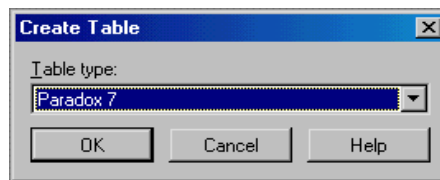


4. IMPLEMENTASI SISTEM

4.1. Pembuatan Tabel pada Database

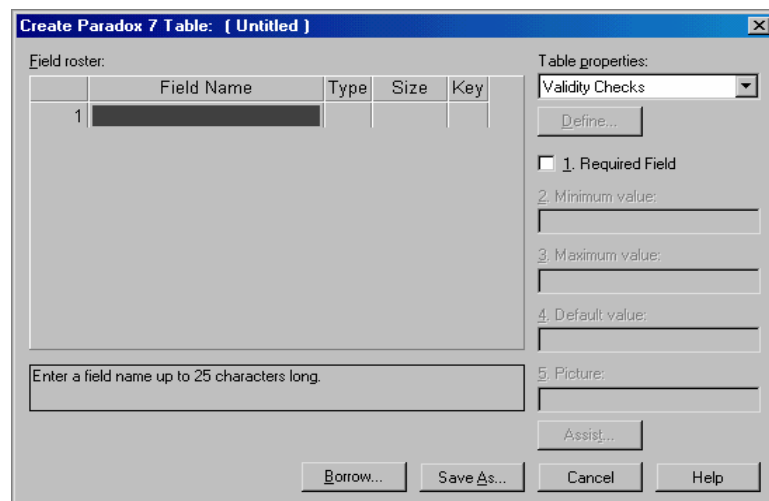
Dalam membuat simulasi *database* perusahaan *multilevel*, diperlukan enam buah tabel *database* untuk menyimpan data-data yang masuk, yaitu tabel Anggota, tabel Produk, tabel Bonus, tabel Jual, tabel Berita dan tabel SMS. Untuk itu diperlukan proses perancangan dan pembuatan tabel-tabel tersebut. Tabel-tabel tersebut dibuat dengan menggunakan Database Desktop Paradoxs 7.0. Langkah-langkah untuk membuat tabel-tabel tersebut sebagai berikut:

- a. Pada program *Database Desktop*, pilih menu *File/New/Table...* ,kemudian muncul jendela *Create Table* seperti gambar 4.1.



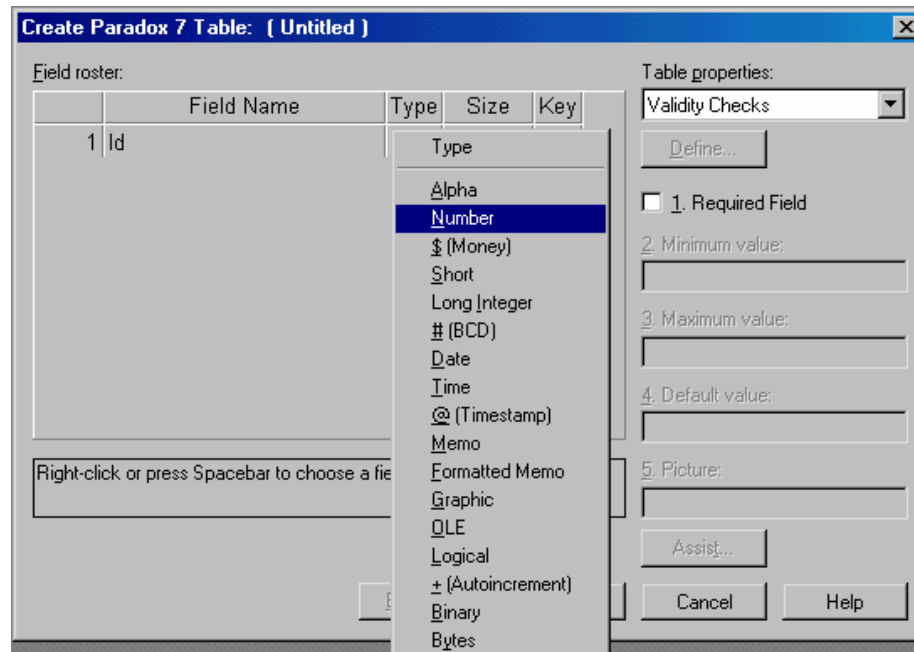
Gambar 4.1. Jendela *Create Table*

- b. Pilih *Table type*-nya *Paradox 7*, kemudian klik *OK*.
- c. Setelah itu akan muncul jendela desain tabel seperti gambar 4.2.



Gambar 4.2. Jendela Desain Tabel

Isikan *Field Name* dengan field-field tabel yang dibuat. Untuk memilih *Type field* tekan tombol *keyword* untuk tipe data yang diinginkan. Misalnya untuk tipe data *Number* maka *keyword*-nya 'N'. Bila tidak mengetahui tipe-tipe data yang ada tekan *Spacebar* sehingga akan keluar *list* dari tipe data yang disediakan. Sedangkan untuk bagian *Key* tidak perlu diisi karena dalam hal ini tidak diperlukan.



Gambar 4.3. Pemilihan Tipe Data pada Jendela Desain Tabel

- d. Setelah semua *field* beserta tipe datanya selesai dimasukkan, maka klik *Save as* untuk menyimpan tabel.

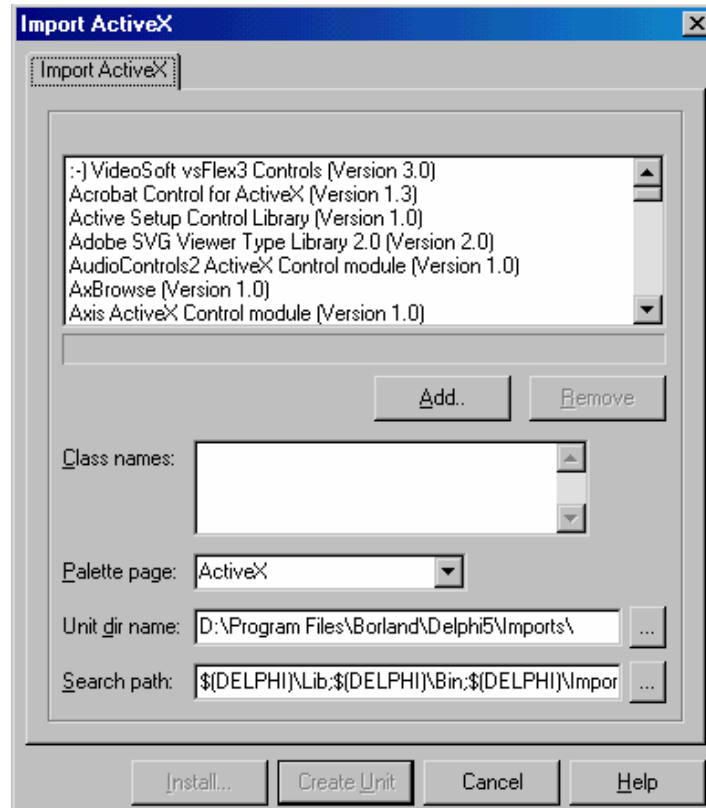
4.2. Instalasi Komponen Tambahan TMSComm

Program ini dibuat dengan menggunakan komponen tambahan yaitu TMSComm yang merupakan komponen ActiveX. Komponen tambahan ini tidak tersedia secara *default* pada *Delphi* sehingga perlu di-*install* terlebih dulu.

