int height = getHeight();

        g.setGrayScale(255);
        g.fillRect(0, 0, width - 1, height - 1);
        g.setGrayScale(0);
        g.drawRect(0, 0, width - 1, height - 1);

        g.drawString(SkrString, 75, 120,
Graphics.TOP|Graphics.LEFT);
    }
}

```

```

public void KirimSMS() {
    KodeIdentitasSel =
System.getProperty("com.sonyericsson.net.cellid");
    // Mengirimkan SMS
    try {
        NomorTelepon = "sms://087884249449";
        MessageConnection Koneksi =
(MessageConnection) Connector.open(NomorTelepon);

```

```
        TextMessage Pesan =
(TextMessage)Koneksi.newMessage(MessageConnection.TEXT_MESSAGE);
        Pesan.setPayloadText(KodeIdentitasSel);
        Koneksi.send(Pesan);

    } catch (Exception e) {
        SkrString = "Selesai";
        // TODO: handle exception
    }
}
```

LAMPIRAN B
PROGRAM PADA SISTEM PENERIMA PESAN KODE
IDENTITAS SEL BTS

```
unit FinalProgTA0422051;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Gauges, ExtCtrls, SMSComp, Buttons, Grids, DBGrids,  
DB, DBTables, jpeg;
```

```
type
```

```
TForm1 = class(TForm)  
    Logo1: TLabel;  
    Tulisan1: TLabel;  
    Tulisan2: TLabel;  
    Tulisan3: TLabel;  
    StaticText1: TStaticText;  
    Parameter1: TGauge;  
    Parameter2: TGauge;  
    OxygenSMS1: TOxygenSMS;  
    Logo2: TLabel;  
    Logo3: TLabel;  
    KetSMS: TMemo;  
    Tulisan4: TLabel;  
    Label1: TLabel;  
    Tulisan5: TLabel;  
    Label2: TLabel;  
    TombolMaju3: TSpeedButton;  
    TombolMaju4: TSpeedButton;  
    ListSMS1: TListBox;  
    Timer1: TTimer;  
    KetSMS2: TMemo;  
    GB1: TGroupBox;  
    Tulisan6: TLabel;  
    GB2: TGroupBox;  
    Tulisan8: TLabel;  
    Isian3: TEdit;  
    DBGrid1: TDBGrid;  
    GB3: TGroupBox;  
    Tulisan7: TLabel;  
    Isian2: TEdit;  
    TombolMaju1: TButton;  
    Table1: TTable;  
    DataSource1: TDataSource;  
    TombolMaju2: TButton;  
    Image1: TImage;  
    Image2: TImage;
```

```

Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
Label19: TLabel;
Label20: TLabel;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
Label24: TLabel;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label31: TLabel;
Label32: TLabel;
Label33: TLabel;
Table2: TTable;
DataSource2: TDataSource;
DBGrid2: TDBGrid;
TdWarning1: TLabel;
Label34: TLabel;
procedure FormCreate(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure ListSMS1Click(Sender: TObject);
procedure OxygenSMS1SMSMessageReceived(Index: Integer; Time:
TDateTime;
    Text, Send: String; Pict: TBitmap);
procedure TombolMaju4Click(Sender: TObject);
procedure TombolMaju3Click(Sender: TObject);
procedure Isian3Change(Sender: TObject);
procedure TombolMaju1Click(Sender: TObject);

```

```

    procedure TombolMaju2Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation

{$R *.dfm}

{FUNGSI YANG PERTAMA}
Function BacaPesanDariLokasi(ox: TOxygenSMS; lokasi: Integer) : TStringList;

var
    PesSMS: WordBool;
    Waktu: TDateTime;
    Teks: String;
    Pengirim: String;
    Laporan: Integer;
    Gambar: TBitmap;

