LAMPIRAN A

LISTING PROGRAM

```
unit UMenuUtama;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs,   Menus,   ImgList,
ComCtrls,   ExtCtrls,   jpeg,
StdCtrls;

type
  TFUtama = class(TForm)
    MenuUtama: TMainMenu;
    MenuUser: TMenuItem;
    ImageList: TImageList;
    MenuLogin: TMenuItem;
    MenuLogOut: TMenuItem;
    Separator: TMenuItem;
    MenuKeluar: TMenuItem;
    MenuDataMaster: TMenuItem;
    MenuDataPasien: TMenuItem;
    MenuDataJenisMakanan:
TMenuItem;
    MenuDataBahanMakanan:
TMenuItem;
    MenuDetailBahanMakanan:
TMenuItem;
    MenuDataMakanan:
TMenuItem;
    MenuDataRuangan:
TMenuItem;
    N1: TMenuItem;
    MenuDataPegawai:
TMenuItem;
    MenuDataKelas: TMenuItem;
    MenuDataRuangRawat:
TMenuItem;
    MenuDataUser: TMenuItem;
    MenuInstalasiGizi:
TMenuItem;
    MenuLaporan: TMenuItem;
    MenuDataIRI: TMenuItem;
    PermintaanMakanan1:
TMenuItem;
    StatusBar: TStatusBar;
    LaporanDataPasien1:
TMenuItem;
    LaporanDataRawatInap1:
TMenuItem;

LaporanDataPermintaanMakanan1:
TMenuItem;
    Timer: TTimer;
    MenuProfileUser:
TMenuItem;
    N2: TMenuItem;
    DataResourceMakanan1:
TMenuItem;
    DataMakanan1: TMenuItem;
    N3: TMenuItem;
    DataJenisMakanan1:
TMenuItem;
    DataBahanMakanan1:
TMenuItem;
    DataDetailBahanMakanan1:
TMenuItem;
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    procedure
MenuDataPasienClick(Sender:
TObject);
    procedure
MenuLoginClick(Sender:
TObject);
    procedure
MenuLogOutClick(Sender:
TObject);
    procedure
FormCloseQuery(Sender:
TObject;     var     CanClose:
Boolean);
    procedure
MenuKeluarClick(Sender:
TObject);
    procedure
MenuDataJenisMakananClick(Send
er: TObject);
    procedure
MenuDataBahanMakananClick(Send
er: TObject);
```

```
    procedure
MenuDataMakananClick(Sender:
TObject);
    procedure
MenuDetailBahanMakananClick(Se
nder: TObject);
    procedure
MenuDataKelasClick(Sender:
TObject);
    procedure
MenuDataRuangRawatClick(Sender
: TObject);
    procedure
MenuDataPegawaiClick(Sender:
TObject);
    procedure
MenuDataUserClick(Sender:
TObject);
    procedure
MenuDataIRIClick(Sender:
TObject);
    procedure
PermintaanMakanan1Click(Sender
: TObject);
    procedure
TimerTimer(Sender: TObject);
    procedure
MenuProfileUserClick(Sender:
TObject);
    procedure
LaporanDataPasien1Click(Sender
: TObject);
    procedure
LaporanDataRawatInap1Click(Sen
der: TObject);
    procedure
LaporanDataPermintaanMakanan1C
lick(Sender: TObject);
    procedure
DataDetailBahanMakanan1Click(S
ender: TObject);
    procedure
DataMakanan1Click(Sender:
TObject);
    procedure
DataBahanMakanan1Click(Sender:
TObject);
    procedure
DataJenisMakanan1Click(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FUtama: TFUtama;

implementation

uses UDataPasien, ULoginUser,
UDataJenisMakanan, UDataBahan,
UDataMakanan,
  UDataDetaiBahanMakanan,
UDataKelas,        UDataRuang,
UDataPegawai, UDataUser,
  UDataIRI,
UDataPermintaanMakanan,
UProfileUser, UPasienReport,
  FLapPasien,      UIRIReport,
UPermintaanMakananReport;

{$R *.dfm}

procedure
TFUtama.MenuDataPasienClick(Se
nder: TObject);
begin
  FDataPasien.ShowModal;
end;

procedure
TFUtama.MenuLoginClick(Sender:
TObject);
begin
  FLoginUser.ShowModal;
end;

procedure
TFUtama.MenuLogOutClick(Sender
: TObject);
begin
  If  MessageDlg('Apakah  Anda
Yakin   Ingin   Log-Out   ?',
```

```
mtConfirmation,
[mbYes,mbNo],0) = mrYes Then
    begin
      MenuLogin.Visible     :=
True;
      MenuLogOut.Visible    :=
False;
      MenuDataMaster.Visible
:= False;
      MenuProfileUser.Visible
:= False;

MenuInstalasiGizi.Visible   :=
False;
      MenuLaporan.Visible   :=
False;
      StatusBar.Panels[1].Text
:=  'Status  Login  :  '  +
FLoginUser.vNIP   +   '-'   +
FLoginUser.vNamaPegawai  +  '
Telah LogOut';
      FLoginUser.vNIP :='';
      FLoginUser.vNamaPegawai
:= '';
      FLoginUser.vHakAkses  :=
'';
    end;
end;

procedure
TFUtama.FormCloseQuery(Sender:
TObject;     var     CanClose:
Boolean);
begin
  If  MessageDlg('Apakah  Anda
Yakin   Ingin   Keluar   ?',
mtConfirmation,
[mbYes,mbNo],0) = mrYes Then
    begin
      CanClose := True;
      Application.Terminate;
    end
  Else
    CanClose := False;
end;

procedure
TFUtama.MenuKeluarClick(Sender
: TObject);
begin
  Close;
end;

procedure
TFUtama.MenuDataJenisMakananCl
ick(Sender: TObject);
begin
  FDataJenisMakanan.ShowModal;
end;

procedure
TFUtama.MenuDataBahanMakananCl
ick(Sender: TObject);
begin
  FDataBahan.ShowModal;
end;

procedure
TFUtama.MenuDataMakananClick(S
ender: TObject);
begin
  FDataMakanan.ShowModal;
end;

procedure
TFUtama.MenuDetailBahanMakanan
Click(Sender: TObject);
begin

FDataDetailBahanMakanan.ShowMo
dal;
end;

procedure
TFUtama.MenuDataKelasClick(Sen
der: TObject);
begin
  FDataKelas.ShowModal;
end;

procedure
TFUtama.MenuDataRuangRawatClic
k(Sender: TObject);
begin
```

```
  FDataRuang.ShowModal;
end;

procedure
TFUtama.MenuDataPegawaiClick(S
ender: TObject);
begin
  FDataPegawai.ShowModal
end;

procedure
TFUtama.MenuDataUserClick(Send
er: TObject);
begin
  FDataUser.ShowModal;
end;

procedure
TFUtama.MenuDataIRIClick(Sende
r: TObject);
begin
  FDataIRI.ShowModal;
end;

procedure
TFUtama.PermintaanMakanan1Clic
k(Sender: TObject);
begin

FDataPermintaanMakanan.ShowMod
al;
end;

procedure
TFUtama.TimerTimer(Sender:
TObject);
begin
  StatusBar.Panels[0].Text  :=
FormatDateTime('dd-mm-yyyy   ,
Pukul hh:mm:ss', Now);
end;

procedure
TFUtama.MenuProfileUserClick(S
ender: TObject);
begin
  FProfileUser.ShowModal;
end;

procedure
TFUtama.LaporanDataPasien1Clic
k(Sender: TObject);
begin
  ReportPasien.Preview;
end;

procedure
TFUtama.LaporanDataRawatInap1C
lick(Sender: TObject);
begin
  ReportIRI.Preview;
end;

procedure
TFUtama.LaporanDataPermintaanM
akanan1Click(Sender: TObject);
begin

ReportPermintaanMakanan.Previe
w;
end;

procedure
TFUtama.DataDetailBahanMakanan
1Click(Sender: TObject);
begin

FDataDetailBahanMakanan.ShowMo
dal;
end;

procedure
TFUtama.DataMakanan1Click(Send
er: TObject);
begin
  FDataMakanan.ShowModal;
end;

procedure
TFUtama.DataBahanMakanan1Click
(Sender: TObject);
begin
  FDataBahan.ShowModal;
end;
```

```
procedure
TFUtama.DataJenisMakanan1Click
(Sender: TObject);
begin
FDataJenisMakanan.ShowModal;
end;

end.

unit USys;

interface

uses
  Classes, SysUtils, TypInfo,
StrUtils ,ADODB, Dialogs, DB;
type
  TStatusConnection          =
(Connected, NotConnected);
  TTipeQuery    =    (Actived,
Executed);
  TStatusQuery    =    (Valid,
NotValid);
type
  TSys = class
  private
    FConnect: Boolean;
    FPort: integer;
    FPassword: String;
    FProtocol: String;
    FDBName: String;
    FConnectionString: String;
    FUser: String;
    FHostName: String;
    FSysConnection:
TADOConnection;
    FQueryConnection:
TADOQuery;
    FStatusConnection:
TStatusConnection;
    FStatusQuery:
TStatusQuery;
    procedure SetConnect(const
Value: Boolean);
    procedure
SetConnectionString(const
Value: String);
    procedure  SetDBName(const
Value: String);
    procedure
SetHostName(const       Value:
String);
    procedure
SetPassword(const       Value:
String);
    procedure    SetPort(const
Value: integer);
    procedure
SetProtocol(const       Value:
String);
    procedure
SetStatusConnection(const
Value: TStatusConnection);
    procedure
SetSysConnection(const  Value:
TADOConnection);
    procedure    SetUser(const
Value: String);
    procedure
SetStatusQuery(const    Value:
TStatusQuery);

  published
    property HostName : String
read      FHostName      write
SetHostName;
    property Password : String
read      FPassword      write
SetPassword;
    property  User  :  String
read FUser write SetUser;
    property Protocol : String
read      FProtocol      write
SetProtocol;
    property  DBName  :  String
read FDBName write SetDBName;
    property  Port  :  integer
read FPort write SetPort;
    property  ConnectionString
:         String         read
FConnectionString        write
SetConnectionString;
    property  SysConnection  :
TADOConnection            read
```

```
FSysConnection           write
SetSysConnection;
    property   QueryConnection
:        TADOQuery        read
FQueryConnection;
    property Connect : Boolean
read      FConnect      write
SetConnect;
    property  StatusConnection
:    TStatusConnection    read
FStatusConnection        write
SetStatusConnection;
    property   StatusQuery   :
TStatusQuery read FStatusQuery
write SetStatusQuery;



    { PROSEDUR DAN FUNGSI PADA
KELAS SYS }
    function
DBAddLessString(ASource      :
String) : String;
    function
DBAddLessNumeric(ASource
:String) : String;
    function
DBCheckNumeric(APattern      :
char) : Boolean;
    function  Query(ASyntax  :
String;      ATipeQuery      :
TTipeQuery) : TADOQuery;
    function
GetDataSource(AQuery         :
TADOQuery) : TDataSource;

    procedure    Log(AUser    :
String;      AUserAction     :
WideString);
    procedure  CallMethod(AObj
:   TObject;   AMethodName   :
String);

    constructor Create;
    destructor        Destroy;
override;
end;

type
  TExec = procedure of object;

const
  CEsc = '''';
implementation

uses UPegawai, ULoginUser;

{ TSys }

procedure
TSys.CallMethod(AObj: TObject;
AMethodName: String);
var
  vObj : TObject;
  vRoutine : TMethod;
  vExec : TExec;
begin
  vObj := AObj;
  vRoutine.Data             :=
Pointer(vObj);
  vRoutine.Code             :=
vObj.MethodAddress(AMethodName
);
  vExec := TExec(vRoutine);
  vExec;
end;

constructor TSys.Create;
begin
  try
    { KONEKSI MENGGUNAKAN ADO
CONNECTION     PADA     MYSQL
DATABASE}
    HostName := 'localhost';
    User := 'root';
    Password := '';
    DBName := 'sigrs';
    Port := 3306;
    Connect := True;
    SysConnection           :=
TADOConnection.Create(Nil);
    SysConnection.LoginPrompt
:= False;

SysConnection.ConnectionString
:= 'Provider=MSDASQL.1;Persist
```

```
Security       Info=False;Data
Source=MySQLDatabase;';
    SysConnection.Connected :=
True;

SysConnection.KeepConnection
:= True;
    StatusConnection        :=
Connected;
  except
    on E : Exception do
      begin
        StatusConnection    :=
NotConnected;
        ShowMessage(E.Message)
;
      end;
  end;
end;

function
TSys.DBAddLessNumeric(ASource:
String): String;
begin
  if  (Length(ASource)  =  0)
then
    begin
      Result := '0';
    end
  else
    begin
      Result := ASource;
    end;
end;

function
TSys.DBAddLessString(ASource:
String): String;
begin
  if  (Length(ASource)  =  0)
then
    begin
      Result := '' ;
    end
  else
    begin
      ASource               :=
StringReplace(ASource,    '\',
'\\',
[rfReplaceAll,rfIgnoreCase]);
      ASource               :=
StringReplace(ASource,   '''',
'\''',          [rfReplaceAll,
rfIgnoreCase]);
      Result := ASource;
    end;
end;

function
TSys.DBCheckNumeric(APattern:
char): Boolean;
begin
  case APattern of
    '0' : Result := True;
    '1' : Result := True;
    '2' : Result := True;
    '3' : Result := True;
    '4' : Result := True;
    '5' : Result := True;
    '6' : Result := True;
    '7' : Result := True;
    '8' : Result := True;
    '9' : Result := True;
    '.' : Result := True;
    #8   :   Result   :=   True
//NILAI UNTUK BACKSPACE
  else
    Result := False;
    Beep;
  end;
end;

destructor TSys.Destroy;
begin
  inherited Destroy;
end;

function
TSys.GetDataSource(AQuery:
TADOQuery): TDataSource;
var
  vDataSource : TDataSource;
begin
  try
    vDataSource             :=
TDataSource.Create(Nil);
```

```
    vDataSource.DataSet     :=
AQuery;
    Result := vDataSource;
  except
    on E : Exception do
      begin

ShowMessage(E.Message);
        Result := Nil;
      end;
  end;
end;

procedure      TSys.Log(AUser:
String;           AUserAction:
WideString);
var
  vQuery : TADOQuery;
  vSyntax, vAction : String;
begin
  try
    vAction                 :=
StringReplace(AUserAction,
'''',                       '',
[rfReplaceAll,rfIgnoreCase]);
    vAction                 :=
StringReplace(vAction   ,   ';'
,'',            [rfReplaceAll,
rfIgnoreCase]);
    vSyntax  :=  'INSERT  INTO
LOG      (Waktu,      UserLog,
ActionLog) VALUES (Now()' +
         ',' + CEsc + AUser +
CEsc + ',' +
         CEsc + vAction + CEsc
+ ');';
    vQuery                  :=
TADOQuery.Create(Nil);
    with vQuery do
      begin
        if
(SysConnection.Connected     =
False) then

SysConnection.Connected     :=
True;
        Connection          :=
SysConnection;
        Active := False;
        SQL.Clear;
        SQL.Add(vSyntax);
        ExecSQL;
      end;
  except
    on E : Exception do
      begin

ShowMessage(E.Message);
      end;
  end;
end;

function   TSys.Query(ASyntax:
String;           ATipeQuery:
TTipeQuery): TADOQuery;
var
  vQuery : TADOQuery;
begin
  try
    vQuery                  :=
TADOQuery.Create(Nil);
    {if  (vLogin.StatusLogin  =
Offline) then
      begin
        Result := Nil;
        Exit;
      end;   }
    with vQuery do
      begin
        if
(SysConnection.Connected     =
False) then

SysConnection.Connected     :=
True;
        Connection          :=
SysConnection;
        Active := False;
        SQL.Clear;
        SQL.Add(ASyntax);
        if    (ATipeQuery    =
Executed) then
          ExecSQL
        else if (ATipeQuery =
Actived) then
          begin
```

```
            ExecSQL;
            Active := True;
          end;
      end;

    //JIKA   USER   MELAKUKAN
MANIPULASI DATA MAKA MASUKKAN
KEDALAM LOG
    if (ATipeQuery = Executed)
then

Log(FLoginUser.vNamaPegawai,
ASyntax);
    StatusQuery := Valid;
    Result := vQuery;
  except
    on E : Exception do
      begin
        StatusQuery         :=
NotValid;
        Result := Nil;

ShowMessage(E.Message);
      end;
  end;
end;

procedure
TSys.SetConnect(const   Value:
Boolean);
begin
  FConnect := Value;
end;

procedure
TSys.SetConnectionString(const
Value: String);
begin
  FConnectionString := Value;
end;

procedure TSys.SetDBName(const
Value: String);
begin
  FDBName := Value;
end;

procedure
TSys.SetHostName(const  Value:
String);
begin
  FHostName := Value;
end;

procedure
TSys.SetPassword(const  Value:
String);
begin
  FPassword := Value;
end;

procedure   TSys.SetPort(const
Value: integer);
begin
  FPort := Value;
end;

procedure
TSys.SetProtocol(const  Value:
String);
begin
  FProtocol := Value;
end;

procedure
TSys.SetStatusConnection(const
Value: TStatusConnection);
begin
  FStatusConnection := Value;
end;

procedure
TSys.SetStatusQuery(const
Value: TStatusQuery);
begin
  FStatusQuery := Value;
end;

procedure
TSys.SetSysConnection(const
Value: TADOConnection);
begin
  FSysConnection := Value;
end;
```

```
procedure   TSys.SetUser(const
Value: String);
begin
  FUser := Value;
end;
end.

unit UProfileUser;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls,
XPMan,  ImgList,  DB,  ComCtrls,
ToolWin, Menus,
  ADODB;

type
  TFProfileUser = class(TForm)
    ToolBar: TToolBar;
    TBUbah: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
    TBKeluar: TToolButton;
    ImageList: TImageList;
    XPStyle: TXPManifest;
    PanelDetail: TPanel;
    LblNIP: TLabel;
    LblNamaPegawai: TLabel;
    EdNIP: TEdit;
    EdNamaPegawai: TEdit;
    LblTglLahir: TLabel;
    DTTglLahir:
TDateTimePicker;
    LblAlamat: TLabel;
    EdAlamat: TMemo;
    LblUserName: TLabel;
    EdUserName: TEdit;
    LblGantiUserName: TLabel;
    LblPassword: TLabel;
    EdPassword: TEdit;
    Label2: TLabel;
    procedure
TBKeluarClick(Sender:
TObject);
    procedure FormShow(Sender:
TObject);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBBatalClick(Sender: TObject);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
LblGantiUserNameClick(Sender:
TObject);
    procedure
Label2Click(Sender: TObject);
  private
    { Private declarations }
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewProfile();
  public
    { Public declarations }
  end;

var
  FProfileUser: TFProfileUser;

implementation

uses   ULoginUser,   UPegawai,
UGantiUserName,
UGantiPassword;

{$R *.dfm}

procedure
TFProfileUser.TBKeluarClick(Se
nder: TObject);
begin
  Close;
end;

procedure
TFProfileUser.SetKontrol(ABool
: Boolean);
begin
```

```
  TBUbah.Enabled := Not ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  DTTglLahir.Enabled := ABool;
  EdNamaPegawai.Enabled     :=
ABool;
  EdAlamat.Enabled := ABool;
end;

procedure
TFProfileUser.ViewProfile;
var
  vQuery : TADOQuery;
  vData : TPegawai;
begin
  try
    vData := TPegawai.Create;
    vQuery                  :=
vData.GetDataSearchPegawai('ni
p',FLoginUser.vNIP);
    EdNIP.Text              :=
vQuery.Fields[0].Text;
    EdNamaPegawai.Text      :=
vQuery.Fields[1].Text;
    DTTglLahir.Date         :=
vQuery.Fields[2].Value;
    EdAlamat.Text           :=
vQuery.Fields[3].Text;
    FreeAndNil(vQuery);
    vQuery                  :=
vData.GetDataSearchUser('nip',
FLoginUser.vNIP);
    EdUserName.Text         :=
vQuery.Fields[1].Text;
    EdPassword.Text         :=
vQuery.Fields[2].Text;
    SetKontrol(False);
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vData);
  end;
end;

procedure
TFProfileUser.FormShow(Sender:
TObject);
begin
  ViewProfile;
end;

procedure
TFProfileUser.TBUbahClick(Send
er: TObject);
begin
  SetKontrol(True);
  EdNamaPegawai.SetFocus;
end;

procedure
TFProfileUser.TBBatalClick(Sen
der: TObject);
begin
  ViewProfile;
end;

procedure
TFProfileUser.TBSimpanClick(Se
nder: TObject);
var
  vData : TPegawai;
begin
  try
    vData := TPegawai.Create;
    If MessageDlg('Apakah Anda
Ingin   Menyimpan   Perubahan
Profile Anda ?',
      mtConfirmation,
[mbYes,mbNo], 0) = mrYes Then
      begin
        If
vData.UpdatePegawai(
            EdNIP.Text,

EdNamaPegawai.Text,
            DTTglLahir.Date,
            EdAlamat.Text)
Then

ShowMessage('Penyimpanan
Perubahan   Profile   Berhasil
!');
        ViewProfile;
      end
  finally
    FreeAndNil(vData);
  end;
```

```
end;

procedure
TFProfileUser.LblGantiUserName
Click(Sender: TObject);
begin

FGantiUserName.UbahUserName(Ed
NIP.Text);
  FGantiUserName.ShowModal;
end;

procedure
TFProfileUser.Label2Click(Send
er: TObject);
begin

FGantiPassword.UbahPassword(Ed
NIP.Text);
  FGantiPassword.ShowModal;
end;

end.