Komponen TMSComm berfungsi untuk mengakses *serial port* yang terhubung dengan telepon selular. *File* yang diperlukan untuk meng-*install* komponen ini adalah **MSComm32.ocx** yang merupakan *file ActiveX control*. *File*

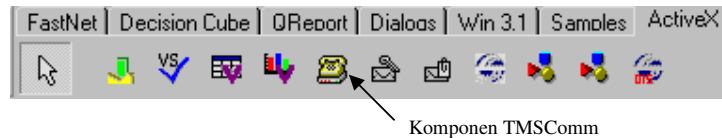
ini dapat ditemui di sub direktori *System* pada direktori *C:\Windows* atau *C:\WINNT*. Cara meng-*install* komponen ini adalah sebagai berikut:

- Pada *Delphi*, pilih menu *Component* kemudian sub menu *Import ActiveX Control*, dan akan muncul jendela dialog *Import ActiveX*.



Gambar 4.4. Jendela dialog *Import ActiveX*

- Pada *combo box* yang ada, pilih *Microsoft Comm Control 6.0*, kemudian klik tombol *Install*. Bila tidak ada pilihan tersebut, maka klik *Add* dan akan muncul jendela *Browse*.
- Arahkan ke direktori *C:\Windows\System* kemudian pilih *MSComm32.ocx* dan klik *Install*.
- Bila proses berhasil maka *class* untuk *ActiveX* ini terdaftar dengan nama *TMSComm*. Komponennya berada pada *Component Palette ActiveX*. Untuk menggunakan komponen ini cukup dengan men-*drag*-nya ke *form*.



Gambar 4.5. Komponen *TMSComm*

Beberapa *properties* yang penting dari komponen *TMSComm* ini yang harus diperhatikan agar dapat berfungsi adalah:

- *Commport*
Bagian ini berisi nomor *serial port* yang terhubung dengan telepon selular. Nilai *default*-nya adalah '1'.
- *Handshaking*
Bagian ini diisi dengan opsi nomor 3, yaitu RTSXonXoff
- *Settings*
Bagian ini berisi *baudrate*, *parity*, *data bit*, dan *stop bit* yang digunakan. *Baudrate* bergantung jenis telepon yang digunakan. Pada program ini, telepon selular yang digunakan memiliki *baudrate* 19200 bps. Untuk bagian *parity* nilai *default*-nya 'N', *data bit* nilai *default*-nya '8', *stop bit* nilai *default*-nya '1' sehingga *Settings*-nya adalah '19200,N,8,1'.

4.3. Proses Inisialisasi

Proses ini dilaksanakan sebelum sistem berjalan pada kondisi *standby*. Proses ini meliputi pengisian variabel-variabel global yang dipakai di seluruh sistem. Variabel-variabel tersebut adalah:

- ComKirim : Integer
berisi nomor *communication port* ponsel yang dipakai untuk mengirim SMS
- ComTerima : Integer
berisi nomor *communication port* ponsel yang dipakai untuk menerima SMS
- brsend : string
berisi *baudrate* ponsel yang dipakai untuk mengirim SMS, dimana ponsel yang digunakan pada program ini memiliki *baudrate* 19200 bps.
- brrcv : string
berisi *baudrate* ponsel yang dipakai untuk menerima SMS, yang nilainya sama dengan *baudrate* yang dipakai untuk mengirim SMS, yaitu 19200 bps.

- smsctr: string

berisi nomor SMS *center* operator selular yang dipakai

Variabel-variabel tersebut nilainya diperoleh dari pengaturan yang dibuat oleh *administrator* sistem atau dari *file* setting.txt.

4.4. Sistem *Standby*

Sistem ini dirancang untuk bekerja secara otomatis tanpa perlu dikontrol terus-menerus. Untuk menjalankan sistem ini hanya perlu menekan tombol 'Start', setelah itu sistem berada pada posisi *standby*, siap untuk menerima dan memproses SMS. Apabila tombol 'Start' ini ditekan maka variabel *Status* bernilai *True* sampai tombol 'Stop' ditekan.

Selama berstatus *True*, sistem memeriksa apakah ada SMS dari ponsel atau dari simulator yang masuk dan menampilkan jumlah pesan yang baru masuk tersebut. *Script* programnya adalah:

```
while statusport = true do
begin
    Berital.Enabled:=False;
    SettingSMS1.Enabled:=False;
    ter:=ReceiveSMS;
    if ter<>0 then Memo_sms.Lines.Add('Menerima 1 SMS baru');
```

Apabila tidak ada pesan baru yang masuk proses ini akan diulangi. Demikian pula apabila sistem telah selesai memproses sebuah SMS, proses ini dilakukan terus-menerus untuk menunggu SMS berikutnya selama status masih bernilai *True*.

4.5. Proses Pengolahan SMS

Sistem ini dibuat dengan metode koneksi ke telepon selular melalui komputer sehingga didalamnya harus terdapat proses-proses untuk mengolah SMS yang masuk atau yang akan dikirimkan sehingga dapat dikenali oleh telepon selular. Sistem ini juga mensyaratkan kode-kode tertentu di dalam SMS yang masuk sehingga isi SMS yang masuk perlu diperiksa terlebih dulu sebelum diproses.

4.5.1. Pengiriman dan Penerimaan SMS

Bentuk asli SMS yang digunakan sebenarnya bukan berupa teks biasa seperti yang dilihat, tetapi berupa sebuah deretan angka dan abjad yang membentuk kode yang dinamakan PDU. Untuk itu diperlukan fungsi-fungsi yang dapat mengkonversi PDU menjadi teks maupun sebaliknya. Fungsi-fungsi tersebut adalah:

- *Procedure Delay(seconds, milliSec : word);*
Prosedur ini berfungsi untuk memberikan waktu *delay*.
- *Function Pangkat(nilai, pang : integer) : LongInt;*
Fungsi ini berfungsi untuk memberikan hasil sama dengan (**nilai**^{pang})
- *Function Hexa(angka : integer) : String;*
Fungsi ini berfungsi untuk mengubah bilangan desimal(**angka**) ke *string* bilangan heksa.
- *Function UnHexa(strHex : string) : integer;*
Fungsi ini berfungsi untuk mengubah *string* bilangan heksa(**strHex**) ke bilangan desimal
- *Function flip(StrDuo : string) : string;*
Fungsi ini berfungsi untuk membalik posisi karakter dari sebuah *string*(**strDuo**) yang terdiri dari 2 buah karakter.
Contoh : *String* 'ab' dibalik menjadi 'ba'.
- *Function Biner7(angka : integer) : String;*
Fungsi ini berfungsi untuk mengubah bilangan desimal(**angka**) menjadi *string* bilangan biner 7 bit.
- *Function Biner4(angka : integer) : string;*
Fungsi ini berfungsi untuk mengubah bilangan desimal(**angka**) menjadi *string* bilangan biner 4 bit.
- *Function Unbiner7(huruf : string) : integer;*
Fungsi ini berfungsi untuk mengubah *string* bilangan biner 7 bit(**huruf**) menjadi bilangan desimal.
- *Function Unbiner4(huruf : string) : integer;*
Fungsi ini berfungsi untuk mengubah *string* bilangan biner 4 bit(**huruf**) menjadi bilangan desimal.