Begin
    Result:= TStringList.Create;
    If ox.ReadSMSMessage(lokasi, PesSMS, Waktu, Teks, Pengirim, Laporan,
    Gambar)
    Then
    Begin
        Result.Add('Lokasi Pesan ' + Format('%2.2d',[lokasi] ));
        Result.Add('Tanggal ' + FormatDateTime('dd/mm/yyyy hh:nn:ss', Waktu));
        Result.Add(' ');
        Result.Add('Isi pesan :');
        Result.Add('Kode Identitas Sel :'+ Teks);
        Result.Add(' ');
        Result.Add('Pengirim : ' + Pengirim);
    End

    Else Result.Add('Tidak ada pesan...');
End;

{FUNGSI YANG PERTAMA+++}
Function BacaPesanDariLokasixxx(ox: TOxygenSMS; lokasi: Integer) :
TStringList;

```



```

var
PesSMS: WordBool;
Waktu: TDateTime;
Teks: String;
Pengirim: String;
Laporan: Integer;
Gambar: TBitmap;

Begin
Result:= TStringList.Create;
If ox.ReadSMSMessage(lokas, PesSMS, Waktu, Teks, Pengirim, Laporan,
Gambar)
Then
Begin
Result.Add(Teks);
End

Else Result.Add("Tidak ada pesan...");
End;

{FUNGSI YANG KEDUA}
Function BacaRingkasPesan(ox: TOxygenSMS; lokasi: Integer): String;

var
PesSMS: WordBool;
Waktu: TDateTime;
Teks, Pengirim, vList: String;
Laporan: Integer;
Gambar: TBitmap;

Begin
If ox.ReadSMSMessage(lokas, PesSMS, Waktu, Teks, Pengirim, Laporan,
Gambar)
Then
Begin
vList:= '[' + Format('%2.2d', [lokasi]) + ']' + FormatDateTime('dd/mm/yyyy
hh:nn:ss', Waktu) + ' ';
if PesSMS then vList := vList + Copy(Teks,0,30) + ' '
else vList := vList + IntToStr(Laporan) + ' ';
vList := vList + Pengirim;
end
else vList := 'Tidak ada pesan...';
Result := vList;
end;

```

```

{FUNGSI YANG KETIGA}
Function BacaPesan(ox: TOxygenSMS;
var
JumHitung, JumIsi: String): TStringList;

var
SlotList: TStringList;
I, Hitung, Isi: Integer;
Locations: String;

Begin
Result:= TStringList.Create;
Screen.Cursor:= crHourGlass;
SlotList:= TStringList.Create;
ox.GetInboxInfo(Hitung, Isi, Locations);
JumHitung:= IntToStr(Hitung);
JumIsi:= IntToStr(Isi);
SlotList.CommaText:= ox.GetInboxLocations;
Screen.Cursor:= crDefault;
For I:= 0 To SlotList.Count-1 do
Result.Add(BacaRingkasPesan(ox,StrToInt(SlotList.Strings[i])));
SlotList.Free;

End;

procedure TForm1.FormCreate(Sender: TObject);

var
jumHitung, JumIsi: String;

begin
if OxygenSMS1.Open
Then StaticText1.Caption:= 'Berhasil Tersambung'
Else StaticText1.Caption:= 'Gagal Tersambung, Periksa Koneksi Kabel!';

ListSMS1.Clear;
ListSMS1.Items:= BacaPesan(OxygenSMS1, jumHitung, JumIsi);
Label1.Caption:= jumHitung;
Label2.Caption:= JumIsi;

end;

procedure TForm1.Timer1Timer(Sender: TObject);

var
JumHitung, JumIsi: String;

```

```

begin
If Not OxygenSMS1.CheckConnection
Then
Begin
Timer1.Enabled:= False;
Application.ProcessMessages;
Parameter1.Progress:= OxygenSMS1.BatteryLevel;
Parameter2.Progress:= OxygenSMS1.SignalLevel;
If OxygenSMS1.Open
Then
Begin
StaticText1.Caption:= 'Berhasil Tersambung!';
ListSMS1.Clear;
ListSMS1.Items:= BacaPesan(OxygenSMS1, JumHitung, JumIsi);
Label1.Caption:= JumHitung;
Label2.Caption:= JumIsi;
End
Else
StaticText1.Caption:= 'Gagal Tersambung, Periksa Koneksi Kabel!';
Application.ProcessMessages;
Timer1.Enabled:= True;
End