unit UPermintaanMakananReport;

interface

uses    Windows,    SysUtils,
Messages,  Classes,  Graphics,
Controls,
  StdCtrls,  ExtCtrls,  Forms,
QuickRpt, QRCtrls;

type
  TReportPermintaanMakanan   =
class(TQuickRep)
    ColumnHeaderBand1:
TQRBand;
    PageHeaderBand1: TQRBand;
    QRGroup1: TQRGroup;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel1: TQRLabel;
    QRLabel7: TQRLabel;
    PageFooterBand1: TQRBand;
    QRSysData1: TQRSysData;
    QRSysData2: TQRSysData;
    QRLabel8: TQRLabel;
    DetailBand1: TQRBand;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    QRDBText9: TQRDBText;
    QRDBText6: TQRDBText;
    QRDBText7: TQRDBText;
    QRDBText8: TQRDBText;
  private

  public

  end;

var
  ReportPermintaanMakanan:
TReportPermintaanMakanan;

implementation

uses UDMReport, FLapPasien;

{$R *.DFM}

end.

unit UPermintaanMakanan;

interface

uses
  Classes, SysUtils, TypInfo,
StrUtils ,ADODB, Dialogs;

type
  TPermintaanMakanan = class
  private
    FIdPengurus: String;
    FNoDokIRI: String;
    FKeterangan: String;
    FIdPermintaanMakanan:
String;
```

```
    FTglPermintaan: TDateTime;
    procedure
SetIdPengurus(const     Value:
String);
    procedure
SetKeterangan(const     Value:
String);
    procedure
SetNoDokIRI(const       Value:
String);
    procedure
SetPermintaanMakanan(const
Value: String);
    procedure
SetTglPermintaan(const  Value:
TDateTime);

  published
    property
IdPermintaaanMakanan : String
read      FIdPermintaanMakanan
write SetPermintaanMakanan;
    property  TglPermintaan  :
TDateTime read FTglPermintaan
write SetTglPermintaan;
    property   Keterangan   :
String read FKeterangan write
SetKeterangan;
    property   IdPengurus   :
String read FIdPengurus write
SetIdPengurus;
    property NoDokIRI : String
read      FNoDokIRI      write
SetNoDokIRI;

    {   PROSEDUR   DAN   FUNGSI
PENGOLAHAN   DATA   PERMINTAAN
MAKANAN }
    function
CreatePermintaanMakanan(
        ATglPermintaan       :
TDateTime;
        AIdPengurus : String)
: Boolean ;
    function
UpdatePermintaanMakanan(
        AIdPermintaanMakanan :
Integer;
        ATglPermintaan       :
TDateTime): Boolean;
    function
DeletePermintaanMakanan(AIdPer
mintaanMakanan  :  Integer)  :
Boolean;

    //DETAIL        PERMINTAAN
MAKANAN
    function
CreateDetailPermintaanMakanan(
        AIdPermintaanMakanan :
Integer;            ANoDokIRI,
AIdMakanan,   AKeterangan   :
String) : Boolean;
    function
UpdateDetailPermintaanMakanan(

AIdDetailPermintaanMakanan   :
integer ;
        AIdMakanan,
AKeterangan   :   String)   :
Boolean;
    function
DeleteDetailPermintaanMakanan(
        AIdPermintaanMakanan :
String;

AIdDetailPermintaanMakanan   :
Integer) : Boolean;

    //AMBIL DATA PERMINTAAN
    function
GetMasterPermintaanMakanan() :
TADOQuery ;
    function
GetDetailPermintaanMakanan(AId
Permintaan   :   String)   :
TADOQuery ;
    function
GetDataSearchPermintaanMakanan
(
        AFieldRef, AKeyWord :
String) : TADOQuery;
    function
GetDataSearchDetailPermintanMa
kanan(AIdPermintaan,
```

```
        AFieldRef, AKeyWord :
String ) : TADOQuery;

    { PROSEDUR DAN FUNGSI
PENGOLAHAN DATA MAKANAN }
    function CreateMakanan(
        ANamaMakanan : String;
AIdJenisMakanan : Integer ) :
Boolean ;
    function UpdateMakanan(
        AIdMakanan : Integer;
ANamaMakanan     :     String;
AIdJenisMakanan : Integer ) :
Boolean ;
    function
DeleteMakanan(AIdMakanan     :
Integer ) : Boolean ;

    function  GetMakanan()  :
TADOQuery ;
    function
GetDataSearchMakanan(AFieldRef
,   AKeyWord   :   String)   :
TADOQuery;

    {PROSEDUR    DAN    FUNGSI
PENGOLAHAN DATA JENIS MAKANAN
}
    function
CreateJenisMakanan(
        ANamaJenisMakanan    :
String) : Boolean;
    function
UpdateJenisMakanan(
        AIdJenisMakanan      :
Integer  ;ANamaJenisMakanan  :
String) : Boolean;
    function
DeleteJenisMakanan(
        AIdJenisMakanan      :
Integer) : Boolean;

    function GetJenisMakanan()
: TADOQuery;
    function
GetDataSearchJenisMakanan(AFie
ldRef, AKeyWord : String) :
TADOQuery;

    { PROSEDURE DAN FUNGSI
PENGOLAHAN DATA BAHAN MAKANAN
}
    function
CreateBahanMakanan(
        ANamaBahanMakanan    :
String) : Boolean ;
    function
UpdateBahanMakanan(
        AIdBahanMakanan      :
Integer; ANamaBahanMakanan :
String) : Boolean ;
    function
DeleteBahanMakanan(AIdBahanMak
anan : Integer) : Boolean;

    //DETAIL BAHAN MAKANAN
    function
CreateDetailBahanMakanan(
        AIdBahan, AIdMakanan :
Integer) : Boolean;
    function
UpdateDetailBahanMakanan(
        AIdDetailBahan,
AIdBahan,     AIdMakanan     :
Integer) : Boolean;
    function
DeleteDetailBahanMakanan(
        AIdDetailBahanMakanan
: Integer) : Boolean;

    //AMBIL DATA BAHAN MAKANAN
    function GetBahanMakanan()
: TADOQuery ;
    function
GetDataSearchBahanMakanan(AFie
ldRef, AKeyWord : String) :
TADOQuery ;
    function
GetDetailBahanMakanan(ADataTyp
e : String; AIdMakananRef :
Integer) : TADOQuery;
    function
GetDataSearchDetailBahanMakana
n(AFieldRef,     AKeyWord     :
String;     AIdMakananRef     :
Integer) : TADOQuery;
```

```
    constructor Create;
    destructor        Destroy;
override;
end;

implementation

uses UIRI, UPegawai, USys;

{ TPermintaanMakanan }

constructor
TPermintaanMakanan.Create;
var
  vPegawai : TPegawai;
begin
  try
    vPegawai                :=
TPegawai.Create;
    IdPengurus              :=
vPegawai.NIP;
  finally
    FreeAndNil(vPegawai);
  end;
end;

function
TPermintaanMakanan.CreateBahan
Makanan(
  ANamaBahanMakanan:  String):
Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'insert   into   bahan
(nama_bahan) ' +
        ' values (' + CEsc +
vSys.DBAddLessString(ANamaBaha
nMakanan)  +  CEsc  +  ')'  ,
Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.CreateDetai
lBahanMakanan(
  AIdBahan,    AIdMakanan    :
Integer): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    //cek apakah detail dengan
id  bahan  dan  id  makanan  yg
sama sudah ada
    vQuery := vSys.Query(
        'select     *     from
detail_bhn_makanan ' +
        ' where id_bhn_makanan
= ' + IntToStr(AIdBahan)  +
        ' and id_makanan = ' +
IntToStr(AIdMakanan),
Actived);
    If (vQuery.RecordCount >0)
Then
      begin
        ShowMessage('Data
Dengan ID.Bahan : ' +
            IntToStr(AIdBahan)
+  '  dan  ID.Makanan  :  '  +
IntToStr(AIdMakanan) +
            ' Sudah Ada, Mohon
Diperbaharui  Data  Yang  Sudah
Ada Saja.');
        Result := False;
        Exit;
      end;
    vQuery := vSys.Query(
        'insert           into
detail_bhn_makanan
```

```
(id_bhn_makanan,id_makanan)  '
+
        '    values    ('    +
IntToStr(AIdBahan)+ ',' +
        IntToStr(AIdMakanan) +
')' , Executed );
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.CreateDetai
lPermintaanMakanan(
  AIdPermintaanMakanan       :
Integer;            ANoDokIRI,
AIdMakanan,    AKeterangan    :
String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
  vMedRec : String;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        '   select   no_med_rec
from iri where ' +
        '  no_dok_iri  =  '  +
CEsc  +  ANoDokIRI  +  CEsc  ,
Actived);
    vMedRec                 :=
vQuery.Fields[0].Text;
    vQuery := vSys.Query(
        '    select    *    from
detail_permintaan_makanan  a  '
+
        '   join   iri   b   on
a.no_dok_iri = b.no_dok_iri '
+
        ' where b.no_med_rec =
' + CEsc + vMedRec + CEsc +
        '                   and
a.id_permintaan_makanan = ' +
IntToStr(AIdPermintaanMakanan)
, Actived);
    If  vQuery.RecordCount  >0
Then
      begin
        ShowMessage('Data
Dengan  No.Dok  IRI  =  '  +
ANoDokIRI +
            '  Sudah  Ada  Pada
Permintaan  Makanan  Ini  !,
Mohon Update Data Yang Sudah
Ada.');
        Result := False;
        FreeAndNil(vQuery);
        FreeAndNil(vSys);
        Exit;
      end;
    vQuery := vSys.Query(
        'insert           into
detail_permintaan_makanan
(id_permintaan_makanan, ' +
        'no_dok_iri,
id_makanan, keterangan) values
(' +

IntToStr(AIdPermintaanMakanan)
+ ',' +
        CEsc  +  ANoDokIRI  +
CEsc + ',' +
        CEsc  +  AIdMakanan  +
CEsc + ',' +
        CEsc                +
vSys.DBAddLessString(AKeterang
an) + CEsc + ')' , Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;
```

```
function
TPermintaanMakanan.CreateJenis
Makanan(ANamaJenisMakanan:
String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    {vQuery := vSys.Query(
        'insert           into
jenis_makanan
(id_jenis_makanan,
nama_jenis_makanan) ' +
        ' values ( ' + CEsc +
AIdJenisMakanan + CEsc + ',' +
       CEsc                +
ANamaJenisMakanan + CEsc + ')'
, Executed);
    *}
    vQuery := vSys.Query(
        'insert           into
jenis_makanan
(nama_jenis_makanan) ' +
        ' values ( ' + CEsc +
ANamaJenisMakanan + CEsc + ')'
, Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.CreateMakan
an(ANamaMakanan : String;
    AIdJenisMakanan          :
Integer): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'insert  into  makanan
(nama_makanan,
id_jenis_makanan) ' +
        ' values (' + CEsc +
vSys.DBAddLessString(ANamaMaka
nan) + CEsc + ',' +

IntToStr(AIdJenisMakanan)    +
')' , Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.CreatePermi
ntaanMakanan(
    ATglPermintaan: TDateTime;
AIdPengurus     :     String):
Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        '     insert     into
permintaan_makanan (' +
        '
tanggal_permintaan_makanan,
nip) values (' +
        CEsc                +
FormatDateTime('yyyy-mm-
dd',ATglPermintaan)  +  CEsc  +
',' +
        CEsc  +  AIdPengurus  +
CEsc +
        ')' , Executed);
```

```
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.DeleteBahan
Makanan(
  AIdBahanMakanan:   Integer):
Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'delete   from   bahan
where id_bhn_makanan = ' +

IntToStr(AIdBahanMakanan),
Executed);
    if (vQuery.RowsAffected =
1) then
      begin
        vQuery := vSys.Query(
            'Delete       from
detail_bhn_makanan where ' +
            ' id_bhn_makanan =
' + IntToStr(AIdBahanMakanan),
Executed);
        Result := True;
      end
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.DeleteDetai
lBahanMakanan(
  AIdDetailBahanMakanan:
Integer): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'delete           from
detail_bhn_makanan       where
id_detail_bhn_makanan = ' +

IntToStr(AIdDetailBahanMakanan
), Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.DeleteDetai
lPermintaanMakanan(AIdPerminta
anMakanan : String;
  AIdDetailPermintaanMakanan:
Integer): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'delete           from
detail_permintaan_makanan
where ' +
        ' id_detail_permintaan
=             '             +
```

```
IntToStr(AIdDetailPermintaanMa
kanan) +
        '                  and
id_permintaan_makanan  =  '  +
CEsc + AIdPermintaanMakanan +
CEsc,
        Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.DeleteJenis
Makanan(
  AIdJenisMakanan:   Integer):
Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'select * from makanan
where ' +
        ' id_jenis_makanan = '
+  IntToStr(AIdJenisMakanan)  ,
Actived);
    If (vQuery.RecordCount >0)
Then
      begin
        ShowMessage('Data
Jenis   Terpakai   Oleh   Data
Makanan !!');
        Result := False;
        Exit;
      end;

    vQuery := vSys.Query(
        'delete           from
jenis_makanan            where
id_jenis_makanan = ' +

IntToStr(AIdJenisMakanan),
Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.DeleteMakan
an(AIdMakanan:       Integer):
Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'delete  from  makanan
where id_makanan = ' +
        IntToStr(AIdMakanan),
Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.DeletePermi
ntaanMakanan(
  AIdPermintaanMakanan:
Integer): Boolean;
```

```
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'delete           from
detail_permintaan_makanan
where  id_permintaan_makanan  =
' +

IntToStr(AIdPermintaanMakanan)
, Executed);
    vQuery := vSys.Query(
        'delete           from
permintaan_makanan       where
id_permintaan_makanan = ' +

IntToStr(AIdPermintaanMakanan)
, Executed);
    if  (vQuery.RowsAffected  =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

destructor
TPermintaanMakanan.Destroy;
begin

  inherited;
end;

function
TPermintaanMakanan.GetBahanMak
anan: TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        '               select
id_bhn_makanan     "ID.Bahan",
nama_bahan "Nama Bahan" ' +
        ' from bahan order by
id_bhn_makanan', Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetDataSear
chBahanMakanan(AFieldRef,
  AKeyWord:           String):
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select id_bhn_makanan
"ID.Bahan",  nama_bahan  "Nama
Bahan", ' +
        '  jumlah_stok  "Stok"
from bahan where ' +
        AFieldRef + ' like ' +
CEsc + '%' + AKeyWord + '%' +
CEsc +
        '       order       by
id_bhn_makanan', Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetDataSear
chDetailBahanMakanan(AFieldRef
,
  AKeyWord:            String;
AIdMakananRef    :    Integer):
TADOQuery;
var
  vSys : TSys;
  vStrSQL : String;
begin
  try
```

```
    vSys := TSys.Create;
    If (AFieldRef = 'Nama
Makanan') Then
      begin
        vStrSQL := ' select
distinct          a.id_makanan
"ID.Makanan", ' +
        ' b.nama_makanan "Nama
Makanan", ' +
        ' c.nama_jenis_makanan
"Jenis Makanan" ' +
        '                 from
detail_bhn_makanan a, ' +
        '      makanan      b,
jenis_makanan c where ' +
        '    a.id_makanan    =
b.id_makanan and ' +
        ' b.id_jenis_makanan =
c.id_jenis_makanan ' +
        '  and  b.nama_makanan
like ' +
        CEsc + '%' + AKeyWord
+ '%' + CEsc +
        '       order       by
a.id_detail_bhn_makanan ';
      end
    Else If(AFieldRef = 'Nama
Bahan') Then
      begin
        vStrSQL := ' select
a.id_detail_bhn_makanan  "ID",
' +
        '     a.id_bhn_makanan
"ID.Bahan" , ' +
        '  b.nama_bahan  "Nama
Bahan" , ' +
        '          a.id_makanan
"ID.Makanan", ' +
        ' c.nama_makanan "Nama
Makanan" ' +
        '                 from
detail_bhn_makanan a, ' +
        ' bahan b, ' +
        ' makanan c where ' +
        '  a.id_bhn_makanan  =
b.id_bhn_makanan and ' +
        '    a.id_makanan    =
c.id_makanan and ' +
        ' a.id_makanan = ' +
IntToStr(AIdMakananRef) +
        '   and   b.nama_bahan
like ' +
        CEsc + '%' + AKeyWord
+ '%' + CEsc +
        '       order       by
a.id_detail_bhn_makanan ';
      end;
    Result                  :=
vSys.Query(vStrSQL, Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetDataSear
chDetailPermintanMakanan(AIdPe
rmintaan, AFieldRef,
  AKeyWord:           String):
TADOQuery;
var
  vSys : TSys;
  vStrSQL  : String;
begin
  try
    vSys := TSys.Create;
    vStrSQL     :=     'select
a.id_detail_permintaan  "ID"  ,
' +
        '
a.id_permintaan_makanan
"ID.Permintaan", ' +
        ' a.no_dok_iri "No.Dok
IRI" , ' +
        ' c.no_med_rec "No.Med
Rec" , ' +
        '  d.nama_pasien  "Nama
Pasien", ' +
        '          a.id_makanan
"ID.Makanan", ' +
        ' b.nama_makanan "Nama
Makanan", ' +
        '          a.keterangan
"Keterangan" from ' +
```

```
        '
detail_permintaan_makanan    a
left outer join makanan b ' +
        '  on  a.id_makanan  =
b.id_makanan  ' +
        ' left outer join iri
c     on     a.no_dok_iri     =
c.no_dok_iri ' +
        '   left   outer   join
pasien  d  on  c.no_med_rec  =
d.no_med_rec ' +
        '                 where
a.id_permintaan_makanan = ' +
CEsc + AIdPermintaan + CEsc +
        ' and ' ;
    If       AFieldRef       =
'Nama_Pasien' Then
      vStrSQL := vStrSQL +  '
d.' + AFieldRef + ' like ' +
CEsc + '%' +
        AKeyWord + '%' + CEsc
+       '       order       by
a.id_detail_permintaan '
    Else   If   AFieldRef   =
'No_Med_Rec' Then
      vStrSQL := vStrSQL + '
c.' + AFieldRef + ' like ' +
CEsc + '%' +
        AKeyWord + '%' + CEsc
+       '       order       by
a.id_detail_permintaan '
    Else
      vStrSQL := vStrSQL +  '
a.' + AFieldRef + ' like ' +
CEsc + '%' +
        AKeyWord + '%' + CEsc
+       '       order       by
a.id_detail_permintaan ';
    Result                  :=
vSys.Query(vStrSQL , Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetDataSear
chJenisMakanan(AFieldRef,
  AKeyWord:           String):
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select
id_jenis_makanan "ID.Jenis" ,
' +
        '   nama_jenis_makanan
"Nama Jenis" from ' +
        ' jenis_makanan ' +
        ' where ' + AFieldRef
+ ' like ' + CEsc + '%' +
        AKeyWord + '%' + CEsc
+ ' order by id_jenis_makanan
' , Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetDataSear
chMakanan(AFieldRef,
  AKeyWord:           String):
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        '  select  a.id_makanan
"ID.Makanan" , ' +
        ' a.nama_makanan "Nama
Makanan" , ' +
        '   a.id_jenis_makanan
"ID Jenis" , ' +
        ' b.nama_jenis_makanan
"Nama Jenis" ' +
        '   from   makanan   a,
jenis_makanan b ' +
        '                 where
a.id_jenis_makanan          =
b.id_jenis_makanan and ' +
```

```
        ' a.' + AFieldRef + '
like ' + CEsc + '%' + AKeyWord
+
        '%' + CEsc + ' order
by a.id_makanan ' , Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetDataSear
chPermintaanMakanan(AFieldRef,
  AKeyWord:           String):
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select
a.id_permintaan_makanan
"ID.Permintaan", ' +
        '
a.tanggal_permintaan_makanan
"Tgl Permintaan",' +
        '      a.nip      "ID
Pengurus",' +
        ' b.nama_pegawai "Nama
Pengurus"' +
        '                 from
permintaan_makanan   a   left
outer join pegawai b ' +
        '  on  a.nip  =  b.nip
where ' +
        ' a.' + AFieldRef + '
like ' + CEsc + '%' +
        AKeyWord + '%' + CEsc
+       '      order      by
a.id_permintaan_makanan'     ,
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetDetailBa
hanMakanan(ADataType : String;
AIdMakananRef  :  Integer)  :
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    If(ADataType  =  'Master')
Then
      begin
        Result := vSys.Query(
          '  select  distinct
a.id_makanan "ID. Makanan", '
+
          '     b.nama_makanan
"Nama Makanan", ' +
          '
c.nama_jenis_makanan    "Jenis
Makanan" ' +
          '               from
detail_bhn_makanan a, makanan
b, jenis_makanan c where ' +
          '   a.id_makanan   =
b.id_makanan and ' +
          ' b.id_jenis_makanan
= c.id_jenis_makanan ' +
          '      order      by
a.id_makanan', Actived);
      end
    Else    If(ADataType    =
'Detail') Then
      begin
        Result := vSys.Query(
          '             select
a.id_detail_bhn_makanan  "ID",
' +
          '   a.id_bhn_makanan
"ID.Bahan" , ' +
          ' b.nama_bahan "Nama
Bahan" , ' +
          '       a.id_makanan
"ID.Makanan", ' +
          '     c.nama_makanan
"Nama Makanan"' +
```

```
          '               from
detail_bhn_makanan a, ' +
          ' bahan b, ' +
          ' makanan c where '
+
          ' a.id_bhn_makanan =
b.id_bhn_makanan and ' +
          '   a.id_makanan   =
c.id_makanan and ' +
          ' a.id_makanan =' +
IntToStr(AIdMakananRef) +
          '      order      by
a.id_detail_bhn_makanan     ',
Actived);
      end;
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetDetailPe
rmintaanMakanan(AIdPermintaan
: String): TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select
a.id_detail_permintaan "ID" ,
' +
        '
a.id_permintaan_makanan
"ID.Permintaan", ' +
        ' a.no_dok_iri "No.Dok
IRI" , ' +
        ' c.no_med_rec "No.Med
Rec" , ' +
        ' d.nama_pasien "Nama
Pasien", ' +
        '         a.id_makanan
"ID.Makanan", ' +
        ' b.nama_makanan "Nama
Makanan", ' +
        '         a.keterangan
"Keterangan" from ' +
        '
detail_permintaan_makanan    a
left outer join makanan b ' +
        '  on  a.id_makanan  =
b.id_makanan  ' +
        ' left outer join iri
c     on     a.no_dok_iri     =
c.no_dok_iri ' +
        '   left   outer   join
pasien  d  on  c.no_med_rec  =
d.no_med_rec ' +
        '                where
a.id_permintaan_makanan = ' +
CEsc + AIdPermintaan + CEsc +
        '       order       by
a.id_detail_permintaan   '   ,
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetJenisMak
anan: TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select
id_jenis_makanan "ID.Jenis" ,
' +
        '   nama_jenis_makanan
"Nama Jenis" from ' +
        ' jenis_makanan order
by     id_jenis_makanan     ',
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetMakanan:
TADOQuery;
var
```

```
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select   a.id_makanan
"ID. Makanan" , ' +
        ' a.nama_makanan "Nama
Makanan" , ' +
        '   a.id_jenis_makanan
"ID. Jenis" , ' +
        ' b.nama_jenis_makanan
"Nama Jenis" ' +
        '   from   makanan   a,
jenis_makanan b ' +
        '                where
a.id_jenis_makanan           =
b.id_jenis_makanan ' +
        '       order       by
a.id_makanan ' , Actived) ;
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.GetMasterPe
rmintaanMakanan: TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select
a.id_permintaan_makanan
"ID.Permintaan", ' +
        '
a.tanggal_permintaan_makanan
"Tgl Permintaan",' +
        '      a.nip      "ID
Pengurus",' +
        ' b.nama_pegawai "Nama
Pengurus"' +
        '                from
permintaan_makanan   a   left
outer join pegawai b ' +
        '  on  a.nip  =  b.nip
order                       by
a.id_permintaan_makanan   '   ,
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

procedure
TPermintaanMakanan.SetIdPengur
us(const Value: String);
begin
  FIdPengurus := Value;
end;

procedure
TPermintaanMakanan.SetKeterang
an(const Value: String);
begin
  FKeterangan := Value;
end;

procedure
TPermintaanMakanan.SetNoDokIRI
(const Value: String);
begin
  FNoDokIRI := Value;
end;

procedure
TPermintaanMakanan.SetPerminta
anMakanan(const         Value:
String);
begin
  FIdPermintaanMakanan      :=
Value;
end;

procedure
TPermintaanMakanan.SetTglPermi
ntaan(const Value: TDateTime);
begin
  FTglPermintaan := Value;
end;

function
TPermintaanMakanan.UpdateBahan
```

```
Makanan(AIdBahanMakanan      :
Integer;
  ANamaBahanMakanan:  String):
Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        ' update bahan set ' +
        '  nama_bahan  =  '  +
CEsc                          +
vSys.DBAddLessString(ANamaBaha
nMakanan) + CEsc +
        ' where id_bhn_makanan
=             '             +
IntToStr(AIdBahanMakanan),
Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.UpdateDetai
lBahanMakanan(AIdDetailBahan,
    AIdBahan,   AIdMakanan   :
Integer): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery   :=   vSys.Query('
select         *         from
detail_bhn_makanan where ' +
        ' id_bhn_makanan = ' +
IntToStr(AIdBahan) +
        ' and id_makanan = ' +
IntToStr(AIdMakanan),
Actived);
    If (vQuery.RecordCount >0)
Then
      begin
        ShowMessage('Data
Dengan   Bahan   dan   Makanan
Tersebut Sudah Ada !');
        Result := False;
        Exit;
      end;
    vQuery := vSys.Query(
        '               update
detail_bhn_makanan set ' +
        ' id_bhn_makanan = ' +
IntToStr(AIdBahan) + ',' +
        '  id_makanan  =  '  +
IntToStr(AIdMakanan) +
        '                where
id_detail_bhn_makanan  =  '  +
IntToStr(AIdDetailBahan),
Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.UpdateDetai
lPermintaanMakanan(
    AIdDetailPermintaanMakanan
: integer ;
    AIdMakanan, AKeterangan :
String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
```

```
        '               update
detail_permintaan_makanan  set
' +
        '  id_makanan  =  '  +
CEsc + AIdMakanan + CEsc + ','
+
        '  keterangan  =  '  +
CEsc                        +
vSys.DBAddLessString(AKeterang
an) + CEsc +
        '                where
id_detail_permintaan  =  '  +
IntToStr(AIdDetailPermintaanMa
kanan), Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.UpdateJenis
Makanan(AIdJenisMakanan      :
Integer;
  ANamaJenisMakanan:  String):
Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        ' update jenis_makanan
set nama_jenis_makanan = ' +
        CEsc                +
ANamaJenisMakanan + CEsc +
        '                where
id_jenis_makanan    =    '    +
IntToStr(AIdJenisMakanan),
Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.UpdateMakan
an(AIdMakanan : Integer;
    ANamaMakanan   :   String;
AIdJenisMakanan   :   Integer):
Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'update makanan set '
+
        ' nama_makanan = ' +
CEsc + ANamaMakanan + CEsc +
',' +
        ' id_jenis_makanan ='
+ IntToStr(AIdJenisMakanan) +
        ' where id_makanan = '
+        IntToStr(AIdMakanan),
Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPermintaanMakanan.UpdatePermi
ntaanMakanan(AIdPermintaanMaka
nan : Integer;
```

```
  ATglPermintaan:  TDateTime):
Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        '               update
permintaan_makanan set ' +
        '
tanggal_permintaan_makanan = '
+ CEsc + FormatDateTime('yyyy-
mm-dd',ATglPermintaan) + CEsc
+
        '               where
id_permintaan_makanan  =  '  +
IntToStr(AIdPermintaanMakanan)
, Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

end.