- *Function Tooth(karakter : char) : integer;*
Fungsi ini berfungsi untuk mengubah suatu karakter(**karakter**) menjadi bilangan ASCII. Untuk beberapa karakter tertentu oleh forum SMS sedunia, diadakan beberapa perubahan.
- *Function Untooth(pc : integer) : string;*
Fungsi ini berfungsi untuk mengubah suatu bilangan ASCII(**pc**) menjadi suatu karakter. Untuk beberapa karakter tertentu oleh forum SMS sedunia, diadakan beberapa perubahan.
- *Function EditIsi(strisi : string) : string;*
Fungsi ini berfungsi untuk mengubah *string* isi dalam teks biasa(**strisi**) menjadi *string* dalam pasangan heksa.
- *Function EditNo(strnomor : string) : string;*
Fungsi ini berfungsi untuk mengubah *string* nomor ponsel dalam teks biasa(**strnomor**) menjadi *string* dalam pasangan heksa.
- *Function FlipIsi(isi : string) : string;*
Fungsi ini berfungsi untuk mengubah *string* isi dalam pasangan heksa(**isi**) menjadi *string* dalam teks biasa.
- *Function FlipNo(strno : string) : string;*
Fungsi ini berfungsi untuk mengubah *string* nomor ponsel(**strno**) dalam pasangan heksa(**isi**) menjadi *string* dalam teks biasa.
- *Function FlipDt(strdt : string) : string;*
Fungsi ini berfungsi untuk mengubah *string* tanggal dalam pasangan heksa(**strdt**) menjadi *string* dalam teks biasa.
- *Function Str2Dt(strdt : string) : string;*
Fungsi ini berfungsi untuk mengubah format *string* tanggal yang berupa pasangan heksa(**strdt**).
- *Function lng2hexa(lngBil : Longint) : string;*
Fungsi ini berfungsi untuk mengubah bilangan desimal(**lngbil**) menjadi *string* pasangan heksa.
- *Function hexa2lng(strhex : string) : Longint;*
Fungsi ini berfungsi untuk mengubah *string* pasangan heksa(**strhex**) menjadi bilangan desimal.

Fungsi-fungsi di atas digunakan bersama dengan fungsi untuk mengirim dan menerima SMS, yaitu:

- *Function SendSMS(StrKpd,StrIsi,smscenter: string) : integer;*

Fungsi ini berfungsi untuk mengirimkan SMS ke nomor **StrKpd** yang isinya **StrIsi** melalui operator yang SMS *center*-nya **smscenter**.

Segmen Program 4.1. Proses Pengiriman SMS

```
function SendSMS(StrKpd,StrIsi,smscenter:string):integer;
var smsctr,gubah2: string;
    gabung: array[1..5] of string;
    panjang:array[1..5] of integer;
    i,j,jmlSMS,ki: integer;
    buffer : array[1..2] of string;
    arrIsi: array[1..5] of string;
    waktu: TDateTime;
begin
    //inisialisasi variabel
    SendSMS:=0;
    ki:=0;
    //insialisasi sms center
    gubah2:=EditNo(smscenter);
    //ubah ke hexa
    smsctr:=lng2hexa(length('91'+gubah2) div 2)+'91'+gubah2;
    StrKpd:=lng2hexa(length(StrKpd)+'91'+EditNo(StrKpd));
    jmlSMS:=(length(StrIsi) div 151)+1;
    for i:= 1 to jmlSMS do
    begin
        if i=jmlSMS then
            for j:= 1 to length(StrIsi)-(151*(i-1)) do
                arrIsi[i]:=arrIsi[i]+StrIsi[151*(i-1)+j]
            else
                for j:= 1 to 151 do
                    arrIsi[i]:=arrIsi[i]+StrIsi[151*(i-1)+j];
                    arrIsi[i]:=lng2hexa(length(arrIsi[i]))+EditIsi(arrIsi[i]);
                    gabung[i]:=smsctr+'0100'+StrKpd+'0000'+arrIsi[i];
                    panjang[i]:=(length(StrKpd)+length(arrIsi[i])+8) div 2;
                end;
            with Form_SMS.ComKirim do
                //inisialisasi setting
                begin
                    CommPort:=com;
                    Settings:='19200,N,8,1';
                    InputLen:=0;
                    PortOpen:=True;
                end;
                buffer[1]:='';
                waktu:=Now+EncodeTime(0,0,5,0);
                //cek koneksi ponsel
                repeat
                    Form_SMS.ComKirim.Output:='ATE1'+Chr(13);
                    repeat
                        Application.ProcessMessages;
                        buffer[1]:=buffer[1]+Form_SMS.ComKirim.Input;
```

```

        until (Pos('OK',buffer[1])<>0) or
        (Pos('ERROR',buffer[1])<>0) or (Now>waktu);
        until (Pos('OK',buffer[1])<>0) or (Pos('ERROR',buffer[1])<>0)
or (Now>waktu);
        if Pos('OK',buffer[1])<>0 then
            //proses kirim sms
            begin
                for i:= 1 to jmlSMS do
                    begin
                        buffer[1]:='';
                        buffer[2]:='';
                        repeat
Form_SMS.ComKirim.Output:='AT+CMGS='+IntToStr(panjang[i])+{'',2'
+}Chr(13);
                            repeat
                                Application.ProcessMessages;
                                buffer[1]:=buffer[1]+Form_SMS.ComKirim.Input;
                                until (Pos('>',buffer[1])<>0) or
(Pos('ERROR',buffer[1])<>0);
                                until (Pos('>',buffer[1])<>0) or
(Pos('ERROR',buffer[1])<>0);
                                if Pos('>',buffer[1])<>0 then
                                    repeat
Form_SMS.ComKirim.Output:=gabung[i]+Chr(26);
                                        repeat
                                            Application.ProcessMessages;
                                            buffer[2]:=buffer[2]+Form_SMS.ComKirim.Input;
                                            until (Pos('OK',buffer[2])<>0) or
(Pos('ERROR',buffer[1])<>0);
                                            until (Pos('OK',buffer[2])<>0) or
(Pos('ERROR',buffer[1])<>0);
                                            if Pos('OK',buffer[2])<>0 then ki:=ki+1;
                                                end;
                                            end;
                                            SendSMS:=ki;
                                            Form_SMS.ComKirim.PortOpen:=False;
                                        end;
                                    end;
                                end;
                                SendSMS:=ki;
                                Form_SMS.ComKirim.PortOpen:=False;
                            end;
                        end;
                    end;
                end;
            end;
        end;
    end;

```

- *Function ReceiveSMS : integer;*

Fungsi ini berfungsi untuk menerima SMS yang masuk ke telepon selular. Setiap pesan yang masuk ke telepon selular dipindahkan ke dalam tabel SMS yang sudah dibuat sebelumnya untuk menampung SMS-SMS yang masuk.