Else
Begin
Application.ProcessMessages;
StaticText1.Caption:= 'Berhasil Tersambung!';
case OxygenSMS1.BatteryLevel of
0..25: Parameter1.ForeColor:= clRed;
26..50: Parameter1.ForeColor:= clGreen;
51..75: Parameter1.ForeColor:= clPurple;
76..100: Parameter1.ForeColor:= clBlue;
End;

Parameter1.Progress:=OxygenSMS1.BatteryLevel;
Application.ProcessMessages;
case OxygenSMS1.SignalLevel of
0..25: Parameter2.ForeColor:= clRed;
26..50: Parameter2.ForeColor:= clGreen;
51..75: Parameter2.ForeColor:= clPurple;
76..100: Parameter2.ForeColor:= clBlue;
End;

Parameter2.Progress:=OxygenSMS1.SignalLevel;
Application.ProcessMessages;
End;

```

```

end;

procedure TForm1.ListSMS1Click(Sender: TObject);

begin
KetSMS.Lines:=
BacaPesanDariLokasi(OxygenSMS1,StrToInt(Copy(ListSMS1.Items.Strings[List
SMS1.itemIndex],2,2)));
KetSMS2.Lines:=
BacaPesanDariLokasixxx(OxygenSMS1,StrToInt(Copy(ListSMS1.Items.Strings[
ListSMS1.itemIndex],2,2)));

end;

procedure TForm1.OxygenSMS1SMSMessageReceived(Index: Integer;
Time: TDateTime; Text, Send: String; Pict: TBitmap);

var
Teks, JumHitung, JumIsi: String;

begin
ListSMS1.Clear;
ListSMS1.Items:= BacaPesan(OxygenSMS1, JumHitung, JumIsi);
Label1.Caption:= JumHitung;
Label2.Caption:= JumIsi;

end;

procedure TForm1.TombolMaju4Click(Sender: TObject);
begin
Close;
end;

procedure TForm1.TombolMaju3Click(Sender: TObject);
begin
OxygenSMS1.DeleteSMSMessage(ListSMS1.ItemIndex + 1);
end;

procedure TForm1.Isian3Change(Sender: TObject);
begin
Table1.FindNearest([Isian3.Text]);
end;

procedure TForm1.TombolMaju1Click(Sender: TObject);
var kode : Integer;
begin

```

```
kode:=StrToInt(Isian2.Text);
begin
Case kode of
0 : Label3.Color:=clRed;
1 : Label4.Color:=clRed;
2 : Label5.Color:=clRed;
3 : Label6.Color:=clRed;
4 : Label7.Color:=clRed;
5 : Label8.Color:=clRed;
6 : Label9.Color:=clRed;
7 : Label10.Color:=clRed;
8 : Label11.Color:=clRed;
9 : Label12.Color:=clRed;
10 : Label13.Color:=clRed;
11 : Label14.Color:=clRed;
12 : Label15.Color:=clRed;
13 : Label16.Color:=clRed;
14 : Label17.Color:=clRed;
15 : Label18.Color:=clRed;
16 : Label19.Color:=clRed;
17 : Label20.Color:=clRed;
18 : Label21.Color:=clRed;
19 : Label22.Color:=clRed;
20 : Label23.Color:=clRed;
21 : Label24.Color:=clRed;
22 : Label25.Color:=clRed;
23 : Label26.Color:=clRed;
24 : Label27.Color:=clRed;
25 : Label28.Color:=clRed;
26 : Label29.Color:=clRed;
27 : Label30.Color:=clRed;
28 : Label31.Color:=clRed;
29 : Label32.Color:=clRed;
30 : Label33.Color:=clRed;
end;
```