unit UPegawai;

interface

uses
  Classes, SysUtils, TypInfo,
StrUtils ,ADODB, Dialogs;
type
  TStatusLogin   =   (Online,
Offline);

type
  TPegawai = class
  private
    FAlamat: String;
    FNIP: String;
    FNamaPegawai: String;
    FTglLahir: TDateTime;
    FStatusLogin:
TStatusLogin;
    FUserName: String;
    FHakAkses: String;
    procedure  SetAlamat(const
Value: String);
    procedure
SetNamaPegawai(const    Value:
String);
    procedure     SetNIP(const
Value: String);
    procedure
SetTglLahir(const       Value:
TDateTime);
    procedure
SetStatusLogin(const    Value:
TStatusLogin);
    procedure
SetUserName(const       Value:
String);
    procedure
SetHakAkses(const       Value:
String);

  published
    property NIP : String read
FNIP write SetNIP;
    property   NamaPegawai   :
String read FNamaPegawai write
SetNamaPegawai;
    property    TglLahir    :
TDateTime read FTglLahir write
SetTglLahir;
    property Alamat : String
read FAlamat write SetAlamat;
    property UserName : String
read     FUserName     write
SetUserName;
    property   StatusLogin   :
TStatusLogin read FStatusLogin
write SetStatusLogin;
    property HakAkses : String
read     FHakAkses     write
SetHakAkses;
```

```
    { PROSEDUR DAN FUNGSI
PENGOLAHAN DATA PEGAWAI }
    function CreatePegawai(
        ANIP, ANamaPegawai :
String;
        ATglLahir : TDateTime
;
        AAlamat, AStatus :
String) : Boolean ;
    function UpdatePegawai(
        ANIP, ANamaPegawai :
String;
        ATglLahir : TDateTime
;
        AAlamat : String) :
Boolean ;
    function
DeletePegawai(ANIP : String) :
Boolean ;
    function
SetStatusPegawai(ANIP, AStatus
: String): Boolean;

    //AMBIL DATA PEGAWAI
    function GetPegawai() :
TADOQuery;
    function
GetDataSearchPegawai(AFieldRef
, AKeyWord : String) :
TADOQuery;
    function
GetDataFinalSearchPegawai(AFie
ldRef, AKeyWord : String) :
TADOQuery;

    { PROSEDUR DAN FUNGSI
PENGOLAHAN DATA USER }
    function CreateUser(
        ANIP,       AUserName,
APassword : String; AHakAkses
: String) : Boolean;
    function UpdateUser(
        ANIP, AHakAkses :
String) : Boolean;
    function
UpdateProfileUser(
        ANIP,       AUserName,
APassword : String) : Boolean;
    function DeleteUser(ANIP :
String) : Boolean ;

    //AMBIL DATA USER
    function GetUser() :
TADOQuery ;
    function
LoginUser(AUserName, APassword
: String) : Boolean;
    function
GetDataSearchUser(AFieldRef,
AKeyWord : String) :
TADOQuery;
    function
GetDataFinalSearchUser(AFieldR
ef, AKeyWord : String) :
TADOQuery;
    constructor Create;
    destructor        Destroy;
override;
end;
implementation

uses USys;


{ TPegawai }

constructor TPegawai.Create;
begin

end;

function
TPegawai.CreatePegawai(ANIP,
ANamaPegawai: String;
  ATglLahir:        TDateTime;
AAlamat,   AStatus:   String):
Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
```

```
    vQuery                  :=
vSys.Query('insert        into
pegawai (nip, nama_pegawai ' +
        ',          tgl_lahir,
alamat,status) values(' + CEsc
+ ANIP + CEsc +
        ',    '    +    CEsc    +
vSys.DBAddLessString(ANamaPega
wai) + CEsc +
        ',    '    +    CEsc    +
FormatDateTime('yyyy-mm-
dd',ATglLahir) + CEsc +
        ',    '    +    CEsc    +
vSys.DBAddLessString(AAlamat)
+ CEsc +
        ', ' + CEsc + AStatus
+ CEsc + ')',Executed);
    if    (vQuery.RowsAffected
=1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPegawai.CreateUser(ANIP,
AUserName,          APassword,
AHakAkses : String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery                  :=
vSys.Query('insert  into  user
(nip, username, ' +
        'password,  hak_akses)
values (' +
        CEsc + ANIP + CEsc +
', ' +
        CEsc                +
vSys.DBAddLessString(AUserName
) + CEsc + ', ' +
        CEsc                +
vSys.DBAddLessString(APassword
) + CEsc + ', ' +
        CEsc  +  AHakAkses  +
CEsc + ')' , Executed);

  if (vQuery.RowsAffected = 1)
then
    begin
      If
(SetStatusPegawai(ANIP,'TerDaf
tar')) Then
        Result := True
      Else
        Result := False;
    end
  else
    Result := False;

  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPegawai.DeletePegawai(ANIP:
String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'delete  from  pegawai
where nip=' + CEsc + ANIP +
CEsc, Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;

  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
```

```
end;

function
TPegawai.DeleteUser(ANIP:
String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'delete   from   user
where nip=' + CEsc + ANIP +
CEsc ,Executed);
    if (vQuery.RowsAffected =
1) then
      begin

If(SetStatusPegawai(ANIP,
'Tidak Terdaftar')) Then
          Result := True
        Else
          Result := False
      end
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

destructor TPegawai.Destroy;
begin

  inherited;
end;

function
TPegawai.GetDataFinalSearchPeg
awai(AFieldRef,
  AKeyWord:           String):
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        '  select  nip  "NIP",
nama_pegawai "Nama Pegawai", '
+
        '    tgl_lahir    "Tgl
Lahir",    alamat    "Alamat",
Status "Status" from pegawai '
+
        ' where ' + AFieldRef
+ ' = ' +
        CEsc                +
vSys.DBAddLessString(AKeyWord)
+ CEsc +
        '   order   by   nip',
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPegawai.GetDataFinalSearchUse
r(AFieldRef,
  AKeyWord:           String):
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        '  select  *  from  user
where ' +
        AFieldRef  +  '  =  '  +
CEsc   +   AKeyWord   +   CEsc,
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPegawai.GetDataSearchPegawai(
AFieldRef, AKeyWord : String)
: TADOQuery;
var
  vSys : TSys;
```

```
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        '  select  nip  "NIP",
nama_pegawai "Nama Pegawai", '
+
        '    tgl_lahir    "Tgl
Lahir",    alamat    "Alamat",
Status "Status" from pegawai '
+
        ' where ' + AFieldRef
+ ' like ' +
        CEsc    +    '%'    +
vSys.DBAddLessString(AKeyWord)
+ '%' + CEsc +
        '   order   by   nip',
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPegawai.GetDataSearchUser(AFi
eldRef,
  AKeyWord:           String):
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        ' select * from user
where ' +
        AFieldRef + ' like ' +
CEsc + '%' + AKeyWord + '%' +
CEsc, Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function  TPegawai.GetPegawai:
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        '  select  nip  "NIP",
nama_pegawai "Nama Pegawai", '
+
        '    tgl_lahir    "Tgl
Lahir", alamat "Alamat" , ' +
        ' status "Status" from
pegawai ' +
        '   order   by   nip',
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function     TPegawai.GetUser:
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select   nip   "NIP",
username "User Name", password
"Password", '  +
        'hak_akses "Hak Akses"
from  user  order  by  nip',
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TPegawai.LoginUser(AUserName,
APassword: String) : Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
```

```
        'select  *  from  user
where username=' +
        CEsc                 +
vSys.DBAddLessString(AUserName
) + CEsc +
        ' and password=' +
        CEsc                 +
vSys.DBAddLessString(APassword
) + CEsc ,Actived);
    if (vQuery.RecordCount = 1
) then
      begin
        StatusLogin := Online;
        NIP                  :=
vQuery.FieldByName('nip').Text
;
        Result := True;
      end
    else
      begin
        StatusLogin          :=
Offline;
        NIP := '';
        Result := False;
      end;
  finally
    FreeAndNil(vSys);
  end;
end;

procedure
TPegawai.SetAlamat(const
Value: String);
begin

end;

procedure
TPegawai.SetHakAkses(const
Value: String);
begin

end;

procedure
TPegawai.SetNamaPegawai(const
Value: String);
begin
end;

procedure
TPegawai.SetNIP(const   Value:
String);
begin

end;

procedure
TPegawai.SetStatusLogin(const
Value: TStatusLogin);
begin

end;

function
TPegawai.SetStatusPegawai(ANIP
, AStatus: String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'update pegawai set '
+
        ' status = ' + CEsc +
AStatus + CEsc +
        ' where nip = ' + CEsc
+ ANIP + CEsc, Executed);
    If (vQuery.RowsAffected =
1) Then
      Result := True
    Else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

procedure
TPegawai.SetTglLahir(const
Value: TDateTime);
begin
```

```
end;

procedure
TPegawai.SetUserName(const
Value: String);
begin

end;

function
TPegawai.UpdatePegawai(ANIP,
ANamaPegawai: String;
  ATglLahir:        TDateTime;
AAlamat: String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'update  pegawai  set
nama_pegawai = ' +
       CEsc                +
vSys.DBAddLessString(ANamaPega
wai) + CEsc + ',' +
        'tgl_lahir = ' + CEsc
+     FormatDateTime('yyyy-mm-
dd',ATglLahir) + CEsc + ',' +
        'alamat = ' + CEsc +
vSys.DBAddLessString(AAlamat)
+ CEsc +
        ' where nip= ' + CEsc
+ ANIP + CEsc , Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPegawai.UpdateProfileUser(ANI
P, AUserName,
  APassword: String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'update    user    set
username   =   '   +   CEsc   +
AUserName + CEsc +  ',' +
        ' password = ' + CEsc
+ APassword + CEsc +
        ' where nip = ' + CEsc
+ ANIP + CEsc, Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TPegawai.UpdateUser(ANIP,
AHakAkses : String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'update    user    set
hak_akses   =   '   +   CEsc   +
AHakAkses + CEsc +
        ' where nip = ' + CEsc
+ ANIP + CEsc, Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
```

```
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

end.

unit UPasienReport;

interface

uses    Windows,    SysUtils,
Messages,  Classes,  Graphics,
Controls,
  StdCtrls,  ExtCtrls,  Forms,
QuickRpt, QRCtrls;

type
  TReportPasien              =
class(TQuickRep)
    ColumnHeaderBand1:
TQRBand;
    DetailBand1: TQRBand;
    PageFooterBand1: TQRBand;
    PageHeaderBand1: TQRBand;
    QRLabel2: TQRLabel;
    QRLabel1: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRDBText1: TQRDBText;
    QRSysData1: TQRSysData;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    QRLabel7: TQRLabel;
    QRDBText6: TQRDBText;
  private

  public

  end;

var
  ReportPasien: TReportPasien;

implementation

uses UDMReport;

{$R *.DFM}

end.

unit ULoginUser;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs,  StdCtrls,  Buttons,
ExtCtrls, ADODB;

type
  TFLoginUser = class(TForm)
    PanelLogin: TPanel;
    LblUserName: TLabel;
    LblPassword: TLabel;
    EdUserName: TEdit;
    EdPassword: TEdit;
    BtnMasuk: TBitBtn;
    BtnBatal: TBitBtn;
    procedure
BtnMasukClick(Sender:
TObject);
    procedure
BtnBatalClick(Sender:
TObject);
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
    procedure SetHakAkses();
  public
    { Public declarations }
    vNIP : String;
    vNamaPegawai : String;
    vHakAkses : String;
  end;
```

```
var
  FLoginUser: TFLoginUser;

implementation

uses   UPegawai,   UMenuUtama,
UDaftarUser;

{$R *.dfm}

procedure
TFLoginUser.BtnMasukClick(Send
er: TObject);
var
  vData : TPegawai;
  vQuery : TADOQuery;
begin
  try
    vData := TPegawai.Create;
    If                      not
(vData.LoginUser(EdUserName.Te
xt, EdPassword.Text)) Then
      begin
        ShowMessage('Username
dan Password Salah !' );
      end
    Else
      begin
        SetHakAkses;
        vQuery              :=
vData.GetDataSearchPegawai('ni
p', vNIP);
        vNamaPegawai        :=
vQuery.Fields[1].Text;
        With FUtama Do
          begin

StatusBar.Panels[1].Text    :=
'Status Login : ' + vHakAkses
+   ','   +   vNip   +   '-'   +
vNamaPegawai;
            MenuLogin.Visible
:= False;
            MenuLogOut.Visible
:= True;
          end;
        Close;
      end;
  finally
    FreeAndNil(vData);
    FreeAndNil(vQuery);
  end;
end;

procedure
TFLoginUser.SetHakAkses;
var
  vQuery : TADOQuery;
  vData : TPegawai;
begin
  try
    vQuery                  :=
vData.GetDataSearchUser('usern
ame', EdUserName.Text);
    vNIP                    :=
vQuery.Fields[0].Text;
    vHakAkses               :=
vQuery.Fields[3].Text;
    if    (    vHakAkses    =
'Administrator') Then
      begin
        With FUtama Do
          begin

MenuDataUser.Visible := True;

MenuDataPegawai.Visible     :=
True;

MenuInstalasiGizi.Visible   :=
False;
          end;
      end
    Else   If   (vHakAkses   =
'Staff') Then
      begin
        With FUtama Do
          begin

MenuDataUser.Visible := False;

MenuDataPegawai.Visible     :=
False;

MenuInstalasiGizi.Visible   :=
True;
```

```
          end;
      end;
    FUtama.MenuLaporan.Visible
:= True;

FUtama.MenuDataMaster.Visible
:= True;

FUtama.MenuProfileUser.Visible
:= True;
  finally
    FreeAndNil(vData);
    FreeAndNil(vQuery);
  end;
end;

procedure
TFLoginUser.BtnBatalClick(Send
er: TObject);
begin
  Close;
end;

procedure
TFLoginUser.FormShow(Sender:
TObject);
begin
  EdUserName.Clear;
  EdPassword.Clear;
  EdUserName.SetFocus;
end;

end.

unit UIRIReport;

interface

uses    Windows,    SysUtils,
Messages,  Classes,  Graphics,
Controls,
  StdCtrls,  ExtCtrls,  Forms,
QuickRpt, QRCtrls;

type
  TReportIRI                =
class(TQuickRep)
    ColumnHeaderBand1:
TQRBand;
    DetailBand1: TQRBand;
    PageFooterBand1: TQRBand;
    PageHeaderBand1: TQRBand;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRDBText1: TQRDBText;
    QRSysData1: TQRSysData;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
  private

  public

  end;

var
  ReportIRI: TReportIRI;

implementation

uses UDMReport;

{$R *.DFM}

end.