Setiap kali melakukan koneksi ke ponsel melalui kabel serial maka sistem mengirimkan perintah. Apabila respon dari ponsel adalah 'OK' maka proses membaca SMS dilakukan, tetapi bila respon adalah 'ERROR' maka sistem menghentikan koneksi, dan beralih dari *function* ini. Perintah tersebut tampak dalam segmen 4.2. berikut ini.

Segmen Program 4.2. Proses Pemeriksaan ke Telepon Selular

Segmen Program 4.4. Pengolahan PDU menjadi teks

```

if Pos('OK',buffer[1])<>0 then
begin
  listing[l+1]:=buffer[1];
  k:=1;
  repeat
    j:=Pos(Chr(13),listing[l+1]);
    splits[k]:=Copy(listing[l+1],1,j-1);
    listing[l+1]:=Copy(listing[l+1],j+1,length(listing[l+1])-j);
    k:=k+1;
  until j = 0;
  splits[k-1]:=listing[l+1];
  if (k-1) < 5 then te:=te
  else
  begin
    jum:=(k-5) div 2;
    te:=te+jum;
    for i:= 1 to jum do
    begin
      no[i]:=splits[i*2+1];
      pdu[i]:=splits[(i*2)+2];
    end;
    for i:= 1 to jum do
    begin
      j:=Pos(' ',no[i]);
      k:=Pos(',',no[i]);
      no[i]:=Copy(no[i],j+1,k-j-1);
    end;
    for i:= 1 to jum do
    if (Pos('40B',pdu[i])<>0) or (Pos('40C',pdu[i])<>0) then
    if Pos('0000',pdu[i])<>0 then
    repeat
      Form_SMS.ComTerima.Output:='AT+CMGD='+no[i]+Chr(13);
    repeat
      Application.ProcessMessages;
      buffer[2]:=buffer[2]+Form_SMS.ComTerima.Input;
      until (Pos('OK',buffer[2])<>0) or
        (Pos('ERROR',buffer[2])<>0);
    until (Pos('OK',buffer[2])<>0) or (Pos('ERROR',buffer[2])<>0);
    for i:= 1 to jum do
    begin
      j:=Pos('40B',pdu[i]);
      k:=Pos('40C',pdu[i]);
      if (j<>0) or (k<>0) then
      begin
        if j<>0 then pdu[i]:=copy(pdu[i],j+5,length(pdu[i])-j-4)
        else pdu[i]:=copy(pdu[i],k+5,length(pdu[i])-k-4);
        j:=Pos('0000',pdu[i]);
        if j<>0 then
        begin
          tmpDu[i,1]:=copy(pdu[i],1,j-1);
          pdu[i]:=copy(pdu[i],j+4,length(pdu[i])-j-3);
          tmpDu[i,2]:=Copy(pdu[i],1,12);
          tmpDu[i,3]:=Copy(pdu[i],15,2);
          pdu[i]:=copy(pdu[i],17,length(pdu[i])-16);
          if (length(pdu[i]) div 2) < unhexa(tmpDu[i,3]) then
          for j:=1 to unhexa(tmpDu[i,3])-(length(pdu[i]) div 2) do
            pdu[i]:=pdu[i]+'00';

```

```

        tmpDu[i, 3] := pdu[i];
        DataModule1.QuerySMS.SQL.Clear;
        DataModule1.QuerySMS.SQL.Add('insert into
        Tabel_SMS (Dari,Waktu,Isi,Status) '+'values(:from,:time,:
        cont,:rep)');
        DataModule1.QuerySMS.Prepare;
        DataModule1.QuerySMS.ParamByName('from').AsString:=
        FlipNo(tmpDu[i,1]);
        DataModule1.QuerySMS.ParamByName('time').AsString:=
        Str2Dt(FlipDt(tmpDu[i,2]));
        DataModule1.QuerySMS.ParamByName('cont').AsString:=
        FlipIsi(tmpDu[i,3]);
        DataModule1.QuerySMS.ParamByName('rep').AsInteger:=0;
        DataModule1.QuerySMS.ExecSQL;
    end;
end;
end;

```

4.5.2. Pengiriman dan Penerimaan SMS melalui Simulator

Untuk memudahkan pengujian dan efisiensi biaya dalam pembuatan sistem ini, dibuat sebuah simulator yang berfungsi layaknya telepon selular. Setiap pesan yang dikirimkan melalui simulator ini dimasukkan ke tabel SMS layaknya SMS yang diterima dari telepon selular, demikian pula pengiriman pesan diterima oleh simulator ini layaknya SMS. Proses penerimaan SMS melalui simulator ditunjukkan dalam segmen program 4.5.

Segmen Program 4.5. Penerimaan SMS melalui simulator

```

procedure TForm_Simulasi.SB_SendClick(Sender: TObject);
begin
    DataModule1.QuerySMS.SQL.Clear;
    DataModule1.QuerySMS.SQL.Add('insert into
    Tabel_SMS (Dari,Waktu,Isi,Status) '+'
    'values(:from,:time,:cont,:rep)');
    DataModule1.QuerySMS.Prepare;
    DataModule1.QuerySMS.ParamByName('from').AsString:=
    'ADMIN';
    DataModule1.QuerySMS.ParamByName('time').AsString:=
    DateTimeToStr(now);
    DataModule1.QuerySMS.ParamByName('cont').AsString:=
    edit_text.text;
    DataModule1.QuerySMS.ParamByName('rep').AsInteger:=0;
    DataModule1.QuerySMS.ExecSQL;
end;

```

Sedangkan untuk pengiriman SMS melalui simulator digabung dalam sebuah *procedure* untuk mengirim lewat telepon selular, yaitu *procedure KirimSMS(nohp,teks:string;nm:integer);*

Segmen Program 4.6. Prosedur Pengiriman SMS

```
begin
  Datamodule1.Query.Close;
  if nohp ='ADMIN' then
  begin
    totrep:=1;
    Form_Simulasi.Memo_simulasi.Lines.Clear;
    Form_Simulasi.Memo_simulasi.Lines.Add(teks);
  end else
  totrep:=SendSMS(nohp,teks,smsctr);
  Datamodule1.Query.SQL.Clear;
  Datamodule1.Query.SQL.Add('update          Tabel_SMS.db          set
    Status='+IntToStr(totrep)+' where Nomor=:n');
  Datamodule1.Query.Prepare;
  Datamodule1.Query.ParamByName('n').AsInteger:=nm;
  Datamodule1.Query.ExecSQL;
end;
```

Dalam prosedur tersebut juga dilakukan proses *update* pada tabel SMS untuk menandakan bahwa SMS tersebut sudah dibalas.