```
case kode of
0 : Label3.Font.Color:=clRed;
1 : Label4.Font.Color:=clRed;
2 : Label5.Font.Color:=clRed;
3 : Label6.Font.Color:=clRed;
4 : Label7.Font.Color:=clRed;
5 : Label8.Font.Color:=clRed;
6 : Label9.Font.Color:=clRed;
7 : Label10.Font.Color:=clRed;
8 : Label11.Font.Color:=clRed;
```

```
9 : Label12.Font.Color:=clRed;
10 : Label13.Font.Color:=clRed;
11 : Label14.Font.Color:=clRed;
12 : Label15.Font.Color:=clRed;
13 : Label16.Font.Color:=clRed;
14 : Label17.Font.Color:=clRed;
15 : Label18.Font.Color:=clRed;
16 : Label19.Font.Color:=clRed;
17 : Label20.Font.Color:=clRed;
18 : Label21.Font.Color:=clRed;
19 : Label22.Font.Color:=clRed;
20 : Label23.Font.Color:=clRed;
21 : Label24.Font.Color:=clRed;
22 : Label25.Font.Color:=clRed;
23 : Label26.Font.Color:=clRed;
24 : Label27.Font.Color:=clRed;
25 : Label28.Font.Color:=clRed;
26 : Label29.Font.Color:=clRed;
27 : Label30.Font.Color:=clRed;
28 : Label31.Font.Color:=clRed;
29 : Label32.Font.Color:=clRed;
30 : Label33.Font.Color:=clRed;
```