unit UIRI;

interface

uses
  Classes, SysUtils, TypInfo,
StrUtils ,ADODB, Dialogs;

type
  TIRI = class
  private
    FCaraMasuk: String;
    FNoDokIRI: String;
    FNoMedRec: String;
```

```
    FCaraBayar: String;
    FIdRuang: String;
    FTglMasuk: TDateTime;
    FTglKeluar: TDateTime;
    procedure
SetCaraBayar(const      Value:
String);
    procedure
SetCaraMasuk(const      Value:
String);
    procedure SetIdRuang(const
Value: String);
    procedure
SetNoDokIRI(const       Value:
String);
    procedure
SetNoMedRec(const       Value:
String);
    procedure
SetTglKeluar(const      Value:
TDateTime);
    procedure
setTglMasuk(const       Value:
TDateTime);

  published
    property NoDokIRI : String
read      FNoDokIRI      write
SetNoDokIRI;
    property NoMedRec : String
read      FNoMedRec      write
SetNoMedRec;
    property IdRuang : String
read      FIdRuang       write
SetIdRuang;
    property    TglMasuk    :
TDateTime read FTglMasuk write
setTglMasuk;
    property    TglKeluar    :
TDateTime   read   FTglKeluar
write SetTglKeluar;
    property    CaraMasuk    :
String read FCaraMasuk write
SetCaraMasuk;
    property    CaraBayar    :
String read FCaraBayar write
SetCaraBayar;

    { PROSEDUR DAN FUNGSI PADA
PENGOLAHAN DATA IRI }
    function CreateIRI(
        ANoDokIRI,
ANoMedRec,AIdRuang : String;
        ATglMasuk,  ATglKeluar
: TDateTime;
        ACaraMasuk, ACaraBayar
: String) : Boolean ;
    function UpdateIRI(
        ANoDokIRI,
ANoMedRec,AIdRuang : String;
        ATglMasuk,  ATglKeluar
: TDateTime;
        ACaraMasuk, ACaraBayar
: String) : Boolean ;
    function
DeleteIRI(ANoDokIRI  :  String)
: Boolean;

    function    GetIRI()    :
TADOQuery ;
    function
GetDataSearchIRI(AFieldRef,
AKeyWord    :    String)    :
TADOQuery;

    { PROSEDUR DAN FUNGSI PADA
PENGOLAHAN DATA PASIEN }
    function CreatePasien(
        ANoMedRec, ANamaPasien
: String;
        ATglLahir : TDateTime;
AJenisKelamin, AAlamat,
        APendidikan,
APekerjaan, AAgama : String):
Boolean ;
    function UpdatePasien(
        ANoMedRec, ANamaPasien
: String;
        ATglLahir : TDateTime;
AJenisKelamin, AAlamat,
        APendidikan,
APekerjaan, AAgama : String):
Boolean ;
    function    DeletePasien(
ANoMedRec : String) : Boolean;
```

```
    function   GetPasien()   :
TADOQuery;
    function
GetDataSearchPasien(AFieldRef,
AKeyWord : String) : TADOQuery
;

    { PROSEDUR DAN FUNGSI PADA
PENGOLAHAN DATA KELAS }
    function CreateKelas(
        AIdKelas, ANamaKelas :
String) : Boolean;
    function
UpdateKelas(AIdKelas,
ANamaKelas    :    String)    :
Boolean;
    function
DeleteKelas(AIdKelas : String)
: Boolean;
    function   GetKelas()   :
TADOQuery;

    {  PROSEDURE  DAN  FUNGSI
PADA PENGOLAHAN DATA RUANG }
    function CreateRuang(
        AIdRuang, ANamaRuang :
String ;
        AKapasitas : integer ;
        AIdKelas  :  String)  :
Boolean ;
    function UpdateRuang(
        AIdRuang, ANamaRuang :
String;
        AKapasitas : integer;
        AIdKelas  :  String)  :
Boolean ;
    function DeleteRuang(
        AIdRuang  :  String)  :
Boolean;
    function   GetRuang()   :
TADOQuery;
    function
GetDataSearchRuang(AFieldRef,A
KeyWord : String) : TADOQuery;

    constructor Create;
    destructor        Destroy;
override;
end;
implementation

uses USys;

{ TIRI }

constructor TIRI.Create;
begin

end;

function
TIRI.CreateIRI(ANoDokIRI,
ANoMedRec,  AIdRuang:  String;
ATglMasuk,
  ATglKeluar    :    TDateTime;
ACaraMasuk,        ACaraBayar:
String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'insert    into    iri
(no_dok_iri,       no_med_rec,
id_ruang, ' +
        'tanggal_masuk,
tanggal_keluar,    cara_masuk,
cara_bayar) ' +
        'values  ('  +  CEsc  +
ANoDokIRI + CESc + ',' +
        CEsc   +   ANoMedRec   +
CEsc + ',' +
        CEsc + AIdRuang + CEsc
+ ',' +
        CEsc                 +
FormatDateTime('yyyy-mm-
dd',ATglMasuk) + CEsc + ',' +
        CEsc                 +
FormatDateTime('yyyy-mm-
dd',ATglKeluar) + CEsc + ',' +
        CEsc  +   ACaraMasuk  +
CEsc + ',' +
        CEsc  +  ACaraBayar  +
CEsc + ') '
```

```
        ,Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TIRI.CreateKelas(AIdKelas,
ANamaKelas: String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'insert   into   kelas
(id_kelas, nama_kelas) values
(' +
        CEsc + AIdKelas + CEsc
+ ',' +
        CEsc  +  ANamaKelas  +
CEsc + ')' , Executed);
    if    (vQuery.RowsAffected
=1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TIRI.CreatePasien(ANoMedRec,
ANamaPasien: String; ATglLahir
: TDateTime;
  AJenisKelamin,      AALamat,
APendidikan,       APekerjaan,
AAgama: String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
//    ATglLahir :=
    vQuery := vSys.Query(
        'insert   into   pasien
(no_med_rec,      nama_pasien,
tgl_lahir,' +
        'jenis_kelamin,
alamat, pendidikan, pekerjaan,
agama) values (' +
        CEsc  +  ANoMedRec  +
CEsc + ',' +
        CEsc                +
vSys.DBAddLessString(ANamaPasi
en) + CEsc + ',' +
        CEsc                +
FormatDateTime('yyyy-mm-
dd',ATglLahir) + CEsc + ',' +
        CEsc + AJenisKelamin +
CEsc + ',' +
        CEsc                +
vSys.DBAddLessString(AAlamat)
+ CEsc + ',' +
        CEsc  +  APendidikan  +
CEsc + ',' +
        CEsc  +  APekerjaan  +
CEsc + ',' +
        CEsc + AAgama + CEsc +
')'  ,Executed);
    if  (vQuery.RowsAffected  =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TIRI.CreateRuang(AIdRuang,
ANamaRuang: String;
  AKapasitas:         integer;
AIdKelas: String): Boolean;
```

```
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'insert   into   ruang
(id_ruang,         nama_ruang,
kapasitas, id_kelas) values ('
+
        CEsc + AIdRuang + CEsc
+ ',' +
        CEsc                 +
vSys.DBAddLessString(ANamaRuan
g) + CEsc + ',' +
        IntToStr(AKapasitas) +
',' +
        CEsc + AIdKelas + CEsc
+ ')' , Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TIRI.DeleteIRI(ANoDokIRI:
String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'delete from iri where
no_dok_iri = ' +
        CEsc   +   ANoDokIRI   +
CEsc, Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TIRI.DeleteKelas(AIdKelas:
String): Boolean;
var
  vSys : TSys;
  vQuery :TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'select  *  from  ruang
where id_kelas = ' +
        CEsc   +   AIdKelas   +
CEsc, Actived);
    If (vQuery.RecordCount >0)
Then
      begin
        ShowMessage('Data
Kelas Terpakai Oleh Data Ruang
!');
        Result := False;
        Exit;
      end;
    vQuery := vSys.Query(
        'delete   from   kelas
where id_kelas = ' +
        CEsc   +   AIdKelas   +
CEsc, Executed);
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;
```

```
function
TIRI.DeletePasien(ANoMedRec:
String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'delete   from   pasien
where no_med_rec = ' +
        CEsc  +  ANoMedRec  +
CEsc, Executed) ;
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TIRI.DeleteRuang(AIdRuang:
String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'Delete   from   ruang
where id_ruang = ' +
        CESc  +  AIdRuang  +
CEsc, Executed);
    If (vQuery.RowsAffected =
1) Then
      Result := True
    Else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

destructor TIRI.Destroy;
begin

  inherited;
end;

function
TIRI.GetDataSearchIRI(AFieldRe
f,     AKeyWord:     String):
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select   a.no_dok_iri
"No. Dok IRI" , ' +
        'a.no_med_rec "No. Med
Rec" , ' +
        'b.nama_pasien   "Nama
Pasien", ' +
        'a.id_ruang        "ID
Ruang", ' +
        'c.nama_ruang    "Nama
Ruang", ' +
        'a.tanggal_masuk "Tgl.
Masuk", ' +
        'a.tanggal_keluar
"Tgl. Keluar", ' +
        'a.cara_masuk    "Cara
Masuk" , ' +
        'a.cara_bayar    "Cara
Bayar" ' +
        'from iri a, pasien b,
ruang c ' +
        'where a.no_med_rec =
b.no_med_rec and a.id_ruang =
c.id_ruang ' +
        'and a.' + AFieldRef +
'  like  '  +  CEsc  +  '%'  +
AKeyWord + '%' + CEsc +
        'order              by
a.no_dok_iri'  ,Actived);
  finally
    FreeAndNil(vSys);
```

```
  end;
end;

function
TIRI.GetDataSearchPasien(AFiel
dRef,    AKeyWord:    String):
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select     no_med_rec
"No. Med Rec", ' +
        'nama_pasien     "Nama
Pasien" , ' +
        'tgl_lahir       "Tgl.
Lahir" , ' +
        'jenis_kelamin  "Jenis
Kelamin", ' +
        'alamat "Alamat" , ' +
        'pendidikan
"Pendidikan" , ' +
        'pekerjaan
"Pekerjaan", ' +
        'agama  "Agama"  from
pasien ' +
        'where ' + AFieldRef +
' like ' + CEsc + '%' +
        AKeyWord + '%' + CEsc
+ ' order by no_med_rec ' ,
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function
TIRI.GetDataSearchRuang(AField
Ref,    AKeyWord:    String):
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select     a.id_ruang
"ID.Ruang", ' +
        'a.nama_ruang    "Nama
Ruang" , ' +
        'a.kapasitas
"Kapasitas" , ' +
        'a.id_kelas "ID.Kelas"
, ' +
        'b.nama_kelas    "Nama
Kelas" ' +
        'from ruang a, kelas b
where ' +
        'a.' + AFieldRef + '
like ' + CEsc + '%' +
        AKeyWord + '%' + CEsc
+   '   and   a.id_kelas   =
b.id_kelas ' +
        ' order by a.id_ruang'
, Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function          TIRI.GetIRI:
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select   a.no_dok_iri
"No. Dok IRI" , ' +
        'a.no_med_rec "No. Med
Rec" , ' +
        'b.nama_pasien   "Nama
Pasien", ' +
        'a.id_ruang        "ID
Ruang", ' +
        'c.nama_ruang    "Nama
Ruang", ' +
        'a.tanggal_masuk "Tgl.
Masuk", ' +
        'a.tanggal_keluar
"Tgl. Keluar", ' +
        'a.cara_masuk    "Cara
Masuk" , ' +
```

```
        'a.cara_bayar    "Cara
Bayar" ' +
        'from iri a, pasien b,
ruang c ' +
        'where  a.no_med_rec  =
b.no_med_rec and a.id_ruang =
c.id_ruang ' +
        'order              by
a.no_dok_iri'  ,Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function        TIRI.GetKelas:
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select       id_kelas
"ID.Kelas" , nama_kelas "Nama
Kelas" from kelas ' +
        'order by id_kelas ' ,
Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function       TIRI.GetPasien:
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
     Result := vSys.Query(
        'select     no_med_rec
"No. Med Rec", ' +
        'nama_pasien     "Nama
Pasien" , ' +
        'tgl_lahir       "Tgl.
Lahir" , ' +
        'jenis_kelamin  "Jenis
Kelamin", ' +
        'alamat "Alamat" , ' +
        'pendidikan
"Pendidikan" , ' +
        'pekerjaan
"Pekerjaan", ' +
        'agama   "Agama"   from
pasien ' +
        'order by no_med_rec '
, Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

function        TIRI.GetRuang:
TADOQuery;
var
  vSys : TSys;
begin
  try
    vSys := TSys.Create;
    Result := vSys.Query(
        'select     a.id_ruang
"ID.Ruang", ' +
        'a.nama_ruang    "Nama
Ruang" , ' +
        'a.kapasitas
"Kapasitas" , ' +
        'a.id_kelas "ID.Kelas"
, ' +
        'b.nama_kelas    "Nama
Kelas" ' +
        'from ruang a, kelas b
' +
        '  where  a.id_kelas  =
b.id_kelas ' +
        'order  by  a.id_ruang'
, Actived);
  finally
    FreeAndNil(vSys);
  end;
end;

procedure
TIRI.SetCaraBayar(const Value:
String);
begin
  FCaraBayar := Value;
```

```
end;

procedure
TIRI.SetCaraMasuk(const Value:
String);
begin
  FCaraMasuk := Value;
end;

procedure
TIRI.SetIdRuang(const   Value:
String);
begin
  FIdRuang := Value;
end;

procedure
TIRI.SetNoDokIRI(const  Value:
String);
begin
  FNoDokIRI := Value;
end;

procedure
TIRI.SetNoMedRec(const  Value:
String);
begin
  FNoMedRec := Value;
end;

procedure
TIRI.SetTglKeluar(const Value:
TDateTime);
begin
  FTglKeluar := Value;
end;

procedure
TIRI.setTglMasuk(const  Value:
TDateTime);
begin
  FTglMasuk := Value;
end;

function
TIRI.UpdateIRI(ANoDokIRI,
ANoMedRec,  AIdRuang:  String;
ATglMasuk,
  ATglKeluar   :   TDateTime;
ACaraMasuk,        ACaraBayar:
String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'update iri set ' +
        ' id_ruang = ' + CEsc
+ AIdRuang + CEsc + ',' +
        ' tanggal_masuk = ' +
CEsc  +  FormatDateTime('yyyy-
mm-dd',ATglMasuk) + CEsc + ','
+
        ' tanggal_keluar = ' +
CEsc  +  FormatDateTime('yyyy-
mm-dd',ATglKeluar)  +  CEsc  +
',' +
        '  cara_masuk  =  '  +
CEsc + ACaraMasuk + CEsc + ','
+
        '  cara_bayar  =  '  +
CEsc + ACaraBayar + CEsc +
        ' where no_dok_iri = '
+ CEsc  +  ANoDokIRI  +  CEsc  ,
Executed);
    if  (vQuery.RowsAffected  =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TIRI.UpdateKelas(AIdKelas,
ANamaKelas: String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
```

```
    vSys := Tsys.Create;
    vQuery := vSys.Query(
        'update   kelas   set
nama_kelas  =  '  +  CEsc  +
vSys.DBAddLessString(ANamaKela
s) + CEsc +
        ' where id_kelas = ' +
CEsc  +  AIdKelas  +  CEsc  ,
Executed);
    if  (vQuery.RowsAffected  =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TIRI.UpdatePasien(ANoMedRec,
ANamaPasien : String;
        ATglLahir : TDateTime;
AJenisKelamin, AAlamat,
        APendidikan,
APekerjaan, AAgama : String):
Boolean ;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'update pasien set ' +
        '  nama_pasien  =  '  +
CEsc                          +
vSys.DBAddLessString(ANamaPasi
en) + CEsc + ',' +
        ' tgl_lahir = ' + CEsc
+ FormatDateTime('yyyy-mm-dd',
ATglLahir) + CEsc  + ',' +
        ' jenis_kelamin = ' +
CEsc + AJenisKelamin + CEsc +
',' +
        ' alamat = ' + CEsc +
vSys.DBAddLessString(AAlamat)
+ CEsc + ',' +
        '  pendidikan  =  '  +
CEsc  +  APendidikan  +  CEsc  +
',' +
        ' pekerjaan = ' + CEsc
+ APekerjaan + CEsc + ',' +
        ' agama = ' + CEsc +
AAgama + CEsc +
        ' where no_med_rec = '
+  CEsc  +  ANoMedRec  +  CEsc  ,
Executed );
    if  (vQuery.RowsAffected  =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

function
TIRI.UpdateRuang(AIdRuang,
ANamaRuang: String;
  AKapasitas:         integer;
AIdKelas: String): Boolean;
var
  vSys : TSys;
  vQuery : TADOQuery;
begin
  try
    vSys := TSys.Create;
    vQuery := vSys.Query(
        'update ruang set ' +
        '  nama_ruang  =  '  +
CEsc                          +
vSys.DBAddLessString(ANamaRuan
g) + CEsc + ',' +
        '  kapasitas  =  '  +
IntToStr(AKapasitas) + ',' +
        ' id_kelas = ' + CEsc
+ AIdKelas + CEsc +
        ' where id_ruang = ' +
CEsc   +   AIdRuang   +   CEsc,
Executed );
```

```
    if (vQuery.RowsAffected =
1) then
      Result := True
    else
      Result := False;
  finally
    FreeAndNil(vQuery);
    FreeAndNil(vSys);
  end;
end;

end.

unit UGantiUserName;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls,
ADODB, StrUtils;

type
  TFGantiUserName            =
class(TForm)
    PanelUbahUserName: TPanel;
    LblStatusPassword: TLabel;
    LblPassword: TLabel;
    LblUserNameBaru: TLabel;
    LblStatusUserName: TLabel;
    EdPasswordSekarang: TEdit;
    EdUserNameBaru: TEdit;
    BtnUbah: TButton;
    BtnBatal: TButton;
    procedure
EdUserNameBaruKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure
EdPasswordSekarangKeyUp(Sender
: TObject; var Key: Word;
      Shift: TShiftState);
    procedure
BtnBatalClick(Sender:
TObject);
    procedure FormShow(Sender:
TObject);
    procedure
BtnUbahClick(Sender: TObject);
  private
    { Private declarations }
    function
CheckValiditas(APasswordValid,
AUserValid   :   Boolean)   :
Boolean;
  public
    { Public declarations }
    procedure
UbahUserName(ANIP : String);
  end;

var
  FGantiUserName:
TFGantiUserName;
  vNIP : String;
  vPasswordLamaValid,
vUserNameValid : Boolean;

implementation

uses UPegawai, UProfileUser;

{$R *.dfm}

procedure
TFGantiUserName.UbahUserName(A
NIP: String);
begin
  vNIP := ANIP;
end;

procedure
TFGantiUserName.EdUserNameBaru
KeyUp(Sender: TObject;
  var   Key:   Word;   Shift:
TShiftState);
var
  vData : TPegawai;
  vQueryTemp : TADOQuery;
begin
  try
    vData := TPegawai.Create;
```

```
    If
(Length(Trim(EdUserNameBaru.Te
xt))=0) Then
      begin

LblStatusUserName.Caption   :=
'Masukkan UserName Baru Anda
:' ;
        vUserNameValid      :=
False;

LblStatusUserName.Font.Color
:= clBlue;

FreeAndNil(vQueryTemp);
        FreeAndNil(vData);
        Exit;
      end;
    vQueryTemp              :=
vData.GetDataFinalSearchUser('
username',
EdUserNameBaru.Text);
    If (vQueryTemp.RecordCount
>0) Then
      begin

LblStatusUserName.Caption   :=
'Username Sudah Terpakai !';

LblStatusUserName.Font.Color
:= clRed;
        vUserNameValid      :=
False;
      end
    Else
      begin

LblStatusUserName.Caption   :=
'Username Tersedia';

LblStatusUserName.Font.Color
:= clGreen;
        vUserNameValid      :=
True;
      end;
  finally
    FreeAndNil(vQueryTemp);
    FreeAndNil(vData);
  end;
end;

procedure
TFGantiUserName.EdPasswordSeka
rangKeyUp(Sender: TObject;
  var   Key:   Word;   Shift:
TShiftState);
var
  vData : TPegawai;
  vQueryTemp : TADOQuery;
  vPasswordLama : String;
begin
  try
    vData := TPegawai.Create;
    If
(Length(Trim(EdPasswordSekaran
g.Text))=0) Then
      begin

LblStatusPassword.Caption   :=
'Masukkan    Password    Anda
Sekarang :' ;

LblStatusPassword.Font.Color
:= clBlue;
        vPasswordLamaValid  :=
False;

FreeAndNil(vQueryTemp);
        FreeAndNil(vData);
        Exit;
      end;
    vQueryTemp              :=
vData.GetDataFinalSearchUser('
nip', vNIP);
    vPasswordLama           :=
vQueryTemp.Fields[2].Text;
    If
(EdPasswordSekarang.Text    <>
vPasswordLama) Then
      begin

LblStatusPassword.Caption   :=
'Password Tidak Cocok !';

LblStatusPassword.Font.Color
:= clRed;
```

```
        vPasswordLamaValid  :=
False;
      end
    Else
      begin

LblStatusPassword.Caption   :=
'Password Cocok';

LblStatusPassword.Font.Color
:= clGreen;
        vPasswordLamaValid  :=
True;
      end;
  finally
    FreeAndNil(vQueryTemp);
    FreeAndNil(vData);
  end;
end;

procedure
TFGantiUserName.BtnBatalClick(
Sender: TObject);
begin
  vUserNameValid := False;
  vPasswordLamaValid := False;
  Close;
end;

procedure
TFGantiUserName.FormShow(Sende
r: TObject);
begin
  EdUserNameBaru.Clear;
  EdPasswordSekarang.Clear;
  EdPasswordSekarang.SetFocus;
  LblStatusPassword.Caption :=
'Mohon Masukkan Password Anda
Sekarang : ';
  LblStatusUserName.Caption :=
'Masukkan UserName Baru Anda :
';
  LblStatusPassword.Font.Color
:= clBlack;
  LblStatusUserName.Font.Color
:= clBlack;
  vUserNameValid := False;
  vPasswordLamaValid := False;
end;

procedure
TFGantiUserName.BtnUbahClick(S
ender: TObject);
var
  vData : TPegawai;
begin
  try
    vData := TPegawai.Create;
    If
(CheckValiditas(vPasswordLamaV
alid, vUserNameValid)) Then
      begin
        If
vData.UpdateProfileUser(
            vNIP,

EdUserNameBaru.Text,

EdPasswordSekarang.Text) Then

ShowMessage('UserName Berhasil
Diubah !' )
        Else

ShowMessage('UserName    Gagal
Diubah !');
        BtnBatalClick(Self);

FProfileUser.FormShow(Nil);
      end
    Else
      begin
        ShowMessage('Kesalahan
Dalam Pengubahan Data, Mohon
Diperbaiki !');
      end;
  finally
    FreeAndNil(vData);
  end;
end;

function
TFGantiUserName.CheckValiditas
(APasswordValid,
  AUserValid:        Boolean):
Boolean;
```

```
begin
  If (APasswordValid = True)
and (AUserValid = True) Then
    Result := True
  Else
    Result := False;
end;

end.