4.5.3. Pemeriksaan Isi SMS

Isi SMS yang ada di tabel SMS diambil satu per satu ketika program berjalan dan diperiksa isinya. Pemeriksaan dilakukan terhadap isi SMS dengan memperhatikan format dan kode SMS yang sudah ditentukan, yaitu BONUS <kode anggota>, DLINE <kode anggota>, HARGA <kode produk>, dan DPROD. Hal yang menjadi pedoman untuk memeriksa format SMS itu adalah:

- Untuk format SMS BONUS dan DLINE, selalu diikuti karakter ' ' (spasi) dan lima karakter kode anggota.
- Untuk format SMS HARGA, selalu diikuti karakter ' ' (spasi) dan lima karakter kode produk.
- Untuk format SMS DPROD, tidak diikuti oleh karakter apapun.

Untuk isi SMS yang sesuai format maka SMS tersebut dipecah menjadi bagian-bagian teks yaitu kode SMS dan kode anggota atau kode produk untuk proses selanjutnya. *Script* untuk memecah isi SMS tersebut adalah:

```
tr_isikode:=Copy(tr_isi,1,5);
tr_isiangg:=Copy(tr_isi,7,11);
tr_isikode:=UpperCase(tr_isikode);
tr_isiangg:=UpperCase(tr_isiangg);
```

Tetapi apabila isi SMS tidak sesuai format maka sistem akan menghentikan proses dan mengirimkan SMS balasan yang berisi pesan kesalahan.

4.6. Proses Pemeriksaan Kode Anggota dan Nomor Telepon Selular

Tahapan selanjutnya adalah pemeriksaan validitas kode anggota dan nomor ponsel. Pemeriksaan ini bertujuan untuk membandingkan nomor ponsel yang digunakan seorang anggota untuk mengirim SMS, apakah sesuai dengan nomor ponsel yang didaftarkan atas nama dirinya. Jika tidak sesuai, maka proses tidak dapat dilanjutkan. Hal ini sekaligus sebagai langkah pengaman agar informasi milik seorang anggota tidak dapat dilihat oleh anggota lain. Sebelumnya, kode anggota dan nomor ponsel telah dipecah kedalam bagian-bagian tersendiri untuk memudahkan pemeriksaan dan pengolahan data, yaitu variabel *kode_angg* dan *ponsel*. Kedua variabel ini kemudian dikirimkan sebagai parameter ke *function CekValidAnggota(oke,nohp:string): boolean;* yang memeriksa kode anggota dan nomor ponsel.

Segmen Program 4.7. *Function* CekValidAnggota

```
DataModule1.Query.Close;
DataModule1.Query.SQL.Clear;
//periksa kecocokan ponsel & kode anggota
DataModule1.Query.SQL.Add ('select kode_angg,ponsel from
tabel_anggota.db where kode_angg=:kt and ponsel=:nhp');
DataModule1.Query.Prepare;
DataModule1.Query.Parambyname('kt').AsString:=oke;
DataModule1.Query.ParamByName('nhp').AsString:=nohp;
DataModule1.Query.Open;
ada:=DataModule1.Query.fieldbyname('kode_angg').AsString;
ada2:=DataModule1.Query.fieldbyname('ponsel').AsString;
if (ada='') and (ada2='') then terdaftar:=false else
terdaftar:=true;
CekValidAnggota:=terdaftar;
```

Nilai *function CekValidAnggota* ini yang menentukan apakah sistem memproses permintaan atau tidak. Bila bernilai *true*, maka sistem memproses permintaan. Bila bernilai *false* maka sistem menampilkan pemberitahuan bahwa permintaan tidak dapat diproses.

4.7. Proses Klasifikasi Kode Permintaan

Kode permintaan yang dikenali ada empat, yaitu BONUS, DLINE, HARGA, dan DPROD. Untuk kode BONUS, DLINE, dan HARGA harus diikuti dengan kode anggota atau kode produk. Sedangkan untuk kode DPROD tidak perlu diikuti oleh kode apapun.

4.8. Proses Pengolahan Informasi Bonus

Proses pengolahan informasi bonus menggunakan kode SMS: BONUS <kode anggota>. Anggota dapat mengakses informasi bonus miliknya hanya dengan menggunakan nomor ponsel yang telah didaftarkan atas nama dirinya. Tabel yang diakses adalah tabel Bonus.

Segmen Program 4.8. Pengolahan permintaan informasi bonus

```
//jika kode=BONUS
if tr_isikode = 'BONUS' then
begin
    teks_sms:='';
    DataModule1.Query.Close;
    DataModule1.Query.SQL.Clear;
    DataModule1.Query.SQL.Add ('select bonus_kenal,bonus_jual from
tabel_bonus.db where kode_angg=:kt');
    DataModule1.Query.Prepare;
    DataModule1.Query.Params[0].AsString:=tr_isiangg;
    DataModule1.Query.Open;
    bkenal:=DataModule1.Query.FieldName('bonus_kenal').AsString;
    bjual:=DataModule1.Query.FieldName('bonus_jual').AsString;
    bttotal:=strtoint(bkenal)+strtoint(bjual);
    teks_sms:='BONUS '+tr_isiangg+'; bonus perkenalan:'+bkenal+'
poin; bonus penjualan:'+bjual+' poin; bonus
total:'+inttostr(bttotal)+' poin';
    Memo_sms.Lines.Add('---SMS sedang diproses---');
    delay (1,0);
    Memo_sms.Lines.Add('Mengirimkan SMS balasan kepada '+tr_no);
    KirimSMS(tr_no,teks_sms,tr_nm);
    Memo_sms.Lines.Add('SMS telah terkirim');
    Memo_sms.Lines.Add('');
End;
```

Output dari *function* tersebut sudah dalam format teks SMS untuk informasi bonus dan siap dikirimkan kepada anggota yang meminta data.

4.9. Proses Pengolahan Informasi Downline

Proses pengolahan informasi *downline* hampir sama dengan proses pengolahan informasi bonus. Kode SMS yang digunakan yaitu: DLINE <kode anggota>. Anggota dapat mengakses informasi *downline* miliknya hanya dengan menggunakan nomor ponsel yang telah didaftarkan atas nama dirinya. Tabel yang diakses adalah tabel Anggota.