```
end;
end;
end;
```

```
procedure TForm1.TombolMaju2Click(Sender: TObject);
begin
Label3.Color:=clSilver;
Label3.Font.Color:=clAppWorkSpace;
Label4.Color:=clSilver;
Label4.Font.Color:=clAppWorkSpace;
Label5.Color:=clSilver;
Label5.Font.Color:=clAppWorkSpace;
Label6.Color:=clSilver;
Label6.Font.Color:=clAppWorkSpace;
Label7.Color:=clSilver;
Label7.Font.Color:=clAppWorkSpace;
Label8.Color:=clSilver;
Label8.Font.Color:=clAppWorkSpace;
Label9.Color:=clSilver;
Label9.Font.Color:=clAppWorkSpace;
Label10.Color:=clSilver;
Label10.Font.Color:=clAppWorkSpace;
Label11.Color:=clSilver;
```

Label11.Font.Color:=clAppWorkSpace;
Label12.Color:=clSilver;
Label12.Font.Color:=clAppWorkSpace;
Label13.Color:=clSilver;
Label13.Font.Color:=clAppWorkSpace;
Label14.Color:=clSilver;
Label14.Font.Color:=clAppWorkSpace;
Label15.Color:=clSilver;
Label15.Font.Color:=clAppWorkSpace;
Label16.Color:=clSilver;
Label16.Font.Color:=clAppWorkSpace;
Label17.Color:=clSilver;
Label17.Font.Color:=clAppWorkSpace;
Label18.Color:=clSilver;
Label18.Font.Color:=clAppWorkSpace;
Label19.Color:=clSilver;
Label19.Font.Color:=clAppWorkSpace;
Label20.Color:=clSilver;
Label20.Font.Color:=clAppWorkSpace;
Label21.Color:=clSilver;
Label21.Font.Color:=clAppWorkSpace;
Label22.Color:=clSilver;
Label22.Font.Color:=clAppWorkSpace;
Label23.Color:=clSilver;
Label23.Font.Color:=clAppWorkSpace;
Label24.Color:=clSilver;
Label24.Font.Color:=clAppWorkSpace;
Label25.Color:=clSilver;
Label25.Font.Color:=clAppWorkSpace;
Label26.Color:=clSilver;
Label26.Font.Color:=clAppWorkSpace;
Label27.Color:=clSilver;
Label27.Font.Color:=clAppWorkSpace;
Label28.Color:=clSilver;
Label28.Font.Color:=clAppWorkSpace;
Label29.Color:=clSilver;
Label29.Font.Color:=clAppWorkSpace;
Label30.Color:=clSilver;
Label30.Font.Color:=clAppWorkSpace;
Label31.Color:=clSilver;
Label31.Font.Color:=clAppWorkSpace;
Label32.Color:=clSilver;
Label32.Font.Color:=clAppWorkSpace;
Label33.Color:=clSilver;
Label33.Font.Color:=clAppWorkSpace;
end; end.

LAMPIRAN C
PROTOKOL DATA UNIT (PDU) SMS DAN
KOMUNIKASI SERIAL RS-232

PROTOKOL DATA UNIT SMS

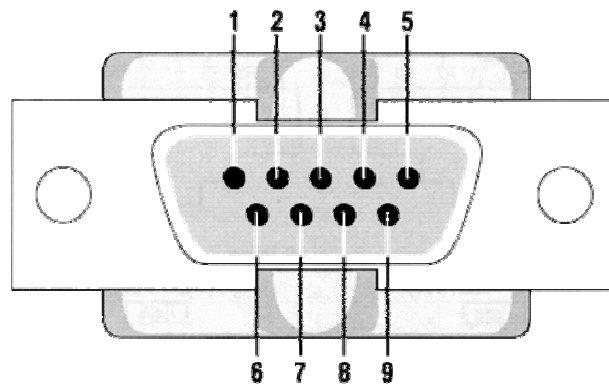
Protokol Data Unit (PDU) SMS ini berfungsi untuk mengirimkan informasi mengenai pesan SMS dari telepon seluler ke perangkat komputer untuk ditampilkan di layar monitor komputer menggunakan kabel serial RS-232. Format PDU yang dikirimkan dari telepon seluler ke perangkat komputer, sebagai berikut :

07 91 2658050000F0 04 0C 91 265836164900 0000 506020 31133180 04 C830FB0D

Dengan keterangan sebagai berikut :

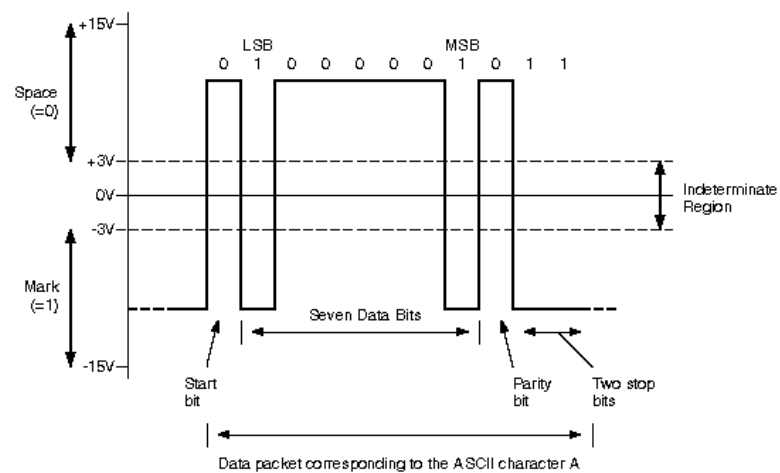
Oktet / Digit Hexa	Keterangan
07	Panjang atau jumlah pasangan digit dari nomor SMSC (service number) yang digunakan, dalam hal ini adalah 7 pasangan (14 digit berikutnya).