unit UGantiPassword;

interface

uses
  Windows, Messages, SysUtils,
Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls,
ADODB, StrUtils;

type
  TFGantiPassword           =
class(TForm)
    PanelUbahUserName: TPanel;
    LblStatusPassword: TLabel;
    LblPassword: TLabel;
    LblPasswordBaru: TLabel;
    LblStatusPasswordBaru:
TLabel;
    EdPasswordSekarang: TEdit;
    EdPasswordBaru1: TEdit;
    BtnUbah: TButton;
    BtnBatal: TButton;
    LblKet: TLabel;
    EdPasswordBaru2: TEdit;
    procedure FormShow(Sender:
TObject);
    procedure
BtnBatalClick(Sender:
TObject);
    procedure
BtnUbahClick(Sender: TObject);
    procedure
EdPasswordBaru2KeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure
EdPasswordSekarangKeyUp(Sender
: TObject; var Key: Word;
      Shift: TShiftState);
  private
    { Private declarations }
    function
CheckValiditas(APasswordLamaVa
lid,   APasswordBaruValid   :
Boolean) : Boolean;
  public
    { Public declarations }
    procedure
UbahPassword(ANIP : String);
  end;

var
  FGantiPassword:
TFGantiPassword;
  vNIP : String;
  vPasswordLamaValid,
vPasswordBaruValid : Boolean;

implementation

uses UPegawai, UProfileUser;

{$R *.dfm}

{ TFGantiPassword }

function
TFGantiPassword.CheckValiditas
(APasswordLamaValid,
  APasswordBaruValid:
Boolean): Boolean;
begin
  If   (APasswordLamaValid   =
True) and (APasswordBaruValid
= True) Then
    Result := True
  Else
    Result := False;
end;

procedure
TFGantiPassword.UbahPassword(A
NIP: String);
```

```
begin
  vNIP := ANIP;
end;

procedure
TFGantiPassword.FormShow(Sende
r: TObject);
begin
  LblStatusPassword.Caption :=
'Mohon Masukkan Password Anda
Sekarang : ';

LblStatusPasswordBaru.Caption
:= 'Masukkan Password Baru
Anda 1x Lagi : ' ;
  LblStatusPassword.Font.Color
:= clBlack;

LblStatusPasswordBaru.Font.Col
or := clBlack;
  EdPasswordBaru1.Clear;
  EdPasswordBaru2.Clear;
  EdPasswordSekarang.Clear;
  EdPasswordSekarang.SetFocus;
  vPasswordLamaValid := False;
  vPasswordBaruValid := False;
end;

procedure
TFGantiPassword.BtnBatalClick(
Sender: TObject);
begin
  vPasswordLamaValid := False;
  vPasswordBaruValid := False;
  Close;
end;

procedure
TFGantiPassword.BtnUbahClick(S
ender: TObject);
var
  vData : TPegawai;
begin
  try
    vData := TPegawai.Create;
    If
(CheckValiditas(vPasswordLamaV
alid,     vPasswordBaruValid))
Then
      begin
        If
vData.UpdateProfileUser(
            vNIP,

FProfileUser.EdUserName.Text,

EdPasswordBaru1.Text) Then

ShowMessage('Password Berhasil
Diubah !' )
        Else

ShowMessage('Password    Gagal
Diubah !');
        BtnBatalClick(Self);

FProfileUser.FormShow(Nil);
      end
    Else
      begin
        ShowMessage('Kesalahan
Dalam Pengubahan Data, Mohon
Diperbaiki !');
      end;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFGantiPassword.EdPasswordBaru
2KeyUp(Sender: TObject;
  var   Key:   Word;   Shift:
TShiftState);
begin
  vPasswordBaruValid := False;
  If
Length(EdPasswordBaru2.Text) =
0 Then
    begin

LblStatusPasswordBaru.Caption
:= 'Masukkan Password Anda 1x
Lagi :' ;
```

```
LblStatusPasswordBaru.Font.Col
or := clBlue;
      vPasswordBaruValid    :=
False;
      Exit;
    end;
  If  (EdPasswordBaru2.Text  =
EdPasswordBaru1.Text) Then
    begin

LblStatusPasswordBaru.Caption
:= 'Password Sesuai' ;

LblStatusPasswordBaru.Font.Col
or := clGreen;
      vPasswordBaruValid    :=
True;
    end
  Else
    begin

LblStatusPasswordBaru.Caption
:= 'Password Tidak Sama' ;

LblStatusPasswordBaru.Font.Col
or := clRed;
      vPasswordBaruValid    :=
False;
    end;
end;

procedure
TFGantiPassword.EdPasswordSeka
rangKeyUp(Sender: TObject;
  var   Key:   Word;   Shift:
TShiftState);
var
  vData : TPegawai;
  vQueryTemp : TADOQuery;
  vPasswordLama : String;
begin
  try
    vData := TPegawai.Create;
    If
(Length(Trim(EdPasswordSekaran
g.Text))=0) Then
      begin
LblStatusPassword.Caption   :=
'Masukkan    Password    Anda
Sekarang :' ;

LblStatusPassword.Font.Color
:= clBlue;
        vPasswordLamaValid  :=
False;

FreeAndNil(vQueryTemp);
        FreeAndNil(vData);
        Exit;
      end;
    vQueryTemp              :=
vData.GetDataFinalSearchUser('
nip', vNIP);
    vPasswordLama           :=
vQueryTemp.Fields[2].Text;
    If
(EdPasswordSekarang.Text    <>
vPasswordLama) Then
      begin

LblStatusPassword.Caption   :=
'Password Tidak Cocok !';

LblStatusPassword.Font.Color
:= clRed;
        vPasswordLamaValid  :=
False;
      end
    Else
      begin

LblStatusPassword.Caption   :=
'Password Cocok';

LblStatusPassword.Font.Color
:= clGreen;
        vPasswordLamaValid  :=
True;
      end;
  finally
    FreeAndNil(vQueryTemp);
    FreeAndNil(vData);
  end;
end;
```

```
end.

unit UDMReport;

interface

uses
  SysUtils,    Classes,    DB,
ADODB;

type
  TDataModuleReport          =
class(TDataModule)
    ADOReport: TADOConnection;
    TPasien: TADOTable;
    QIRI: TADOQuery;
    QIRIno_dok_iri:
TStringField;
    QIRInama_pasien:
TStringField;
    QIRInama_ruang:
TStringField;
    QIRItanggal_masuk:
TDateField;
    QIRItanggal_keluar:
TDateField;
    QPermintaanMakanan:
TADOQuery;

QPermintaanMakananid_permintaa
n_makanan: TStringField;

QPermintaanMakananno_dok_iri:
TStringField;

QPermintaanMakananid_ruang:
TStringField;

QPermintaanMakanannama_ruang:
TStringField;

QPermintaanMakanannama_kelas:
TStringField;

QPermintaanMakanannama_pasien:
TStringField;

QPermintaanMakanannama_makanan
: TStringField;

QPermintaanMakanantanggal_perm
intaan_makanan: TDateField;

QPermintaanMakanannama_pegawai
: TStringField;

QPermintaanMakananketerangan:
TStringField;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  DataModuleReport:
TDataModuleReport;

implementation

{$R *.dfm}

end.

unit UDataUser;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs,  StdCtrls,  DBCtrls,
Grids,    DBGrids,    ExtCtrls,
ComCtrls, ToolWin,
  DB, ImgList, XPMan, ADODB;
type
  TModeData   =   (BrowseData,
UpdateData,        DeleteData,
SearchData);
type
  TFDataUser = class(TForm)
    XPStyle: TXPManifest;
    ImageList: TImageList;
```

```
    DSDataUser: TDataSource;
    ToolBar: TToolBar;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSeparator2: TToolButton;
    TBRefresh: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    PanelDetail: TPanel;
    LblNIP: TLabel;
    LblUserName: TLabel;
    EdNIP: TEdit;
    EdUserName: TEdit;
    PanelDBGrid: TPanel;
    DBGridDataUser: TDBGrid;
    DBControl: TDBNavigator;
    PanelCari: TPanel;
    LblFieldCari: TLabel;
    LblKataKunci: TLabel;
    CBFieldCari: TComboBox;
    EdKataKunci: TEdit;
    TBUbah: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    LblHakAkses: TLabel;
    CBHakAkses: TComboBox;
    StatusBar: TStatusBar;
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
TBBatalClick(Sender: TObject);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
DBGridDataUserCellClick(Column
: TColumn);
    procedure
DBControlClick(Sender:
TObject;               Button:
TNavigateBtn);
    procedure
TBKeluarClick(Sender:
TObject);
    procedure FormShow(Sender:
TObject);
    procedure
CBFieldCariClick(Sender:
TObject);
    procedure
CBFieldCariKeyPress(Sender:
TObject; var Key: Char);
    procedure
CBHakAksesKeyPress(Sender:
TObject; var Key: Char);
    procedure
EdKataKunciKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
  private
    { Private declarations }
    procedure
ClearFieldData();
    procedure
SetPropertyGridUser();
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewDetail();
    procedure RefreshData();
    procedure
SetStatusBar(AModeData       :
TModeData);
  public
    { Public declarations }
  end;

var
  FDataUser: TFDataUser;
  vQuery : TADOQuery;
  vModeData : TModeData;

implementation

uses UPegawai;

{$R *.dfm}

{ TFDataUser }
```

```
procedure
TFDataUser.ClearFieldData;
begin
  EdNIP.Clear;
  EdUserName.Clear;
  CBHakAkses.Text := '';
  EdKataKunci.Clear;
end;

procedure
TFDataUser.RefreshData;
var
  vData : TPegawai;
begin
  try
    vData := TPegawai.Create;
    vQuery := vData.GetUser;
    DSDataUser.DataSet      :=
vQuery;
    DBGridDataUser.DataSource
:= DSDataUser;
    DBControl.DataSource    :=
DSDataUser;
    SetKontrol(False);
    vModeData := BrowseData;
    ClearFieldData;
    ViewDetail;
    SetStatusBar(vModeData);
    SetPropertyGridUser;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataUser.SetKontrol(ABool:
Boolean);
begin
  PanelDetail.Enabled       :=
ABool;
  PanelDBGrid.Enabled  :=  Not
ABool;
  PanelCari.Enabled   :=   Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBRefresh.Enabled   :=   Not
ABool;
end;

procedure
TFDataUser.SetPropertyGridUser
;
var
  vCount : Integer;
begin
  With DBGridDataUser do
    begin
      for vCount :=0 to 3 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
      Columns[0].Width := 100;
      Columns[1].Width := 160;
      Columns[3].Width := 100;
      Columns[2].Visible    :=
False;
      Columns[0].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;

procedure
TFDataUser.SetStatusBar(AModeD
ata: TModeData);
begin
  If (AModeData = UpdateData)
Then
    StatusBar.Panels[0].Text
:= 'Update Mode - Pengubahan
Data'
  Else    If    (AModeData    =
BrowseData) Then
```

```
    StatusBar.Panels[0].Text
:= 'Browse Mode - Penelusuran
Data'
  Else    If    (AModeData    =
DeleteData) Then
    StatusBar.Panels[0].Text
:= 'Delete Mode - Penghapusan
Data'
  Else    If    (AModeData    =
SearchData) Then
    StatusBar.Panels[0].Text
:= 'Search Mode - Pencarian
Data'
  Else
    StatusBar.Panels[0].Text
:= 'Unknown Mode';
  StatusBar.Panels[1].Text  :=
'Data     User     Ke-'     +
IntToStr(vQuery.RecNo)   +   '
Dari            '            +
IntToStr(vQuery.RecordCount) +
' Data';
end;

procedure
TFDataUser.ViewDetail;
begin
  EdNIP.Text                :=
DBGridDataUser.Fields[0].Text;
  EdUserName.Text           :=
DBGridDataUser.Fields[1].Text;
  CBHakAkses.Text           :=
DBGridDataUser.Fields[3].Text;
  If     CBHakAkses.Text     =
'Administrator' Then
    TBHapus.Enabled := False
  Else
    TBHapus.Enabled := True;
end;

procedure
TFDataUser.TBUbahClick(Sender:
TObject);
begin
  If (vModeData = BrowseData)
Then
    vModeData := UpdateData;
  SetKontrol(True);
  SetStatusBar(vModeData);
end;

procedure
TFDataUser.TBHapusClick(Sender
: TObject);
var
  vData : TPegawai;
begin
  vModeData := DeleteData;
  SetStatusBar(vModeData);
  If  MessageDlg('Anda  Yakin
Menghapus Data User Dengan NIP
: ' +
      EdNIP.Text  +  '  ?'  ,
mtConfirmation,        [mbYes,
mbNo],0) = mrYes Then
    begin
      try
        vData               :=
TPegawai.Create;
        If
vData.DeleteUser(EdNIP.Text)
Then
          ShowMessage('Data
User Berhasil Dihapus !')
        Else
          ShowMessage('Data
User Gagal Dihapus !');
        RefreshData;
      finally
        FreeAndNil(vData);
      end;
    end
  else
    begin
      vModeData := BrowseData;
      SetStatusBar(vModeData);
    end;
end;

procedure
TFDataUser.TBSimpanClick(Sende
r: TObject);
var
  vData : TPegawai;
begin
```

```
  If (vModeData = UpdateData)
or (vModeData = SearchData)
Then
    begin
      If(MessageDlg('Apakah
Anda Yakin Ingin Mengubah Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData          :=
TPegawai.Create;
            If
(vData.UpdateUser(
                EdNIP.Text,

CBHakAkses.Text)) Then

ShowMessage('Data         User
Berhasil Diubah !')
            Else

ShowMessage('Data  User  Gagal
Diubah !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end;
  RefreshData;
end;

procedure
TFDataUser.TBBatalClick(Sender
: TObject);
begin
    If      (vModeData      =
UpdateData) Then
    begin
      vModeData := BrowseData;
      ViewDetail;
      SetKontrol(False);
      SetStatusBar(vModeData);
    end
  Else     If(vModeData     =
SearchData) Then
    begin
      ViewDetail;
      SetKontrol(False);
      TBUbah.Enabled := True;
      TBHapus.Enabled := True;
    end;
end;

procedure
TFDataUser.TBRefreshClick(Send
er: TObject);
begin
  RefreshData;
end;

procedure
TFDataUser.DBGridDataUserCellC
lick(Column: TColumn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataUser.DBControlClick(Send
er:      TObject;      Button:
TNavigateBtn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataUser.TBKeluarClick(Sende
r: TObject);
begin
  Close;
end;

procedure
TFDataUser.FormShow(Sender:
TObject);
begin
  RefreshData;
end;
```

```
procedure
TFDataUser.CBFieldCariClick(Se
nder: TObject);
begin
  EdKataKunci.Clear;
EdKataKunci.SetFocus;
end;

procedure
TFDataUser.CBFieldCariKeyPress
(Sender:  TObject;  var  Key:
Char);
begin
  Key := #0;
end;

procedure
TFDataUser.CBHakAksesKeyPress(
Sender:  TObject;  var  Key:
Char);
begin
  Key := #0;
end;

procedure
TFDataUser.EdKataKunciKeyUp(Se
nder: TObject; var Key: Word;
  Shift: TShiftState);
var
  vData : TPegawai;
begin
  try
    vData := TPegawai.Create;
    If (Trim(CBFieldCari.Text)
= '') Then
      begin
        FreeAndNil(vData);
        Exit;
      end;
    vQuery                  :=
vData.GetDataSearchUser(CBFiel
dCari.Text, EdKataKunci.Text);
    DSDataUser.DataSet      :=
vQuery;
    DBGridDataUser.DataSource
:= DSDataUser;
    DBControl.DataSource    :=
DSDataUser;
    vModeData := SearchData;
    SetKontrol(False);
    ViewDetail;
    SetStatusBar(vModeData);
    SetPropertyGridUser;
    If  (vQuery.RecordCount  =
0) Then
      begin
        TBUbah.Enabled      :=
False;
        TBHapus.Enabled     :=
False;
      end
    Else
      begin
        TBUbah.Enabled      :=
True;
        TBHapus.Enabled     :=
True;
      end;
  finally
    FreeAndNil(vData);
  end;
end;

end.

unit UDataRuang;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs,   DBCtrls,   Grids,
DBGrids,  ComCtrls,  StdCtrls,
ExtCtrls, XPMan,
  ImgList, DB, ToolWin, ADODB,
StrUtils;

type
  TModeData   =   (BrowseData,
UpdateData,        InsertData,
DeleteData, SearchData);
type
  TFDataRuang = class(TForm)
    ToolBar: TToolBar;
```

```
    TBTambah: TToolButton;
    TBUbah: TToolButton;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
    TBRefresh: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    DSDataRuang: TDataSource;
    ImageList: TImageList;
    XPStyle: TXPManifest;
    PanelDetail: TPanel;
    PanelDBGrid: TPanel;
    DBGridDataRuang: TDBGrid;
    DBControl: TDBNavigator;
    LblIdRuang: TLabel;
    EdIdRuang: TEdit;
    LblNamaRuang: TLabel;
    EdNamaRuang: TEdit;
    LblKapasitas: TLabel;
    EdKapasitas: TEdit;
    LblIdKelas: TLabel;
    StatusBar: TStatusBar;
    PanelCari: TPanel;
    LblFieldCari: TLabel;
    LblKataKunci: TLabel;
    CBFieldCari: TComboBox;
    EdKataKunci: TEdit;
    CBKelas: TComboBox;
    procedure
TBKeluarClick(Sender:
TObject);
    procedure
DBControlClick(Sender:
TObject;               Button:
TNavigateBtn);
    procedure
DBGridDataRuangCellClick(Colum
n: TColumn);
    procedure
TBTambahClick(Sender:
TObject);
    procedure
CBKelasKeyPress(Sender:
TObject; var Key: Char);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
TBBatalClick(Sender: TObject);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
CBFieldCariClick(Sender:
TObject);
    procedure
EdKataKunciKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure
CBFieldCariKeyPress(Sender:
TObject; var Key: Char);
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
    procedure
SetPropertyGridRuang();
    procedure
ClearFieldData();
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewDetail();
    procedure RefreshData();
    procedure
SetStatusBar(AModeData       :
TModeData);
    procedure
LoadKelasToCombo();
    function
GetIdKelasFromCombo(ACombo   :
TComboBox) : String;
    function
CekValiditasInput() : Boolean;
  public
    { Public declarations }
  end;
```

```
var
  FDataRuang: TFDataRuang;
  vQuery : TADOQuery;
  vModeData : TModeData;

implementation

uses UIRI;

{$R *.dfm}

{ TFDataRuang }

procedure
TFDataRuang.ClearFieldData;
begin
  EdIdRuang.Clear;
  EdNamaRuang.Clear;
  EdKapasitas.Clear;
  CBKelas.Text := '';
  EdKataKunci.Clear;
end;

procedure
TFDataRuang.RefreshData;
var
  vData : TIRI;
begin
  try
    vData := TIRI.Create;
    vQuery := vData.GetRuang;
    DSDataRuang.DataSet     :=
vQuery;
    DBGridDataRuang.DataSource
:= DSDataRuang;
    DBControl.DataSource    :=
DSDataRuang;
    //FIRST LOAD - CHECK THE
STATUS DATA IF EMPTY OR ANY
DATA IS IN

//============================
============================
=====
    If  (vQuery.RecordCount  =
0) Then
      begin
        ShowMessage('Data
Ruang   Kosong,   Anda   Harus
Melakukan Penambahan Data !');
        ClearFieldData;
        SetKontrol(True);
        TBTambahClick(Nil);
        TBBatal.Enabled     :=
False;
      end
    else
      begin
        SetKontrol(False);
        vModeData           :=
BrowseData;
        ClearFieldData;
        ViewDetail;
      end;

//============================
============================
=====
    SetStatusBar(vModeData);
    SetPropertyGridRuang;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataRuang.SetKontrol(ABool:
Boolean);
begin
  PanelDetail.Enabled       :=
ABool;
  PanelDBGrid.Enabled  :=  Not
ABool;
  PanelCari.Enabled   :=   Not
ABool;
  TBTambah.Enabled    :=    Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBRefresh.Enabled   :=   Not
ABool;
```

```
  //TBCetak.Enabled   :=   Not
ABool;
end;

procedure
TFDataRuang.SetPropertyGridRua
ng;
var
  vCount : Integer;
begin
  With DBGridDataRuang do
    begin
      for vCount :=0 to 4 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
      Columns[0].Width := 60;
      Columns[1].Width := 120;
      Columns[3].Width := 60;
      Columns[0].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;

procedure
TFDataRuang.SetStatusBar(AMode
Data: TModeData);
begin
  If (AModeData = InsertData)
Then
    StatusBar.Panels[0].Text
:=  'Insert Mode - Penambahan
Data'
  Else    If    (AModeData    =
UpdateData) Then
    StatusBar.Panels[0].Text
:= 'Update Mode - Pengubahan
Data'
  Else    If    (AModeData    =
BrowseData) Then
    StatusBar.Panels[0].Text
:= 'Browse Mode - Penelusuran
Data'
  Else    If    (AModeData    =
DeleteData) Then
    StatusBar.Panels[0].Text
:= 'Delete Mode - Penghapusan
Data'
  Else    If    (AModeData    =
SearchData) Then
    StatusBar.Panels[0].Text
:=  'Search  Mode  -  Pencarian
Data'
  Else
    StatusBar.Panels[0].Text
:= 'Unknown Mode';
  StatusBar.Panels[1].Text  :=
'Data     Ruang     Ke-'     +
IntToStr(vQuery.RecNo)   +   '
Dari            '            +
IntToStr(vQuery.RecordCount) +
' Data';
end;

procedure
TFDataRuang.ViewDetail;
begin
  EdIdRuang.Text            :=
DBGridDataRuang.Fields[0].Text
;
  EdNamaRuang.Text          :=
DBGridDataRuang.Fields[1].Text
;
  If (vQuery.RecordCount = 0)
Then
    EdKapasitas.Text := '0';
  EdKapasitas.Text          :=
DBGridDataRuang.Fields[2].Text
;
  CBKelas.Text              :=
DBGridDataRuang.Fields[3].Text
+ '-' +

DBGridDataRuang.Fields[4].Text
;
end;
```

```
procedure
TFDataRuang.TBKeluarClick(Send
er: TObject);
begin
  Close;
end;

procedure
TFDataRuang.DBControlClick(Sen
der: TObject;
  Button: TNavigateBtn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataRuang.DBGridDataRuangCel
lClick(Column: TColumn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataRuang.TBTambahClick(Send
er: TObject);
begin
  ClearFieldData;
  vModeData := InsertData;
  SetKontrol(True);
  EdIdRuang.Enabled := True;
  EdIdRuang.SetFocus;
  SetStatusBar(vModeData);
end;

procedure
TFDataRuang.LoadKelasToCombo;
var
  vData : TIRI;
  vQueryTemp : TADOQuery;
begin
  try
    vData := TIRI.Create;
    vQueryTemp              :=
vData.GetKelas;
    CBKelas.Clear;
    While  Not  vQueryTemp.Eof
Do
      begin

CBKelas.Items.Add(vQueryTemp.F
ields[0].Text    +    '-'    +
vQueryTemp.Fields[1].Text);
        vQueryTemp.Next;
      end;
  finally
    FreeAndNil(vQueryTemp);
    FreeAndNil(vData);
  end;
end;

procedure
TFDataRuang.CBKelasKeyPress(Se
nder: TObject; var Key: Char);
begin
  Key := #0;
end;

procedure
TFDataRuang.TBUbahClick(Sender
: TObject);
begin
  If (vModeData = BrowseData)
Then
    vModeData := UpdateData;
  SetKontrol(True);
  EdIdRuang.Enabled := False;
  SetStatusBar(vModeData);
end;

procedure
TFDataRuang.TBHapusClick(Sende
r: TObject);
var
  vData : TIRI;
begin
  vModeData := DeleteData;
  SetStatusBar(vModeData);
  If   MessageDlg('Anda   Yakin
Menghapus  Data  Ruang  Dengan
ID. Ruang : ' +
      EdIdRuang.Text + ' ?' ,
mtConfirmation,        [mbYes,
mbNo],0) = mrYes Then
```

```
    begin
      try
        vData := TIRI.Create;
        If
vData.DeleteRuang(EdIdRuang.Te
xt) Then
          ShowMessage('Data
Ruang Berhasil Dihapus !')