Segmen Program 4.9. Pengolahan permintaan informasi *downline*

```
//jika kode=DLINE
if tr_isikode = 'DLINE' then
begin
    teks_sms:='';
    DataModule1.Query.Close;
    DataModule1.Query.SQL.Clear;
    DataModule1.Query.SQL.Add ('select kode_angg from
tabel_anggota.db where kode_upline=:kt');
    DataModule1.Query.Prepare;
    DataModule1.Query.Params[0].AsString:=tr_isiangg;
    DataModule1.Query.Open;
    dline:=DataModule1.Query.FieldName('kode_angg').AsString;
    //jika belum punya downline
    if dline = '' then teks_sms:='ANDA BELUM MEMILIKI DOWNLINE' else
begin
    teks_sms:='DOWNLINE '+tr_isiangg+' adalah ';
    while not Datamodule1.Query.Bof do Datamodule1.Query.Prior;
    while not Datamodule1.Query.Eof do
begin
        dline:=DataModule1.Query.FieldName('kode_angg').AsString;
        teks_sms:=teks_sms+dline+' ';
        Datamodule1.Query.next;
    end;
end;
Memo_sms.Lines.Add('---SMS sedang diproses---');
delay(1,0);
Memo_sms.Lines.Add('Mengirimkan SMS balasan kepada '+tr_no);
KirimSMS(tr_no,teks_sms,tr_nm);
Memo_sms.Lines.Add('SMS telah terkirim');
Memo_sms.Lines.Add('');
End;
```

Output dari *function* tersebut sudah dalam format teks SMS untuk informasi *downline* dan siap dikirimkan kepada anggota yang meminta data.

4.10. Proses Pengolahan Informasi Harga

Proses ini dilakukan untuk SMS yang kode permintaannya adalah HARGA <kode produk>. Untuk mengakses informasi harga, pengiriman SMS harus melalui nomor ponsel yang telah terdaftar di *database*. Tabel yang diakses adalah tabel Produk.

Segmen Program 4.10. Pengolahan permintaan informasi harga

```
//jika kode=HARGA
if tr_isikode = 'HARGA' then
begin
    //jika tidak valid
    if (not Cekvalidproduk(tr_isiangg)) then
begin
```

```

    teks_sms:='KODE PRODUK YANG ANDA MASUKKAN SALAH ATAU BELUM
TERDAFTAR';
    Memo_sms.Lines.Add('---SMS sedang diproses---');
    delay(1,0);
    Memo_sms.Lines.Add('Mengirimkan pesan error kepada '+tr_no);
    KirimSMS(tr_no,teks_sms,tr_nm);
    Memo_sms.Lines.Add('SMS telah terkirim');
    Memo_sms.Lines.Add('');
end else
begin
    teks_sms:='';
    DataModule1.Query.Close;
    DataModule1.Query.SQL.Clear;
    DataModule1.Query.SQL.Add ('select nama_produk, harga_produk
from  tabel_produk.db where kode_produk=:kt');
    DataModule1.Query.Prepare;
    DataModule1.Query.Params[0].AsString:=tr_isiangg;
    DataModule1.Query.Open;
    nprod:=DataModule1.Query.FieldName('nama_produk').AsString;
    hprod:=DataModule1.Query.FieldName('harga_produk').AsString;
    teks_sms:='HARGA PRODUK '+tr_isiangg+' '+nprod+:Rp '+hprod;
    Memo_sms.Lines.Add('---SMS sedang diproses---');
    delay(1,0);
    Memo_sms.Lines.Add('Mengirimkan SMS balasan kepada '+tr_no);
    KirimSMS(tr_no,teks_sms,tr_nm);
    Memo_sms.Lines.Add('SMS telah terkirim');
    Memo_sms.Lines.Add('');
end;
end;

```

Output dari *function* tersebut sudah dalam format teks SMS untuk informasi harga dan siap dikirimkan kepada anggota yang meminta data.

4.11. Proses Pengolahan Informasi Produk

Proses ini dilakukan untuk SMS yang kode permintaannya adalah DPROD. Untuk mengakses informasi produk, pengiriman SMS harus melalui nomor ponsel yang telah terdaftar di *database*. Tabel yang diakses adalah tabel Produk.

Segmen Program 4.11. Pengolahan permintaan informasi produk

```

//jika kode=DPROD
if tr_isikode = 'DPROD' then
begin
    teks_sms:='';
    DataModule1.Query.Close;
    DataModule1.Query.SQL.Clear;
    DataModule1.Query.SQL.Add ('select * from tabel_produk.db');
    DataModule1.Query.Prepare;
    DataModule1.Query.Open;
    if not (DataModule1.Query.RecordCount=0) then

```

```

begin
    DataModule1.Query.Last;
    i:=1;
    teks_sms:='DAFTAR 5 PRODUK TERBARU =';
    while (not DataModule1.Query.Bof) and (i<=5) do
    begin
        kprod:=
            DataModule1.Query.FieldName('kode_produk').AsString;
        nprod:=
            DataModule1.Query.FieldName('nama_produk').AsString;
        hprod:=
            DataModule1.Query.FieldName('harga_produk').AsString;
        teks_sms:=teks_sms+' '+kprod+' '+nprod+':Rp '+hprod+'';
        DataModule1.Query.Prior;
        i:=i+1;
    end;
end;
Memo_sms.Lines.Add('---SMS sedang diproses---');
delay(1,0);
Memo_sms.Lines.Add('Mengirimkan SMS balasan kepada '+tr_no);
KirimSMS(tr_no,teks_sms,tr_nm);
Memo_sms.Lines.Add('SMS telah terkirim');
Memo_sms.Lines.Add('');
end;

```

Output dari *function* tersebut sudah dalam format teks SMS untuk informasi produk dan siap dikirimkan kepada anggota yang meminta data.

4.12. Proses Pengiriman Berita

Proses ini dilakukan untuk mengirimkan SMS kepada seluruh anggota yang terdaftar. Input berita dilakukan oleh operator maksimal 160 karakter, yang kemudian dikirim dan disimpan dalam *database*. Tabel yang diakses adalah tabel Berita.