91	Jenis nomor SMSC. Angka 91 menandakan format nomor internasional (misal +6281xxx). Untuk 081xxx menggunakan angka 81.
2658050000F0	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'.

KOMUNIKASI SERIAL RS-232



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Satu byte dari asynchronous data :



Spesifikasi RS-232 :

SPECIFICATIONS		RS232	RS423
Mode of Operation		SINGLE -ENDED	SINGLE -ENDED
Total Number of Drivers and Receivers on One Line		1 DRIVER 1 RECVR	1 DRIVER 10 RECVR
Maximum Cable Length		50 FT.	4000 FT.
Maximum Data Rate		20kb/s	100kb/s
Maximum Driver Output Voltage		+/-25V	+/-6V
Driver Output Signal Level (Loaded Min.)	Loaded	+/-5V to +/-15V	+/-3.6V
Driver Output Signal Level (Unloaded Max)	Unloaded	+/-25V	+/-6V
Driver Load Impedance (Ohms)		3k to 7k	>=450
Max. Driver Current in High Z State	Power On	N/A	N/A
Max. Driver Current in High Z State	Power Off	+/-6mA @ +/-2v	+/-100uA
Slew Rate (Max.)		30V/Us	Adjustable
Receiver Input Voltage Range		+/-15V	+/-12V
Receiver Input Sensitivity		+/-3V	+/-200mV
Receiver Input Resistance (Ohms)		3k to 7k	4k min.

Pinnings :

RS232-C	Description	Circuit EIA	Circuit CCITT	RJ45	TIA 457
1	Shield Ground	AA			
7	Signal Ground	AB	102	4	5
2	Transmitted Data	BA	103	6	3
3	Received Data	BB	104	5	2
4	Request To Send	CA	105	8	7
5	Clear To Send	CB	106	7	8
6	DCE Ready	CC	107	1	6
20	DTE Ready	CD	108.2	3	4
22	Ring Indicator	CE	125	1	9
8	Received Line Signal Detector	CF	109	2	1

23	Data Signal Rate Select (DTE/DCE Source>	CH/CI	111/112		
24	Transmit Signal Element Timing (DTE Source)	DA	113		
15	Transmitter Signal Element Timing (DCE Source)	DB	114		
17	Receiver Signal Element Timing (DCE Source)	DD	115		
18	Local Loopback / Quality Detector	LL	141		
21	Remote Loopback	RL/CG	140/110		
14	Secondary Transmitted Data	SBA	118		
16	Secondary Received Data	SBB	119		
19	Secondary Request To Send	SCA	120		
13	Secondary Clear To Send	SCB	121		
12	Secondary Received Line Signal Detector/ Data signal Rate Select (DCE Source)	SCF/CI	122/112		
25	Test Mode	TM	142		
9	Reserved for Testing				
10	Reserved for Testing				
11	Unassigned				

Functional description :

Description	Circuit	Function
Shield Ground	AA	Also known as protective ground. This is the chassis ground connection between DTE and DCE.
Signal Ground	AB	The reference ground between a DTE and a DCE. Has the value 0 Vdc.
Transmitted Data	BA	Data send by the DTE.
Received Data	BB	Data received by the DTE.
Request To Send	CA	Originated by the DTE to initiate transmission by the

		DCE.
Clear To Send	CB	Send by the DCE as a reply on the RTS after a delay in ms, which gives the DCEs enough time to energize their circuits and synchronize on basic modulation patterns.