        Else
          ShowMessage('Data
Ruang Gagal Dihapus !');
        RefreshData;
      finally
        FreeAndNil(vData);
      end;
    end
  else
    begin
      vModeData := BrowseData;
      SetStatusBar(vModeData);
    end;
end;

procedure
TFDataRuang.TBSimpanClick(Send
er: TObject);
var
  vData : TIRI;
begin
  If   not   CekValiditasInput
Then
    begin
      ShowMessage('Terjadi
Kesalahan  Dalam  Penginputan
Data, Mohon Diperbaiki');
      Exit;
    end;
  If (vModeData = InsertData)
Then
    begin
      If(MessageDlg('Apakah
Anda  Yakin  Ingin  Menambahkan
Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData           :=
TIRI.Create;
            If
(vData.CreateRuang(

EdIdRuang.Text,

EdNamaRuang.Text,

StrToInt(EdKapasitas.Text),

GetIdKelasFromCombo(CBKelas)))
Then

ShowMessage('Data        Ruang
Berhasil Ditambahkan !')
            Else

ShowMessage('Data  Ruang  Gagal
Ditambahkan !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end
  Else    If    (vModeData    =
UpdateData)  or  (vModeData  =
SearchData) Then
    begin
      If(MessageDlg('Apakah
Anda Yakin Ingin Mengubah Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData           :=
TIRI.Create;
            If
(vData.UpdateRuang(

EdIdRuang.Text,

EdNamaRuang.Text,

StrToInt(EdKapasitas.Text),
```

```
GetIdKelasFromCombo(CBKelas)))
Then

ShowMessage('Data        Ruang
Berhasil Diubah !')
            Else

ShowMessage('Data  Ruang  Gagal
Diubah !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end;
  RefreshData;
end;

function
TFDataRuang.GetIdKelasFromComb
o(ACombo : TComboBox): String;
var
  vPos : integer;
begin
  if ACombo.Text <> '' then
    begin
      vPos      :=      Pos('-
',ACombo.Text);
      Result                :=
LeftStr(ACombo.Text,vPos-1);
    end
  else
    Result := '';
end;

procedure
TFDataRuang.TBBatalClick(Sende
r: TObject);
begin
  If (vModeData = InsertData)
or (vModeData =   UpdateData)
Then
    begin
      vModeData := BrowseData;
      ViewDetail;
      SetKontrol(False);
      SetStatusBar(vModeData);
    end
  Else     If(vModeData     =
SearchData) Then
    begin
      ViewDetail;
      SetKontrol(False);
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := True;
      TBHapus.Enabled := True;
    end;
end;

procedure
TFDataRuang.TBRefreshClick(Sen
der: TObject);
begin
  RefreshData;
end;

procedure
TFDataRuang.CBFieldCariClick(S
ender: TObject);
begin
  EdKataKunci.Text := '';
  EdKataKunci.SetFocus;
end;

procedure
TFDataRuang.EdKataKunciKeyUp(S
ender: TObject; var Key: Word;
  Shift: TShiftState);
var
  vData : TIRI;
begin
  try
    vData := TIRI.Create;
    If (Trim(CBFieldCari.Text)
= '') Then
      Exit;
    vQuery                  :=
vData.GetDataSearchRuang(CBFie
ldCari.Text,
EdKataKunci.Text);
    DSDataRuang.DataSet     :=
vQuery;
```

```
    DBGridDataRuang.DataSource
:= DSDataRuang;
    DBControl.DataSource    :=
DSDataRuang;
    vModeData := SearchData;
    SetKontrol(False);
    ViewDetail;
    SetStatusBar(vModeData);
    SetPropertyGridRuang;
    TBTambah.Enabled := False;
    If  (vQuery.RecordCount  =
0) Then
      begin
        TBUbah.Enabled      :=
False;
        TBHapus.Enabled     :=
False;
      end
    Else
      begin
        TBUbah.Enabled      :=
True;
        TBHapus.Enabled     :=
True;
      end;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataRuang.CBFieldCariKeyPres
s(Sender:  TObject;  var  Key:
Char);
begin
  Key := #0;
end;

function
TFDataRuang.CekValiditasInput:
Boolean;
begin
  If(Trim(EdIdRuang.Text)    =
'') or (Trim(EdNamaRuang.Text)
= '') Then
    Result := False
  Else
    Result := True;
end;

procedure
TFDataRuang.FormShow(Sender:
TObject);
begin

LoadKelasToCombo;RefreshData;
end;

end.

unit UDataPermintaanMakanan;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs,    ComCtrls,    DB,
ImgList,   XPMan,   StdCtrls,
DBCtrls, Grids, DBGrids,
  ExtCtrls,  ToolWin,  ADODB,
StrUtils, Buttons;

type
  TModeData                  =
(BrowseDataMaster,
UpdateDataMaster,
InsertDataMaster,
   DeleteDataMaster,
SearchDataMaster,
UpdateDataDetail,
InsertDataDetail,
   DeleteDataDetail,
SearchDataDetail);
type
  TFDataPermintaanMakanan    =
class(TForm)
    ToolBar: TToolBar;
    TBTambah: TToolButton;
    TBUbah: TToolButton;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
```

```
    TBRefresh: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    PanelMaster: TPanel;
    DBGridDataPermintaan:
TDBGrid;
    DBControlMaster:
TDBNavigator;
    PanelCariMaster: TPanel;
    LblFieldCariMaster:
TLabel;
    LblKataKunciMaster:
TLabel;
    CBFieldCariMaster:
TComboBox;
    EdKataKunciMaster: TEdit;
    XPStyle: TXPManifest;
    ImageList: TImageList;
    DSDataPermintaan:
TDataSource;
    StatusBar: TStatusBar;
    DTTglPermintaan:
TDateTimePicker;
    LblTglPermintaa: TLabel;
    LblIdPermintaan: TLabel;
    EdIdPermintaan: TEdit;
    LblIdPengurus: TLabel;
    EdIdPengurus: TEdit;
    PanelDetail: TPanel;

DBGridDataDetailPermintaan:
TDBGrid;
    DBControlDetail:
TDBNavigator;
    LblNoDokIRI: TLabel;
    EdNoDokIRI: TEdit;
    BtnPilihIRI: TButton;
    LblIdMakanan: TLabel;
    EdIdMakanan: TEdit;
    EdIdDetailPermintaan:
TEdit;
    LblKeterangan: TLabel;
    EdKeterangan: TMemo;
    BtnTambahDetail: TButton;
    BtnUbahDetail: TButton;
    BtnSimpanDetail: TButton;
    BtnBatalDetail: TButton;
    BtnHapusDetail: TButton;
    PanelCariDetail: TPanel;
    LblFieldCariDetail:
TLabel;
    LblKataKunciDetail:
TLabel;
    CBFieldCariDetail:
TComboBox;
    EdKataKunciDetail: TEdit;
    BtnPilihMakanan: TButton;
    DSDataDetailPermintaan:
TDataSource;
    BtnRefreshDetail: TBitBtn;
    procedure
TBKeluarClick(Sender:
TObject);
    procedure
BtnPilihIRIClick(Sender:
TObject);
    procedure
BtnPilihMakananClick(Sender:
TObject);
    procedure
BtnRefreshDetailClick(Sender:
TObject);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
TBTambahClick(Sender:
TObject);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
    procedure
BtnTambahDetailClick(Sender:
TObject);
    procedure
TBBatalClick(Sender: TObject);
    procedure
BtnUbahDetailClick(Sender:
TObject);
    procedure
BtnBatalDetailClick(Sender:
TObject);
```

```
    procedure
BtnHapusDetailClick(Sender:
TObject);
    procedure
BtnSimpanDetailClick(Sender:
TObject);
    procedure
DBGridDataDetailPermintaanCell
Click(Column: TColumn);
    procedure
DBControlDetailClick(Sender:
TObject;              Button:
TNavigateBtn);
    procedure
DBGridDataPermintaanCellClick(
Column: TColumn);
    procedure
DBControlMasterClick(Sender:
TObject;              Button:
TNavigateBtn);
    procedure
CBFieldCariMasterKeyPress(Send
er: TObject; var Key: Char);
    procedure
CBFieldCariDetailKeyPress(Send
er: TObject; var Key: Char);
    procedure
CBFieldCariMasterClick(Sender:
TObject);
    procedure
CBFieldCariDetailClick(Sender:
TObject);
    procedure
EdKataKunciMasterKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure
EdKataKunciDetailKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
    function GetIdFromEd(AEd :
TEdit) : String;
    procedure
RefreshDataMaster();
    procedure
RefreshDataDetail();
    procedure
SetAllKontrol(AModeData      :
TModeData);
    procedure
SetKontrolMaster(ABool       :
Boolean);
    procedure
SetKontrolDetail(ABool       :
Boolean);
    procedure
DisableKontrolMaster();
    procedure
DisableKontrolDetail();
    procedure
SetPropertyGridMaster();
    procedure
SetPropertyGridDetail();
    procedure
ViewDataMaster();
    procedure
ViewDataDetail();
    procedure
ClearAllFieldData();
    procedure
ClearFieldDataMaster();
    procedure
ClearFieldDataDetail();
    procedure
SetStatusBar(AModeData       :
TModeData);
    function
CheckMasterInput() : Boolean;
    function
CheckDetailInput() : Boolean;

  public
    { Public declarations }
  end;

var
  FDataPermintaanMakanan:
TFDataPermintaanMakanan;
  vModeData : TModeData;
  vQueryMaster : TADOQuery;
  vQueryDetail : TADOQuery;
  vInputMasterValid : Boolean;
```

```
  vInputDetailValid : Boolean;

implementation

uses  UDataIRI,  UDataMakanan,
UPermintaanMakanan,
ULoginUser;

{$R *.dfm}



{ TFDataPermintaanMakanan }

function
TFDataPermintaanMakanan.CheckD
etailInput(): Boolean;
begin
  If  (EdNoDokIRI.Text='')  or
(EdIdMakanan.Text ='') Then
    Result := False
  Else
    Result := True;
end;

function
TFDataPermintaanMakanan.CheckM
asterInput(): Boolean;
begin
  If   (EdIdPermintaan.Text   =
'') Then
      vInputMasterValid     :=
False
  Else
      vInputMasterValid     :=
True;
  Result := vInputMasterValid;
end;

function
TFDataPermintaanMakanan.GetIdF
romEd(AEd: TEdit): String;
var
  vPos : Integer;
begin
  If AEd.Text <> '' Then
    begin
      vPos := Pos('-',AEd.Text
);
      Result                :=
LeftStr(AEd.Text,vPos-1);
    end
  Else
    Result := '';
end;

procedure
TFDataPermintaanMakanan.Refres
hDataDetail;
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    vQueryDetail            :=
vData.GetDetailPermintaanMakan
an(EdIdPermintaan.Text);

DSDataDetailPermintaan.DataSet
:= vQueryDetail;

DBGridDataDetailPermintaan.Dat
aSource                     :=
DSDataDetailPermintaan;
    DBControlDetail.DataSource
:= DSDataDetailPermintaan;
    If
(vQueryDetail.RecordCount = 0)
Then
      begin
        ClearFieldDataDetail;
        DisableKontrolDetail;

BtnTambahDetail.Enabled     :=
True;
      end
    Else
      begin
        ClearFieldDataDetail;
        SetPropertyGridDetail;
        ViewDataDetail;

SetKontrolDetail(False);
      end;
```

```
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataPermintaanMakanan.Refres
hDataMaster;
var
  vData : TPermintaanMakanan;
begin
  try
    vData                    :=
TPermintaanMakanan.Create;
    vQueryMaster             :=
vData.GetMasterPermintaanMakan
an;
    DSDataPermintaan.DataSet
:= vQueryMaster;

DBGridDataPermintaan.DataSourc
e := DSDataPermintaan;
    DBControlMaster.DataSource
:= DSDataPermintaan;
    If
(vQueryMaster.RecordCount=0)
Then
      begin
        ClearAllFieldData;
        TBTambahClick(Nil);
        //vModeData          :=
InsertDataMaster;

//SetAllKontrol(vModeData);
        TBBatal.Enabled      :=
False;
        RefreshDataDetail;
        ShowMessage('Data
Permintaan Makanan Kosong !');
      end
    Else
      begin
        ClearFieldDataMaster;
        SetPropertyGridMaster;
        ViewDataMaster;
        vModeData            :=
BrowseDataMaster;
SetAllKontrol(vModeData);
        RefreshDataDetail;
      end;
    SetStatusBar(vModeData);
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataPermintaanMakanan.SetKon
trolDetail(ABool: Boolean);
begin
  //set kontrol untuk button
  BtnTambahDetail.Enabled   :=
Not ABool;
  BtnUbahDetail.Enabled := Not
ABool;
  BtnHapusDetail.Enabled    :=
Not ABool;
  BtnSimpanDetail.Enabled   :=
ABool;
  BtnBatalDetail.Enabled    :=
ABool;
  BtnRefreshDetail.Enabled  :=
Not ABool;
  //set   kontrol   untuk   db
control

DBGridDataDetailPermintaan.Ena
bled := Not ABool;
  DBControlDetail.Enabled   :=
Not ABool;
  PanelCariDetail.Enabled   :=
Not ABool;
  //set  kontrol  untuk  field
data
  EdKeterangan.Enabled      :=
ABool;
  BtnPilihIRI.Enabled       :=
ABool;
  BtnPilihMakanan.Enabled   :=
ABool;
end;
```

```
procedure
TFDataPermintaanMakanan.SetKon
trolMaster(ABool: Boolean);
begin
  //set kontrol untuk toolbar
  TBTambah.Enabled   :=   Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBRefresh.Enabled   :=   Not
ABool;

  //set   kontrol   untuk   db
control
  DBGridDataPermintaan.Enabled
:= Not ABool;
  DBControlMaster.Enabled   :=
Not ABool;
  PanelCariMaster.Enabled   :=
Not ABool;
  //set  kontrol  untuk  field
data
  DTTglPermintaan.Enabled   :=
ABool;
  //EdIdPermintaan.Enabled  :=
ABool;
end;

procedure
TFDataPermintaanMakanan.SetPro
pertyGridDetail;
var
  vCount : Integer;
begin
  With
DBGridDataDetailPermintaan Do
    begin
      For vCount := 0 To 7 Do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
        Columns[0].Font.Color
:= clWhite;
        Columns[0].Color    :=
clWindowFrame;

        Columns[0].Width := 0;
        Columns[1].Width := 0;
        Columns[2].Width    :=
75;
        Columns[3].Width    :=
75;
        Columns[4].Width    :=
120;
        Columns[5].Width := 0;
        Columns[6].Width    :=
90;
        Columns[7].Width    :=
120;
    end;
end;

procedure
TFDataPermintaanMakanan.SetPro
pertyGridMaster;
var
  vCount : Integer;
begin
  With DBGridDataPermintaan Do
    begin
      For vCount := 0 To 3 Do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
        Columns[0].Font.Color
:= clWhite;
        Columns[0].Color    :=
clWindowFrame;
        Columns[0].Width    :=
75;
        Columns[1].Width    :=
90;
```

```
        Columns[2].Width    :=
90;
        Columns[3].Width    :=
120;
    end;
end;

procedure
TFDataPermintaanMakanan.SetSta
tusBar(AModeData: TModeData);
begin

end;

procedure
TFDataPermintaanMakanan.ViewDa
taDetail;
begin
  EdIdDetailPermintaan.Text :=
vQueryDetail.Fields[0].Text;
  EdNoDokIRI.Text           :=
vQueryDetail.Fields[2].Text  +
'-(' +

vQueryDetail.Fields[3].Text  +
'-'                          +
vQueryDetail.Fields[4].Text  +
')';
  EdIdMakanan.Text          :=
vQueryDetail.Fields[5].Text  +
'-'                          +
vQueryDetail.Fields[6].Text;
  EdKeterangan.Text         :=
vQueryDetail.Fields[7].Text;
end;

procedure
TFDataPermintaanMakanan.ViewDa
taMaster;
begin
  If
(vQueryMaster.RecordCount=0)
Then
    DTTglPermintaan.Date    :=
Date
  Else
    DTTglPermintaan.Date    :=
vQueryMaster.Fields[1].Value;
  EdIdPermintaan.Text       :=
vQueryMaster.Fields[0].Text;
  EdIdPengurus.Text         :=
vQueryMaster.Fields[2].Text  +
'-'                          +
vQueryMaster.Fields[3].Text;

end;

procedure
TFDataPermintaanMakanan.TBKelu
arClick(Sender: TObject);
begin
  Close;
end;

procedure
TFDataPermintaanMakanan.ClearA
llFieldData;
begin
  ClearFieldDataMaster;
  ClearFieldDataDetail;
end;

procedure
TFDataPermintaanMakanan.ClearF
ieldDataMaster;
begin
  DTTglPermintaan.Date      :=
Date;
  EdIdPermintaan.Clear;
  EdIdPengurus.Clear;
  EdKataKunciMaster.Clear;
end;

procedure
TFDataPermintaanMakanan.ClearF
ieldDataDetail;
begin
  EdNoDokIRI.Clear;
  EdIdMakanan.Clear;
  EdKeterangan.Clear;
  EdKataKunciDetail.Clear;
end;

procedure
TFDataPermintaanMakanan.BtnPil
ihIRIClick(Sender: TObject);
```

```
begin
  FDataIRI.SetPilihIRI(True);
  FDataIRI.ShowModal;
end;

procedure
TFDataPermintaanMakanan.BtnPil
ihMakananClick(Sender:
TObject);
begin

FDataMakanan.SetPilihMakanan(F
DataPermintaanMakanan, True);
  FDataMakanan.ShowModal;
end;

procedure
TFDataPermintaanMakanan.SetAll
Kontrol(AModeData: TModeData);
begin
  If       (AModeData       =
InsertDataMaster) Then
    begin
      SetKontrolMaster(True);
      ClearFieldDataMaster;
      DisableKontrolDetail;
    end
  Else    If    (AModeData    =
UpdateDataMaster) Then
    begin
      SetKontrolMaster(True);
      EdIdPermintaan.Enabled
:= False;
      DisableKontrolDetail;
    end
  Else    If    (AModeData    =
InsertDataDetail) Then
    begin
      SetKontrolDetail(True);
      ClearFieldDataDetail;
      DisableKontrolMaster;
    end
  Else    If    (AModeData    =
UpdateDataDetail) Then
    begin
      SetKontrolDetail(True);
      DisableKontrolMaster;
      BtnPilihIRI.Enabled   :=
False;
      DisableKontrolMaster;
    end
  Else    If    (AModeData    =
SearchDataMaster) Then
    begin
      If
(vQueryMaster.RecordCount = 0)
Then
        begin
          TBUbah.Enabled    :=
False;
          TBHapus.Enabled   :=
False;
        end;
      TBTambah.Enabled      :=
False;
      RefreshDataDetail;
    end
  Else    If    (AModeData    =
SearchDataDetail) Then
    begin
      If
(vQueryDetail.RecordCount = 0)
Then
        begin

BtnUbahDetail.Enabled       :=
False;

BtnHapusDetail.Enabled      :=
False
        end;
      BtnTambahDetail.Enabled
:= False;
    end
  Else    If    (AModeData    =
BrowseDataMaster) Then
    begin
      SetKontrolMaster(False);
      SetKontrolDetail(False);
      ViewDataMaster;
    end;
end;
```

```
procedure
TFDataPermintaanMakanan.Disabl
eKontrolMaster;
begin
  DTTglPermintaan.Enabled   :=
False;
  EdIdPermintaan.Enabled    :=
False;
  TBTambah.Enabled := False;
  TBUbah.Enabled := False;
  TBHapus.Enabled := False;
  TBSimpan.Enabled := False;
  TBBAtal.Enabled := False;
  TBRefresh.Enabled := False;
  DBGridDataPermintaan.Enabled
:= False;
  DBControlMaster.Enabled   :=
False;
  PanelCariMaster.Enabled   :=
False;
end;

procedure
TFDataPermintaanMakanan.BtnRef
reshDetailClick(Sender:
TObject);
begin
  RefreshDataDetail;
end;

procedure
TFDataPermintaanMakanan.TBRefr
eshClick(Sender: TObject);
begin
  RefreshDataMaster;
end;

procedure
TFDataPermintaanMakanan.Disabl
eKontrolDetail;
begin
  BtnPilihIRI.Enabled       :=
False;
  BtnPilihMakanan.Enabled   :=
False;
  EdKeterangan.Enabled      :=
False;
  BtnTambahDetail.Enabled   :=
False;
  BtnUbahDetail.Enabled     :=
False;
  BtnSimpanDetail.Enabled   :=
False;
  BtnBatalDetail.Enabled    :=
False;
  BtnHapusDetail.Enabled    :=
False;

DBGridDataDetailPermintaan.Ena
bled := False;
  PanelCariDetail.Enabled   :=
False;
  DBControlDetail.Enabled   :=
False;
  BtnRefreshDetail.Enabled  :=
False;
end;

procedure
TFDataPermintaanMakanan.TBTamb
ahClick(Sender: TObject);
begin
  vModeData                 :=
InsertDataMaster;
  SetAllKontrol(vModeData);
  EdIdPermintaan.Text       :=
'Auto';
  EdIdPengurus.Text         :=
FLoginUser.vNIP   +   '-'   +
FLoginUser.vNamaPegawai;
  DTTglPermintaan.SetFocus;
end;

procedure
TFDataPermintaanMakanan.TBSimp
anClick(Sender: TObject);
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    If   not   CheckMasterInput
Then
      begin
```

```
        ShowMessage('Mohon Isi
ID.Permintaan Makanan!');
        FreeAndNil(vData);
        Exit;
      end;
    If      (vModeData      =
InsertDataMaster) Then
      begin
        If  MessageDlg('Apakah
Anda  Ingin  Menambahkan  Data
Ini ?',
          mtConfirmation,
[mbYes,mbNo],0) = mrYes Then
          begin
            If
vData.CreatePermintaanMakanan(

DTTglPermintaan.Date,

GetIdFromEd(EdIdPengurus))
Then

ShowMessage('Data   Permintaan
Makanan  Berhasil  Ditambahkan
!')