Segmen Program 4.12. Proses pengiriman berita

```

begin
    if (input_berita.Text<>'') then
    begin
        DataModule1.Query.Close;
        DataModule1.Query.SQL.Clear;
        //simpan dalam tabel berita
        DataModule1.Query.SQL.Add ('insert into tabel_berita.db values
(:prm0,:prm1)');
        DataModule1.Query.Prepare;
        DataModule1.Query.Params[0].AsString:=Input_no_berita.Text;
        DataModule1.Query.Params[1].AsString:=Input_berita.Lines.Text;
        DataModule1.Query.ExecSQL;
        DataModule1.Query.Close;
        DataModule1.QuerySMS.Open;
    end;
end;

```

```

while not DataModule1.QuerySMS.Bof do
DataModule1.QuerySMS.Prior;
while not DataModule1.QuerySMS.Eof do
begin
    nohp:=DataModule1.QuerySMS.FieldByName('Ponsel').AsString;
    //pengecekan nomor ponsel
    tno:=nohp[1]+nohp[2]+nohp[3];
    if tno='628' then
    begin
        //kirim sms
        SendSMS(nohp,input_berita.Text,smsctr);
    end;
    DataModule1.QuerySMS.Next;
end;
DataModule1.QuerySMS.Close;
MessageDlg('DATA TELAH TERKIRIM DAN
TERSIMPAN',MtInformation,[MbOk],0);
Input_berita.Clear;
sisa_karakter.Text:=('160');
Input_no_berita.Text:=Form_Menu_Utama.GenberitaCode;
Input_berita.SetFocus;
end else
begin
    MessageDlg('KOLOM INPUT BERITA HARUS DIISI',MtError,[MbOk],0);
end;
Tampil_berita;
end;

```

Output dari *procedure* tersebut sudah dalam format teks SMS dan siap dikirimkan kepada anggota yang meminta data.

4.13. Master Anggota

Seseorang yang ingin menggunakan fasilitas yang disediakan oleh sistem ini haruslah terdaftar sebagai anggota, oleh karena itu diperlukan suatu manajemen anggota sehingga hak penggunaan fasilitas ini dapat dikontrol.

4.13.1. Pendaftaran Anggota Baru

Untuk mendaftar menjadi anggota, harus melalui operator, dimana operator mencatat data-data calon anggota. Sistem akan memeriksa apakah data yang dimasukkan valid atau tidak. Bila valid, maka pendaftaran diterima, dan anggota tersebut akan memiliki kode anggota. Data-data yang harus dimasukkan antara lain: nama, alamat, nomor ponsel, dan kode upline, yaitu kode anggota yang merekrut calon anggota yang bersangkutan. Dengan demikian, anggota yang merekrut mendapatkan bonus pengenalan sebanyak 50 poin. Data-data tersebut kemudian disimpan ke tabel Anggota dan tabel Bonus.

Segmen Program 4.13. Proses pendaftaran anggota baru

```

begin
    //cek validitas data
    if CekInputValid and cekponselValid(input_ponsel.text) and
    Cekupline then
        begin
            DataModule1.Query.Close;
            DataModule1.Query.SQL.Clear;
            //simpan dalam tabel anggota
            DataModule1.Query.SQL.Add ('insert into tabel_anggota.db
values (:prm0,:prm1,:prm2,:prm3,:prm4,:prm5)');
            DataModule1.Query.Prepare;
            DataModule1.Query.Params[0].AsString:=Input_Kode.Text;
            DataModule1.Query.Params[1].AsString:=Input_Nama.Text;
            DataModule1.Query.Params[2].AsString:=Input_Alamat.Lines.Text;
            DataModule1.Query.Params[3].AsString:=Input_ponsel.Text;
            DataModule1.Query.Params[4].AsString:=
uppercase(Input_upline.Text);
            DataModule1.Query.Params[5].AsString:=Input_Ket.Lines.Text;
            DataModule1.Query.ExecSQL;
            DataModule1.Query.Close;
            DataModule1.Query.SQL.Clear;
            //input penambahan bonus pengenalan
            DataModule1.Query.SQL.Add ('insert into tabel_bonus.db values
(:prm0,:prm1,:prm2)');
            DataModule1.Query.Prepare;
            DataModule1.Query.Params[0].AsString:=Input_Kode.Text;
            DataModule1.Query.Params[1].AsInteger:=0;
            DataModule1.Query.Params[2].AsInteger:=0;
            DataModule1.Query.ExecSQL;
            DataModule1.Query.Close;
            DataModule1.Query.SQL.Clear;
            DataModule1.Query.SQL.Add ('select bonus_kenal from
tabel_bonus.db where kode_angg=:prm0');
            DataModule1.Query.Prepare;
            DataModule1.Query.Params[0].AsString:=Input_upline.Text;
            DataModule1.Query.Open;
            bonus_tambah:=
            DataModule1.Query.FieldByName('bonus_kenal').asinteger;
            bonus_tambah:=bonus_tambah+50;
            DataModule1.Query.SQL.Clear;
            DataModule1.Query.SQL.Add ('update tabel_bonus.db set
bonus_kenal=:prm0 where kode_angg=:prm1');
            DataModule1.Query.Prepare;
            DataModule1.Query.Params[0].AsInteger:=bonus_tambah;
            DataModule1.Query.Params[1].AsString:=Input_upline.Text;
            DataModule1.Query.ExecSQL;
            DataModule1.Query.Close;
            MessageDlg('DATA TELAH TERSIMPAN',MtInformation,[MbOk],0);
            Input_Clear;
            Input_Kode.Text:=Form_Menu_Utama.GenAnggCode;
            Input_Nama.SetFocus;
        end else
        begin
            MessageDlg('KOLOM NAMA, PONSEL DAN KODE UPLINE HARUS DIISI
ATAU PERHATIKAN KODE UPLINE DAN NOMOR PONSEL
ANDA',MtError,[MbOk],0);
        end;
    end;
end;

```

4.13.2. Perubahan Data Anggota

Setiap anggota bisa melakukan perubahan pada data pribadinya. Tetapi data yang bisa diubah terbatas. Anggota hanya bisa merubah nama, alamat, dan nomor ponsel.

Segmen Program 4.14. Proses perubahan data anggota

```
begin
  //cek validitas data
  if cekponselvalid(ubah_ponsel.Text) and cekinputvalid then
    begin
      //update data pada tabel anggota
      DataModule1.Query.Close;
      DataModule1.Query.SQL.Clear;
      DataModule1.Query.SQL.Add ('update tabel_anggota.db');
      DataModule1.Query.SQL.Add ('set
nama_angg=:prm0,alamat_angg=:prm1,ponsel=:prm2,kode_upline=:prm3,k
et=:prm4');
      DataModule1.Query.SQL.Add ('where kode_angg=:prm5');
      DataModule1.Query.Prepare;
      DataModule1.Query.Params[0].AsString:=Ubah_Nama.Text;
      DataModule1.Query.Params[1].AsMemo:=Ubah_Alamat.Lines.Text;
      DataModule1.Query.Params[2].AsString:=Ubah_ponsel.Text;
      DataModule1.Query.Params[3].AsString:=Ubah_upline.Text;
      DataModule1.Query.Params[4].AsMemo:=Ubah_Ket.Lines.Text;
      DataModule1.Query.Params[5].AsString:=Ubah_Kode.Text;
      DataModule1.Query.ExecSQL;
      DataModule1.Query.Close;
      Form_Anggota.SG_Cari.Cells[1,Tag]:=Ubah_Nama.Text;
      Form_Anggota.SG_Cari.Cells[2,Tag]:=Ubah_alamat.Lines.Text;
      Form_Anggota.SG_Cari.Cells[3,Tag]:=Ubah_Ponsel.Text;
      Form_Anggota.SG_Cari.Cells[4,Tag]:=Ubah_upline.Text;
      MessageDlg('UBAH DATA SELESAI',MtInformation,[MbOk],0);
      Form_Ubah_angg.Close;
    end else
    begin
      MessageDlg('KOLOM NAMA DAN PONSEL HARUS DIISI ATAU PERHATIKAN
NOMOR PONSEL ANDA',MtError,[MbOk],0);
    end;
  end;
end;
```

4.14. Master Produk

Kegiatan utama sebuah perusahaan *multilevel* adalah menjual produknya untuk memperoleh laba. Jumlah produk-produk yang ditawarkan tentu tidak sedikit. Selain itu, dalam jangka waktu tertentu perusahaan pasti mendatangkan jenis produk yang baru, oleh karena itu diperlukan suatu manajemen produk sehingga dapat dikontrol.