DCE Ready	CC	Known as DSR. Originated by the DCE indicating that it is basically operating (power on, and in functional mode).
DTE Ready	CD	Known as DTR. Originated by the DTE to instruct the DCE to setup a connection. Actually it means that the DTE is up and running and ready to communicate.
Ring Indicator	CE	A signal from the DCE to the DTE that there is an incoming call (telephone is ringing). Only used on switched circuit connections.
Received Line Signal Detector	CF	Known as DCD. A signal send from DCE to its DTE to indicate that it has received a basic carrier signal from a (remote) DCE.
Data Signal Rate Select (DTE/DCE Source>	CH/CI	A control signal that can be used to change the transmission speed.
Transmit Signal Element Timing (DTE Source)	DA	Timing signals used by the DTE for transmission, where the clock is originated by the DTE and the DCE is the slave.
Transmitter Signal Element Timing (DCE Source)	DB	Timing signals used by the DTE for transmission.
Receiver Signal Element Timing (DCE Source)	DD	Timing signals used by the DTE when receiving data.
Local Loopback / Quality Detector	LL	
Remote	RL/CG	Originated by the DCE that changes state when the

Loopback		analog signal received from the (remote) DCE becomes marginal.
Test Mode	TM	
Reserved for Testing		

LAMPIRAN D
BTS (BASE TRANSCEIVER STATION)

Kelas-kelas power BTS :

Table 2.5: BTS power classes.

BTS power class	Maximum output power	
	GSM900	DCS1800
1	320-(< 640) W	20-(< 40) W
2	160-(< 320) W	10-(< 20) W
3	80-(< 160) W	5-(< 10) W
4	40-(< 80) W	2.5-(< 5) W
5	20-(< 40) W	-
6	10-(< 20) W	-
7	5-(< 10) W	-
8	2.5-(< 5) W	-
Micro-BTS 1	(> 0.08)-0.25 W	(> 0.5)-1.6 W
Micro-BTS 2	(> 0.03)-0.08 W	(> 0.16)-0.5 W
Micro-BTS 3	(> 0.00)-0.03 W	(> 0.05)-0.16 W

Bagian-bagian GSM, BTS dapat dimasukkan pada bagian Normal Burst. Normal burst terdiri dari 26 bit urutan pelatihan (training sequences) dan sekitar 58 bit blok-blok informasi. Tiga tail bit dimasukkan di permulaan dan di akhir bagian. Total durasi bagian adalah 148 bit dengan periode (Guard Period) 8,25 bit.

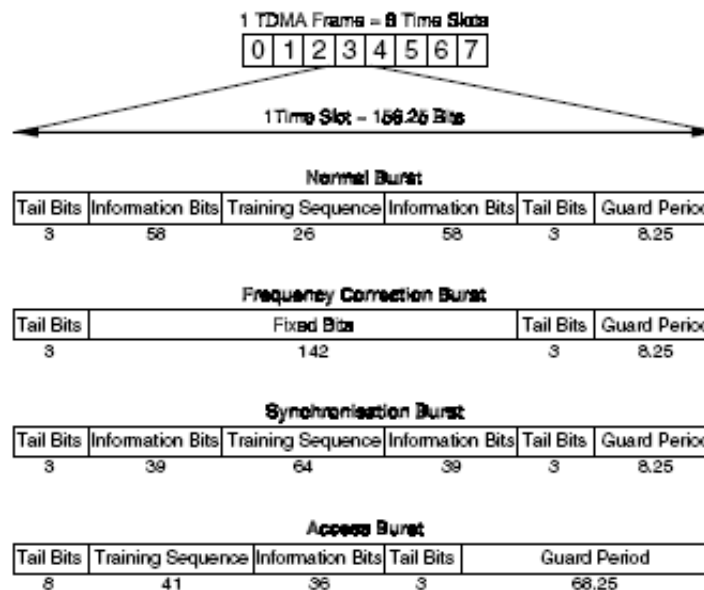


Figure 2.9: The GSM bursts.

Urutan pelatihan GSM :

Table 2.6: The GSM training sequences.

Training sequence code (TSC)	Training sequence bits (b61, b62, ..., b86)
0	(0,0,1,0,0,1,0,1,1,1,0,0,0,0,1,0,0,0,1,0,0,1,0,0,1,0,1,1,1)
1	(0,0,1,0,1,1,0,1,1,1,0,1,1,1,1,0,0,0,1,0,1,1,0,1,1,1,1)
2	(0,1,0,0,0,0,1,1,1,0,1,1,0,1,0,0,0,1,0,0,0,1,1,1,1,0)
3	(0,1,0,0,0,1,1,1,1,0,1,1,0,1,0,0,0,1,0,0,0,1,1,1,1,0)
4	(0,0,0,1,1,0,1,0,1,1,1,0,0,1,0,0,0,0,0,1,1,0,1,0,1,1)
5	(0,1,0,0,1,1,1,0,1,0,1,1,0,0,0,0,0,1,0,0,1,1,1,0,1,0)
6	(1,0,1,0,0,1,1,1,1,1,0,1,1,0,0,0,1,0,1,0,0,1,1,1,1,1)
7	(1,1,1,0,1,1,1,1,0,0,0,1,0,0,1,0,1,1,1,0,1,1,1,1,0,0)