            Else

ShowMessage('Data   Permintaan
Makanan Gagal Ditambahkan !');
            RefreshDataMaster;
          end;
      end
    Else   If   (vModeData   =
UpdateDataMaster) Then
      begin
        If  MessageDlg('Apakah
Anda Ingin Mengubah Data Ini?'
,

mtConfirmation,[mbYes,mbNo],
0) = mrYes Then
          begin
            If
vData.UpdatePermintaanMakanan(

StrToInt(EdIdPermintaan.Text),

DTTglPermintaan.Date) Then

ShowMessage('Data   Permintaan
Makanan Berhasil Diubah !')
            Else

ShowMessage('Data   Permintaan
Makanan Gagal Diubah !');
            RefreshDataMaster;
          end;
      end;
  finally
    FreeAndNil(vData)
  end;
end;

procedure
TFDataPermintaanMakanan.TBUbah
Click(Sender: TObject);
begin
  vModeData                 :=
UpdateDataMaster;
  SetAllKontrol(vModeData);
  DTTglPermintaan.SetFocus;
end;

procedure
TFDataPermintaanMakanan.TBHapu
sClick(Sender: TObject);
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    If MessageDlg('Apakah Anda
Ingin      Menghapus      Data
Permintaaan     Dengan     ID.
Permintaan : ' +
        EdIdPermintaan.Text  +
'?'           ,mtConfirmation,
[mbYes,mbNo],0) = mrYes Then
      begin
        If
vData.DeletePermintaanMakanan(
StrToInt(EdIdPermintaan.Text))
Then
          begin
```

```
            ShowMessage('Data
Permintaan   Berhasil   Dihapus
!');
            RefreshDataMaster;
          end
        Else
          ShowMessage('Data
Permintan Gagal Dihapus !')
      end;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataPermintaanMakanan.BtnTam
bahDetailClick(Sender:
TObject);
begin
  vModeData                 :=
InsertDataDetail;
  SetAllKontrol(vModeData);
  BtnPilihIRI.SetFocus;
end;

procedure
TFDataPermintaanMakanan.TBBata
lClick(Sender: TObject);
begin
  vModeData                 :=
BrowseDataMaster;
  SetAllKontrol(vModeData);
end;

procedure
TFDataPermintaanMakanan.BtnUba
hDetailClick(Sender: TObject);
begin
  vModeData                 :=
UpdateDataDetail;
  SetAllKontrol(vModeData);
  BtnPilihMakanan.SetFocus;
end;

procedure
TFDataPermintaanMakanan.BtnBat
alDetailClick(Sender:
TObject);
begin
  vModeData                 :=
BrowseDataMaster;
  SetAllKontrol(vModeData);
  RefreshDataDetail;
end;

procedure
TFDataPermintaanMakanan.BtnHap
usDetailClick(Sender:
TObject);
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    If MessageDlg('Apakah Anda
Ingin  Menghapus  Data  Detail
Permintaan Ini ?' ,
      mtConfirmation,
[mbYes,mbNo],0) = mrYes Then
      begin
        If
vData.DeleteDetailPermintaanMa
kanan(
          EdIdPermintaan.Text,

StrToInt(EdIdDetailPermintaan.
Text)) Then
          begin
            ShowMessage('Data
Detail   Permintaan   Makanan
Berhasil Dihapus !');
            RefreshDataDetail;
          end
        Else
          ShowMessage('Data
Detail   Permintaan   Makanan
Gagal Dihapus !');
      end;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataPermintaanMakanan.BtnSim
```

```
panDetailClick(Sender:
TObject);
var
  vData : TPermintaanMakanan;
begin
  try
    vData                    :=
TPermintaanMakanan.Create;
    If   not   CheckDetailInput
Then
      begin
        ShowMessage('No.Dok
IRI   Dan   ID.Makanan   Harus
Diisi!');
        FreeAndNil(vData);
        Exit;
      end;
    If      (vModeData      =
InsertDataDetail) Then
      begin
        If  MessageDlg('Apakah
Anda Ingin Menambahkan Data '
+
          'Detail   Permintaan
Makanan Dengan No. Dok IRI : '
+ GetIdFromEd(EdNoDokIRI) +
          '       ?'       ,
mtConfirmation,
[mbYes,mbNo],0) = mrYes Then
          begin
            If
vData.CreateDetailPermintaanMa
kanan(

StrToInt(EdIdPermintaan.Text),

GetIdFromEd(EdNoDokIRI),

GetIdFromEd(EdIdMakanan),

EdKeterangan.Text) Then

ShowMessage('Data       Detail
Permintaan   Makanan   Berhasil
Ditambahkan !')
            Else

ShowMessage('Data       Detail
Permintaan    Makanan    Gagal
Ditambahkan !');
            RefreshDataDetail;
            vModeData       :=
BrowseDataMaster;

SetAllKontrol(vModeData);
          end;
      end
    Else    If(vModeData    =
UpdateDataDetail) Then
      begin
        If  MessageDlg('Apakah
Anda Ingin Mengubah Data ' +
          'Detail   Permintaan
Makanan Dengan No. Dok IRI : '
+ GetIdFromEd(EdNoDokIRI) +
          '       ?'       ,
mtConfirmation,
[mbYes,mbNo],0) = mrYes Then
          begin
            If
vData.UpdateDetailPermintaanMa
kanan(

StrToInt(EdIdDetailPermintaan.
Text),

GetIdFromEd(EdIdMakanan),

EdKeterangan.Text) Then

ShowMessage('Data       Detail
Permintaan   Makanan   Berhasil
Diubah !')
            Else

ShowMessage('Data       Detail
Permintaan    Makanan    Gagal
Diubah !');
            RefreshDataDetail;
            vModeData       :=
BrowseDataMaster;

SetAllKontrol(vModeData);
          end;
      end;
  finally
```

```
    FreeAndNil(vData);
  end;
end;

procedure
TFDataPermintaanMakanan.DBGrid
DataDetailPermintaanCellClick(
  Column: TColumn);
begin
  ViewDataDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataPermintaanMakanan.DBCont
rolDetailClick(Sender:
TObject;
  Button: TNavigateBtn);
begin
  ViewDataDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataPermintaanMakanan.DBGrid
DataPermintaanCellClick(
  Column: TColumn);
begin
  ViewDataMaster;
SetStatusBar(vModeData);
RefreshDataDetail;
end;

procedure
TFDataPermintaanMakanan.DBCont
rolMasterClick(Sender:
TObject;
  Button: TNavigateBtn);
begin
  ViewDataMaster;
SetStatusBar(vModeData);
RefreshDataDetail;
end;

procedure
TFDataPermintaanMakanan.CBFiel
dCariMasterKeyPress(
  Sender: TObject; var Key:
Char);
begin
  Key := #0;
end;

procedure
TFDataPermintaanMakanan.CBFiel
dCariDetailKeyPress(
  Sender: TObject; var Key:
Char);
begin
  Key := #0;
end;

procedure
TFDataPermintaanMakanan.CBFiel
dCariMasterClick(Sender:
TObject);
begin
  EdKataKunciMaster.Clear;
  EdKataKunciMaster.SetFocus;
end;

procedure
TFDataPermintaanMakanan.CBFiel
dCariDetailClick(Sender:
TObject);
begin
  EdKataKunciDetail.Clear;
  EdKataKunciDetail.SetFocus;
end;

procedure
TFDataPermintaanMakanan.EdKata
KunciMasterKeyUp(Sender:
TObject;
  var Key: Word; Shift:
TShiftState);
var
  vData : TPermintaanMakanan;
begin
  try
    vData                  :=
TPermintaanMakanan.Create;
    If
(Trim(CBFieldCariMaster.Text)
= '') Then
```

```
      Exit;
    vQueryMaster            :=
vData.GetDataSearchPermintaanM
akanan(CBFieldCariMaster.Text,
EdKataKunciMaster.Text);
    DSDataPermintaan.DataSet
:= vQueryMaster;

DBGridDataPermintaan.DataSourc
e := DSDataPermintaan;
    DBControlMaster.DataSource
:= DSDataPermintaan;
    vModeData               :=
SearchDataMaster;
    SetKontrolMaster(False);
    SetStatusBar(vModeData);
    SetPropertyGridMaster;
    ViewDataMaster;
    TBTambah.Enabled := False;
    If
(vQueryMaster.RecordCount = 0)
Then
      begin
        TBUbah.Enabled      :=
False;
        TBHapus.Enabled     :=
False;
      end
    Else
      begin
        TBUbah.Enabled      :=
True;
        TBHapus.Enabled     :=
True;
      end;
    RefreshDataDetail;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataPermintaanMakanan.EdKata
KunciDetailKeyUp(Sender:
TObject;
  var   Key:   Word;   Shift:
TShiftState);
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    If
(Trim(CBFieldCariDetail.Text)
= '') Then
      Exit;
    vQueryDetail            :=
vData.GetDataSearchDetailPermi
ntanMakanan(EdIdPermintaan.Tex
t,
CBFieldCariDetail.Text,EdKataK
unciDetail.Text);

DSDataDetailPermintaan.DataSet
:= vQueryDetail;

DBGridDataDetailPermintaan.Dat
aSource                     :=
DSDataDetailPermintaan;
    DBControlDetail.DataSource
:= DSDataDetailPermintaan;
    vModeData               :=
SearchDataMaster;
    SetKontrolMaster(False);
    SetKontrolDetail(False);
    SetStatusBar(vModeData);
    SetPropertyGridMaster;
    SetPropertyGridDetail;
    ViewDataMaster;
    ViewDataDetail;
    TBTambah.Enabled := False;
    BtnTambahDetail.Enabled :=
False;
    If
(vQueryDetail.RecordCount = 0)
Then
      begin
        BtnUbahDetail.Enabled
:= False;
        BtnHapusDetail.Enabled
:= False;
      end
    Else
      begin
```

```
        BtnUbahDetail.Enabled
:= True;
        BtnHapusDetail.Enabled
:= True;
      end;
  finally
    FreeAndNil(vData);
  end;
end;
procedure
TFDataPermintaanMakanan.FormSh
ow(Sender: TObject);
begin
  RefreshDataMaster;
end;

end.

unit UDataPegawai;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs,  StdCtrls,  DBCtrls,
Grids,  DBGrids,  DB,  ImgList,
XPMan, ExtCtrls,
  ComCtrls,   ToolWin,   ADODB,
StrUtils;

type
  TModeData   =   (BrowseData,
UpdateData,        InsertData,
DeleteData, SearchData);
type
  TFDataPegawai = class(TForm)
    ToolBar: TToolBar;
    TBTambah: TToolButton;
    TBUbah: TToolButton;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
    TBRefresh: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    PanelDetail: TPanel;
    LblNIP: TLabel;
    LblNamaPegawai: TLabel;
    LblTanggalLahir: TLabel;
    LblAlamatPegawai: TLabel;
    EdNIP: TEdit;
    EdNamaPegawai: TEdit;
    XPStyle: TXPManifest;
    ImageList: TImageList;
    DSDataPegawai:
TDataSource;
    PanelDBGrid: TPanel;
    DBGridDataPegawai:
TDBGrid;
    DBControl: TDBNavigator;
    PanelCari: TPanel;
    LblFieldCari: TLabel;
    LblKataKunci: TLabel;
    CBFieldCari: TComboBox;
    EdKataKunci: TEdit;
    EdAlamatPegawai: TMemo;
    DTTglLahir:
TDateTimePicker;
    TBDaftar: TToolButton;
    StatusBar: TStatusBar;
    LblStatus: TLabel;
    EdStatus: TEdit;
    procedure
TBKeluarClick(Sender:
TObject);
    procedure
TBTambahClick(Sender:
TObject);
    procedure
DBGridDataPegawaiCellClick(Col
umn: TColumn);
    procedure
DBControlClick(Sender:
TObject;              Button:
TNavigateBtn);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
    procedure
TBBatalClick(Sender: TObject);
```

```
    procedure
CBFieldCariClick(Sender:
TObject);
    procedure
CBFieldCariKeyPress(Sender:
TObject; var Key: Char);
    procedure
EdKataKunciKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
TBDaftarClick(Sender:
TObject);
    procedure RefreshData();
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
    procedure
SetPropertyGridPegawai();
    procedure
ClearFieldData();
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewDetail();
    procedure
SetStatusBar(AModeData       :
TModeData);
    function
CekValiditasInput() : Boolean;
  public
    { Public declarations }
  end;

var
  FDataPegawai: TFDataPegawai;
  vQuery : TADOQuery;
  vModeData : TModeData;

implementation

uses UPegawai, UDaftarUser;

{$R *.dfm}

function
TFDataPegawai.CekValiditasInpu
t: Boolean;
begin
  If  (Trim(EdNIP.Text)  =  '')
or   (Trim(EdNamaPegawai.Text)
='') Then
    Result := False
  Else
    Result := True;
end;

procedure
TFDataPegawai.ClearFieldData;
begin
  EdNIP.Clear;
  EdNamaPegawai.Clear;
  DTTglLahir.Date := Date;
  EdAlamatPegawai.Clear;
  EDKataKunci.Clear;
end;

procedure
TFDataPegawai.RefreshData;
var
  vData : TPegawai;
begin
  try
    vData := TPegawai.Create;
    vQuery                  :=
vData.GetPegawai;
    DSDataPegawai.DataSet   :=
vQuery;

DBGridDataPegawai.DataSource
:= DSDataPegawai;
    DBControl.DataSource    :=
DSDataPegawai;
    If  (vQuery.RecordCount  =
0) Then
      begin
        ShowMessage('Data
Pegawai  Kosong,  Anda  Harus
Melakukan Penambahan Data !');
        ClearFieldData;
```

```
        SetKontrol(True);
        TBBatal.Enabled     :=
False;
        TBTambahClick(Nil);
      end
    else
      begin
        SetKontrol(False);
        vModeData           :=
BrowseData;
        ClearFieldData;
        ViewDetail;
      end;

//============================
============================
=====
    SetStatusBar(vModeData);
    SetPropertyGridPegawai;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataPegawai.SetKontrol(ABool
: Boolean);
begin
  PanelDetail.Enabled       :=
ABool;
  PanelDBGrid.Enabled  :=  Not
ABool;
  PanelCari.Enabled   :=   Not
ABool;
  TBTambah.Enabled    :=    Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBDaftar.Enabled    :=    Not
ABool;
  TBRefresh.Enabled   :=   Not
ABool;
end;

procedure
TFDataPegawai.SetPropertyGridP
egawai;
var
  vCount : Integer;
begin
  With DBGridDataPegawai do
    begin
      for vCount :=0 to 4 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
      Columns[0].Width := 100;
      Columns[1].Width := 150;
      Columns[2].Width := 100;
      Columns[3].Width := 150;
      Columns[4].Width := 80;
      Columns[0].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;

procedure
TFDataPegawai.SetStatusBar(AMo
deData: TModeData);
begin
  If (AModeData = InsertData)
Then
    StatusBar.Panels[0].Text
:=  'Insert Mode - Penambahan
Data'
  Else    If    (AModeData    =
UpdateData) Then
    StatusBar.Panels[0].Text
:= 'Update Mode - Pengubahan
Data'
  Else    If    (AModeData    =
BrowseData) Then
```

```
    StatusBar.Panels[0].Text
:= 'Browse Mode - Penelusuran
Data'
  Else    If    (AModeData    =
DeleteData) Then
    StatusBar.Panels[0].Text
:= 'Delete Mode - Penghapusan
Data'
  Else    If    (AModeData    =
SearchData) Then
    StatusBar.Panels[0].Text
:= 'Search Mode - Pencarian
Data'
  Else
    StatusBar.Panels[0].Text
:= 'Unknown Mode';
  StatusBar.Panels[1].Text  :=
'Data    Pegawai    Ke-'    +
IntToStr(vQuery.RecNo)   +   '
Dari            '            +
IntToStr(vQuery.RecordCount) +
' Data';
end;

procedure
TFDataPegawai.TBKeluarClick(Se
nder: TObject);
begin
  Close;
end;

procedure
TFDataPegawai.TBTambahClick(Se
nder: TObject);
begin
  ClearFieldData;
  vModeData := InsertData;
  SetKontrol(True);
  EdStatus.Text    :=    'Tidak
Terdaftar';
  EdNIP.Enabled := True;
  EdNIP.SetFocus;
  SetStatusBar(vModeData);
end;

procedure
TFDataPegawai.ViewDetail;
begin
  EdNIP.Text                :=
DBGridDataPegawai.Fields[0].Te
xt;
  EdNamaPegawai.Text        :=
DBGridDataPegawai.Fields[1].Te
xt;
  If (vQuery.RecordCount = 0)
Then
    DTTglLahir.Date := Date
  Else
    DTTglLahir.Date         :=
DBGridDataPegawai.Fields[2].Va
lue;
  EdAlamatPegawai.Text      :=
DBGridDataPegawai.Fields[3].Te
xt;
  EdStatus.Text             :=
DBGridDataPegawai.Fields[4].Te
xt;
  If     (EdStatus.Text     =
'Terdaftar') Then
    begin
      TBDaftar.Visible      :=
False;
      TBHapus.Enabled       :=
False;
    end
  Else
    begin
      TBDaftar.Visible      :=
True;
      TBHapus.Enabled := True;
    end;

end;

procedure
TFDataPegawai.DBGridDataPegawa
iCellClick(Column: TColumn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataPegawai.DBControlClick(S
ender: TObject;
  Button: TNavigateBtn);
```

```
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataPegawai.TBUbahClick(Send
er: TObject);
begin
  If (vModeData = BrowseData)
Then
    vModeData := UpdateData;
  SetKontrol(True);
  EdNIP.Enabled := False;
  EdNamaPegawai.SetFocus;
  SetStatusBar(vModeData);
end;

procedure
TFDataPegawai.TBHapusClick(Sen
der: TObject);
var
  vData : TPegawai;
begin
  vModeData := DeleteData;
  SetStatusBar(vModeData);
  If  MessageDlg('Anda  Yakin
Menghapus Data Pegawai Dengan
NIP : ' +
      EdNIP.Text  +  '  ?'  ,
mtConfirmation,        [mbYes,
mbNo],0) = mrYes Then
    begin
      try
        vData               :=
TPegawai.Create;
        If
vData.DeletePegawai(EdNIP.Text
) Then
          ShowMessage('Data
Pegawai Berhasil Dihapus !')