4.14.1. Pencatatan Produk Baru

Semua produk yang dijual oleh perusahaan, dicatat datanya oleh operator dan disimpan di dalam tabel Produk. Data-data yang disimpan meliputi kode produk, nama produk, dan harga produk.

Segmen Program 4.15. Proses pencatatan produk baru

```
begin
  //cek validitas data
  if CekInputValid then
    begin
      //simpan dalam tabel produk
      DataModule1.Query.Close;
      DataModule1.Query.SQL.Clear;
      DataModule1.Query.SQL.Add ('insert into tabel_produk.db values
(:prm0,:prm1,:prm2,:prm3)');
      DataModule1.Query.Prepare;
      DataModule1.Query.Params[0].AsString:=Input_Kode.Text;
      DataModule1.Query.Params[1].AsString:=Input_Nama.Text;
      DataModule1.Query.Params[2].AsString:=Input_harga.Text;
      DataModule1.Query.Params[3].AsString:=Input_Ket.Lines.Text;
      DataModule1.Query.ExecSQL;
      DataModule1.Query.Close;
      MessageDlg('DATA TELAH TERSIMPAN',MtInformation,[MbOk],0);
      Input_Clear;
      Input_Kode.Text:=Form_Menu_Utama.GenProdCode;
      Input_Nama.SetFocus;
    end else
    begin
      MessageDlg('KOLOM NAMA DAN HARGA HARUS
DIISI',MtError,[MbOk],0);
    end;
  end;
end;
```

4.14.2. Perubahan Data Produk

Data-data produk memungkinkan untuk berubah, sehingga disediakan fasilitas untuk merubah data produk. Tetapi data yang bisa diubah terbatas, yaitu nama produk dan harga produk.

Segmen Program 4.16. Proses perubahan data produk

```
begin
  if (Ubah_Nama.Text<>'') and (Ubah_harga.Text<>'') then
    begin
      //update data pada tabel produk
      DataModule1.Query.Close;
      DataModule1.Query.SQL.Clear;
      DataModule1.Query.SQL.Add ('update tabel_produk.db');
      DataModule1.Query.SQL.Add ('set nama_produk=:
prm0,harga_produk=:prm1,ket=:prm2');
    end;
  end;
```

```

DataModule1.Query.SQL.Add ('where kode_produk=:prm3');
DataModule1.Query.Prepare;
DataModule1.Query.Params[0].AsString:=Ubah_Nama.Text;
DataModule1.Query.Params[1].AsString:=Ubah_harga.Text;
DataModule1.Query.Params[2].AsMemo:=Ubah_Ket.Lines.Text;
DataModule1.Query.Params[3].AsString:=Ubah_Kode.Text;
DataModule1.Query.ExecSQL;
DataModule1.Query.Close;
Form_Produk.SG_Cari.Cells[1,Tag]:=Ubah_Nama.Text;
Form_Produk.SG_Cari.Cells[2,Tag]:=Ubah_harga.Text;
MessageDlg('UBAH DATA SELESAI',MtInformation,[MbOk],0);
Form_Ubah_prod.Close;
end else
begin
    MessageDlg('KOLOM          NAMA          DAN          HARGA          HARUS
DIISI',MtError,[MbOk],0);
end;
end;

```

4.15. Proses Transaksi Penjualan

Seperti yang telah dijelaskan pada sub bab 4.14, bahwa kegiatan utama dari sebuah perusahaan *multilevel* adalah menjual produk. Oleh karena itu, setiap transaksi penjualan yang dilakukan anggota harus dicatat. Pencatatan transaksi penjualan ini juga bertujuan untuk perhitungan bonus penjualan setiap anggota. Setiap anggota memperoleh bonus sebesar 1 poin untuk setiap kelipatan Rp 10.000 dari total penjualan dalam sekali transaksi. Setiap transaksi disimpan dalam tabel Jual.

Segmen Program 4.17. Proses transaksi penjualan

```

begin
    //cek validitas inputan
    if CekInputValid then
        begin
            //simpan dalam tabel jual
            DataModule1.Query.Close;
            DataModule1.Query.SQL.Clear;
            DataModule1.Query.SQL.Add ('insert into tabel_jual.db values
(:prm0,:prm1,:prm2,:prm3)');
            DataModule1.Query.Prepare;
            DataModule1.Query.Params[0].AsString:=Input_no_nota.Text;
            DataModule1.Query.Params[1].AsString:=CB_Kode_Angg.Text;
            DataModule1.Query.Params[2].AsString:=Input_Nama_angg.Text;
            DataModule1.Query.Params[3].AsInteger:=
StrToInt(Total_jual.Text);
            DataModule1.Query.ExecSQL;
            DataModule1.Query.Close;
            DataModule1.Query.SQL.Clear;
            //penambahan bonus penjualan
            DataModule1.Query.SQL.Add ('select          bonus_jual          from
tabel_bonus.db where kode_angg=:prm0');

```

```

DataModule1.Query.Prepare;
DataModule1.Query.Params[0].AsString:=CB_kode_angg.Text;
DataModule1.Query.Open;
bonus_tambah:=
DataModule1.Query.FieldByName('bonus_jual').asinteger;
poin:=(strtoint(total_jual.Text) div 10000);
bonus_tambah:=bonus_tambah+poin;
DataModule1.Query.SQL.Clear;
DataModule1.Query.SQL.Add      ('update      tabel_bonus.db      set
bonus_jual=:prm0 where kode_angg=:prm1');
DataModule1.Query.Prepare;
DataModule1.Query.Params[0].AsInteger:=bonus_tambah;
DataModule1.Query.Params[1].AsString:=CB_kode_angg.Text;
DataModule1.Query.ExecSQL;
DataModule1.Query.Close;
MessageDlg('TRANSAKSI SELESAI ',MtInformation,[MbOk],0);
SG_Daftar_jual_Clear;
Input_No_Nota.Text:=Form_Menu_Utama.GenNotaCode;
CB_kode_angg.ItemIndex:=0;
CB_Kode_Angg.SetFocus;
Input_nama_angg.Text:=Get_Nama_Angg(CB_Kode_angg.Text);
end
end;

```