        Else
          ShowMessage('Data
Pegawai Gagal Dihapus !');
        RefreshData;
      finally
        FreeAndNil(vData);
      end;
    end
  else
    begin
      vModeData := BrowseData;
      SetStatusBar(vModeData);
    end;
end;


procedure
TFDataPegawai.TBBatalClick(Sen
der: TObject);
begin
  If (vModeData = InsertData)
or (vModeData =   UpdateData)
Then
    begin
      vModeData := BrowseData;
      ViewDetail;
      SetKontrol(False);
      SetStatusBar(vModeData);
    end
  Else     If(vModeData     =
SearchData) Then
    begin
      ViewDetail;
      SetKontrol(False);
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := True;
      TBHapus.Enabled := True;
    end;
end;

procedure
TFDataPegawai.CBFieldCariClick
(Sender: TObject);
begin
  EdKataKunci.Clear;
  EdKataKunci.SetFocus;
end;

procedure
TFDataPegawai.CBFieldCariKeyPr
ess(Sender: TObject;
  var Key: Char);
begin
  Key := #0;
```

```
end;

procedure
TFDataPegawai.EdKataKunciKeyUp
(Sender: TObject; var Key:
Word;
  Shift: TShiftState);
var
  vData : TPegawai;
begin
  try
    vData := TPegawai.Create;
    If (Trim(CBFieldCari.Text)
= '') Then
        Exit;
    vQuery                  :=
vData.GetDataSearchPegawai(CBF
ieldCari.Text,
EdKataKunci.Text);
    DSDataPegawai.DataSet   :=
vQuery;

DBGridDataPegawai.DataSource
:= DSDataPegawai;
    DBControl.DataSource    :=
DSDataPegawai;
    vModeData := SearchData;
    SetKontrol(False);
    ViewDetail;
    SetStatusBar(vModeData);
    SetPropertyGridPegawai;
    TBTambah.Enabled := False;
    If  (vQuery.RecordCount  =
0) Then
      begin
        TBUbah.Enabled      :=
False;
        TBHapus.Enabled     :=
False;
      end
    Else
      begin
        TBUbah.Enabled      :=
True;
        TBHapus.Enabled     :=
True;
      end;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataPegawai.TBSimpanClick(Se
nder: TObject);
var
  vData : TPegawai;
begin
  If   not   CekValiditasInput
Then
    begin
      ShowMessage('Terjadi
Kesalahan  Dalam  Penginputan
Data, Mohon Diperbaiki');
      Exit;
    end;
  If (vModeData = InsertData)
Then
    begin
      If(MessageDlg('Apakah
Anda  Yakin  Ingin  Menambahkan
Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData           :=
TPegawai.Create;
            If
(vData.CreatePegawai(
                EdNIP.Text,

EdNamaPegawai.Text,

DTTglLahir.Date,

EdAlamatPegawai.Text,
                'Tidak
Terdaftar')) Then

ShowMessage('Data      Pegawai
Berhasil Ditambahkan !')
            Else

ShowMessage('Data      Pegawai
Ditambahkan !');
```

```
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end
  Else   If   (vModeData   =
UpdateData)  or  (vModeData  =
SearchData) Then
    begin
      If(MessageDlg('Apakah
Anda Yakin Ingin Mengubah Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData          :=
TPegawai.Create;
            If
(vData.UpdatePegawai(
                EdNIP.Text,

EdNamaPegawai.Text,

DTTglLahir.Date,

EdAlamatPegawai.Text)) Then

ShowMessage('Data      Pegawai
Berhasil Diubah !')
            Else

ShowMessage('Data      Pegawai
Gagal Diubah !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end;
  RefreshData;
end;

procedure
TFDataPegawai.TBRefreshClick(S
ender: TObject);
begin
  RefreshData;
end;

procedure
TFDataPegawai.TBDaftarClick(Se
nder: TObject);
begin
  FDaftarUser.ShowModal;
end;

procedure
TFDataPegawai.FormShow(Sender:
TObject);
begin
  RefreshData();
end;

end.

unit UDataPasien;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Controls,
Forms,
  Dialogs, StdCtrls, ComCtrls,
ImgList,   DBGrids,   ToolWin,
Grids, ExtCtrls,
  XPMan,  DBCtrls,  DB,  ADODB,
Buttons, StrUtils, Graphics;

type
  TModeData   =   (BrowseData,
UpdateData,        InsertData,
DeleteData, SearchData);
type
  TFDataPasien = class(TForm)
    StatusBar: TStatusBar;
    ImageList: TImageList;
    ToolBar: TToolBar;
    TBTambah: TToolButton;
    TBUbah: TToolButton;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
```

```
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
    TBRefresh: TToolButton;
    TBCetak: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    PanelDBGrid: TPanel;
    DBGridDataPasien: TDBGrid;
    PanelDetail: TPanel;
    EdNoMedRec: TEdit;
    EdNamaPasien: TEdit;
    CBJenisKelamin: TComboBox;
    CBAgama: TComboBox;
    EdPekerjaan: TEdit;
    EdAlamat: TMemo;
    XPStyle: TXPManifest;
    LblNoMedRec: TLabel;
    LblNamaPasien: TLabel;
    LblTglLahir: TLabel;
    LblJenisKelamin: TLabel;
    LblAgama: TLabel;
    LblPendidikan: TLabel;
    LblPekerjaan: TLabel;
    LblAlamat: TLabel;
    DBControl: TDBNavigator;
    PanelCari: TPanel;
    LblFieldCari: TLabel;
    LblKataKunci: TLabel;
    CBFieldCari: TComboBox;
    EdKataKunci: TEdit;
    DSDataPasien: TDataSource;
    CBPendidikan: TComboBox;
    DTTglLahir:
TDateTimePicker;
    TBPilihPasien:
TToolButton;
    procedure
TBKeluarClick(Sender:
TObject);
    procedure
DBControlClick(Sender:
TObject;              Button:
TNavigateBtn);
    procedure
DBGridDataPasienCellClick(Colu
mn: TColumn);
    procedure
TBTambahClick(Sender:
TObject);
    procedure
TBBatalClick(Sender: TObject);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
EdKataKunciKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure
CBFieldCariKeyPress(Sender:
TObject; var Key: Char);
    procedure
CBFieldCariClick(Sender:
TObject);
    procedure
TBPilihPasienClick(Sender:
TObject);
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
    procedure
SetPropertyGridPasien();
    procedure
ClearFieldData();
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewDetail();
    procedure RefreshData();
    procedure
SetStatusBar(AModeData       :
TModeData);
    function
CekValiditasInput() : Boolean;
  public
    { Public declarations }
```

```
    procedure
SetPilihPasien(APilihPasien  :
Boolean);
  end;

var
  FDataPasien: TFDataPasien;
  vQuery : TADOQuery;
  vModeData : TModeData;
  vPilihPasien : Boolean;

implementation

uses  UIRI,  USys,  UPegawai,
UDataIRI;

{$R *.dfm}

procedure
TFDataPasien.TBKeluarClick(Sen
der: TObject);
begin
  vPilihPasien := False;
  Close;
end;

procedure
TFDataPasien.SetKontrol(ABool
: Boolean);
begin
  PanelDetail.Enabled       :=
ABool;
  PanelDBGrid.Enabled  :=  Not
ABool;
  PanelCari.Enabled   :=   Not
ABool;
  TBTambah.Enabled    :=    Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBRefresh.Enabled   :=   Not
ABool;
  TBCetak.Enabled    :=    Not
ABool;
  TBPilihPasien.Enabled := Not
ABool;
  If  (vPilihPasien  =  True)
Then
    TBPilihPasien.Visible   :=
True
  Else
    TBPilihPasien.Visible   :=
False;
end;

procedure
TFDataPasien.SetPropertyGridPa
sien;
var
  vCount : Integer;
begin
  With DBGridDataPasien do
    begin
      for vCount :=0 to 7 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
      Columns[0].Width := 90;
      Columns[0].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;

procedure
TFDataPasien.ViewDetail;
begin
  EdNoMedRec.Text           :=
DBGridDataPasien.Fields[0].Tex
t;
  EdNamaPasien.Text         :=
DBGridDataPasien.Fields[1].Tex
t;
```

```
  If (vQuery.RecordCount = 0)
Then
    DTTglLahir.Date := Date
  Else
    DTTglLahir.Date         :=
DBGridDataPasien.Fields[2].Val
ue;
  CBJenisKelamin.Text       :=
DBGridDataPasien.Fields[3].Tex
t;
  EdAlamat.Text             :=
DBGridDataPasien.Fields[4].Tex
t;
  CBPendidikan.Text         :=
DBGridDataPasien.Fields[5].Tex
t;
  EdPekerjaan.Text          :=
DBGridDataPasien.Fields[6].Tex
t;
  CBAgama.Text              :=
DBGridDataPasien.Fields[7].Tex
t;
end;

procedure
TFDataPasien.DBControlClick(Se
nder: TObject;
  Button: TNavigateBtn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataPasien.DBGridDataPasienC
ellClick(Column: TColumn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataPasien.ClearFieldData;
begin
  EdNoMedRec.Clear;
  EdNamaPasien.Clear;
  DTTglLahir.Date := Date;
  CBJenisKelamin.Text := '';
  cbAgama.Text := '';
  CBPendidikan.Text := '';
  EdPekerjaan.Clear;
  EdAlamat.Clear;
  EdKataKunci.Clear;
end;

procedure
TFDataPasien.SetStatusBar(AMod
eData : TModeData);
begin
  If (AModeData = InsertData)
Then
    StatusBar.Panels[0].Text
:=  'Insert Mode - Penambahan
Data'
  Else   If   (AModeData   =
UpdateData) Then
    StatusBar.Panels[0].Text
:= 'Update Mode - Pengubahan
Data'
  Else   If   (AModeData   =
BrowseData) Then
    StatusBar.Panels[0].Text
:= 'Browse Mode - Penelusuran
Data'
  Else   If   (AModeData   =
DeleteData) Then
    StatusBar.Panels[0].Text
:= 'Delete Mode - Penghapusan
Data'
  Else   If   (AModeData   =
SearchData) Then
    StatusBar.Panels[0].Text
:= 'Search Mode - Pencarian
Data'
  Else
    StatusBar.Panels[0].Text
:= 'Unknown Mode';
  StatusBar.Panels[1].Text  :=
'Status Data Pasien : Data Ke-
' + IntToStr(vQuery.RecNo) + '
Dari           '           +
IntToStr(vQuery.RecordCount) +
' Data';
  StatusBar.Panels[2].Text  :=
'IRI-RS. HasanSadikin';
end;
```

```
procedure
TFDataPasien.TBTambahClick(Sen
der: TObject);
begin
  ClearFieldData;
  vModeData := InsertData;
  SetKontrol(True);
  EdNoMedRec.Enabled := True;
  EdNoMedRec.SetFocus;
  SetStatusBar(vModeData);
end;

procedure
TFDataPasien.TBBatalClick(Send
er: TObject);
begin
  If (vModeData = InsertData)
or (vModeData =   UpdateData)
Then
    begin
      vModeData := BrowseData;
      ViewDetail;
      SetKontrol(False);
      SetStatusBar(vModeData);
    end
  Else     If(vModeData     =
SearchData) Then
    begin
      ViewDetail;
      SetKontrol(False);
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := True;
      TBHapus.Enabled := True;
    end;
end;

procedure
TFDataPasien.TBRefreshClick(Se
nder: TObject);
begin
  RefreshData;
end;

procedure
TFDataPasien.TBUbahClick(Sende
r: TObject);
begin
  If (vModeData = BrowseData)
Then
    vModeData := UpdateData;
  SetKontrol(True);
  EdNoMedRec.Enabled := False;
  SetStatusBar(vModeData);
end;

procedure
TFDataPasien.TBHapusClick(Send
er: TObject);
var
  vData : TIRI;
begin
  vModeData := DeleteData;
  SetStatusBar(vModeData);
  If  MessageDlg('Anda  Yakin
Menghapus  Data  Pasien  Dengan
No. Medical Rec. : ' +
      EdNoMedRec.Text + ' ?' ,
mtConfirmation,        [mbYes,
mbNo],0) = mrYes Then
    begin
      try
        vData := TIRI.Create;
        If
vData.DeletePasien(EdNoMedRec.
Text) Then
          ShowMessage('Data
Pasien Berhasil Dihapus !')
        Else
          ShowMessage('Data
Pasien Gagal Dihapus !');
        RefreshData;
      finally
        FreeAndNil(vData);
      end;
    end
  else
    begin
      vModeData := BrowseData;
      SetStatusBar(vModeData);
    end;
end;

procedure
TFDataPasien.RefreshData();
```

```
var
  vData : TIRI;
begin
  try
    vData := TIRI.Create;
    vQuery := vData.GetPasien;
    DSDataPasien.DataSet    :=
vQuery;

DBGridDataPasien.DataSource :=
DSDataPasien;
    DBControl.DataSource    :=
DSDataPasien;
    //FIRST LOAD - CHECK THE
STATUS DATA IF EMPTY OR ANY
DATA IS IN

//============================
============================
=====
    If  (vQuery.RecordCount  =
0) Then
      begin
        ShowMessage('Data
Pasien  Kosong,  Anda  Harus
Melakukan Penambahan Data !');
        ClearFieldData;
        SetKontrol(True);
        TBTambahClick(Nil);
        TBBatal.Enabled     :=
False;
      end
    else
      begin
        SetKontrol(False);
        CBFieldCari.ItemIndex
:= 0;
        vModeData           :=
BrowseData;
        ClearFieldData;
        ViewDetail;
      end;

//============================
============================
=====
    SetStatusBar(vModeData);
    SetPropertyGridPasien;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataPasien.TBSimpanClick(Sen
der: TObject);
var
  vData : TIRI;
begin
  If   not   CekValiditasInput
Then
    begin
      ShowMessage('Terjadi
Kesalahan  Dalam  Penginputan
Data, Mohon Diperbaiki');
      Exit;
    end;
  If (vModeData = InsertData)
Then
    begin
      If(MessageDlg('Apakah
Anda  Yakin  Ingin  Menambahkan
Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData           :=
TIRI.Create;
            If
(vData.CreatePasien(

EdNoMedRec.Text,

EdNamaPasien.Text,

DTTglLahir.Date,

CBJenisKelamin.Text,
                EdAlamat.Text,

CBPendidikan.Text,

EdPekerjaan.Text,
                CBAgama.Text))
Then
```

```
ShowMessage('Data       Pasien
Berhasil Ditambahkan !')
            Else

ShowMessage('Data Pasien Gagal
Ditambahkan !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end
  Else    If    (vModeData    =
UpdateData)  or  (vModeData  =
SearchData) Then
    begin
      If(MessageDlg('Apakah
Anda Yakin Ingin Mengubah Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData           :=
TIRI.Create;
            If
(vData.UpdatePasien(

EdNoMedRec.Text,

EdNamaPasien.Text,

DTTglLahir.Date,

CBJenisKelamin.Text,
                EdAlamat.Text,

CBPendidikan.Text,

EdPekerjaan.Text,
                CBAgama.Text))
Then

ShowMessage('Data       Pasien
Berhasil Diubah !')
            Else

ShowMessage('Data Pasien Gagal
Diubah !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end;
  RefreshData;
end;


procedure
TFDataPasien.EdKataKunciKeyUp(
Sender:   TObject;   var   Key:
Word;
  Shift: TShiftState);
var
  vData : TIRI;
begin
  try
    vData := TIRI.Create;
    If (Trim(CBFieldCari.Text)
= '') Then
      Exit;
    vQuery                  :=
vData.GetDataSearchPasien(CBFi
eldCari.Text,
EdKataKunci.Text);
    DSDataPasien.DataSet    :=
vQuery;

DBGridDataPasien.DataSource :=
DSDataPasien;
    DBControl.DataSource    :=
DSDataPasien;
    vModeData := SearchData;
    SetKontrol(False);
    ViewDetail;
    SetStatusBar(vModeData);
    SetPropertyGridPasien;
    TBTambah.Enabled := False;
    If  (vQuery.RecordCount  =
0) Then
      begin
```

```
        TBUbah.Enabled      :=
False;
        TBHapus.Enabled     :=
False;
      end
    Else
      begin
        TBUbah.Enabled      :=
True;
        TBHapus.Enabled     :=
True;
      end;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataPasien.CBFieldCariKeyPre
ss(Sender: TObject; var Key:
Char);
begin
  Key := #0;
end;

procedure
TFDataPasien.CBFieldCariClick(
Sender: TObject);
begin
  EdKataKunci.Text := '';
  EdKataKunci.SetFocus;
end;

procedure
TFDataPasien.SetPilihPasien(AP
ilihPasien: Boolean);
begin
  vPilihPasien              :=
APilihPasien;
end;

procedure
TFDataPasien.TBPilihPasienClic
k(Sender: TObject);
begin
  vPilihPasien := False;
  FDataIRI.EdNoMedRec.Text  :=
EdNoMedRec.Text   +   '-'   +
EdNamaPasien.Text;
  Close;
end;

procedure
TFDataPasien.FormShow(Sender:
TObject);
begin
  RefreshData;
end;

function
TFDataPasien.CekValiditasInput
: Boolean;
begin
  If(Trim(EdNoMedRec.Text)   =
'')                         or
(Trim(EdNamaPasien.Text) = '')
Then
    Result := False
  Else
    Result := True;
end;
end.

unit UDataMakanan;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs, DB, ImgList, XPMan,
ComCtrls,  StdCtrls,  DBCtrls,
Grids, DBGrids,
  ExtCtrls,   ToolWin,   ADODB,
StrUtils;

type
  TModeData   =   (BrowseData,
UpdateData,        InsertData,
DeleteData, SearchData);
type
  TFDataMakanan = class(TForm)
    ToolBar: TToolBar;
```

```
    TBTambah: TToolButton;
    TBUbah: TToolButton;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
    TBRefresh: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    PanelDetail: TPanel;
    LblIdMakanan: TLabel;
    LblNamaMakanan: TLabel;
    EdIdMakanan: TEdit;
    EdNamaMakanan: TEdit;
    PanelDBGrid: TPanel;
    DBGridDataMakanan:
TDBGrid;
    DBControl: TDBNavigator;
    PanelCari: TPanel;
    LblFieldCari: TLabel;
    LblKataKunci: TLabel;
    CBFieldCari: TComboBox;
    EdKataKunci: TEdit;
    StatusBar: TStatusBar;
    XPStyle: TXPManifest;
    ImageList: TImageList;
    DSDataMakanan:
TDataSource;
    LblIdJenisMakanan: TLabel;
    CBIdJenisMakanan:
TComboBox;
    TBPilih: TToolButton;
    procedure
DBGridDataMakananCellClick(Col
umn: TColumn);
    procedure
DBControlClick(Sender:
TObject;              Button:
TNavigateBtn);
    procedure
CBFieldCariClick(Sender:
TObject);
    procedure
CBFieldCariKeyPress(Sender:
TObject; var Key: Char);
    procedure
EdKataKunciKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure
TBTambahClick(Sender:
TObject);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
TBBatalClick(Sender: TObject);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
TBKeluarClick(Sender:
TObject);

    procedure
TBPilihClick(Sender: TObject);
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
    procedure
SetPropertyGridMakanan();
    procedure
ClearFieldData();
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewDetail();
    procedure RefreshData();
    procedure
SetStatusBar(AModeData       :
TModeData);
    procedure
LoadDataJenisMakananToCombo();
    function
GetIdJenisMakananFromCombo(ACo
mbo : TComboBox) : String;
    function
CekValiditasInput() : Boolean;
  public
```

```
    { Public declarations }
    procedure
SetPilihMakanan(AFormSender  :
TForm;     APilihMakanan     :
Boolean);
  end;

var
  FDataMakanan: TFDataMakanan;
  vFormSender : TForm;
  vQuery : TADOQuery;
  vModeData : TModeData;
  vPilihMakanan : Boolean;

implementation

uses       UPermintaanMakanan,
UDataDetaiBahanMakanan,
UDataPermintaanMakanan,
  ULoginUser;

{$R *.dfm}

{ TFDataMakanan }

procedure
TFDataMakanan.ClearFieldData;
begin
  EdIdMakanan.Clear;
  EdNamaMakanan.Clear;
  CBIdJenisMakanan.Text := '';
  EdKataKunci.Clear;
end;

procedure
TFDataMakanan.DBGridDataMakana
nCellClick(Column: TColumn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

function
TFDataMakanan.GetIdJenisMakana
nFromCombo(
  ACombo: TComboBox): String;
var
  vPos : integer;
begin
  if ACombo.Text <> '' then
    begin
      vPos      :=      Pos('-
',ACombo.Text);
      Result                :=
LeftStr(ACombo.Text,vPos-1);
    end
  else
    Result := '';
end;

procedure
TFDataMakanan.LoadDataJenisMak
ananToCombo;
var
  vData : TPermintaanMakanan;
  vQueryTemp : TADOQuery;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    vQueryTemp              :=
vData.GetJenisMakanan;
    CBIdJenisMakanan.Clear;
    While  Not  vQueryTemp.Eof
Do
      begin

CBIdJenisMakanan.Items.Add(vQu
eryTemp.Fields[0].Text + '-' +
vQueryTemp.Fields[1].Text);
        vQueryTemp.Next;
      end;
  finally
    FreeAndNil(vQueryTemp);
    FreeAndNil(vData);
  end;
end;

procedure
TFDataMakanan.RefreshData;
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
```

```
    vQuery                  :=
vData.GetMakanan;
    DSDataMakanan.DataSet   :=
vQuery;

DBGridDataMakanan.DataSource
:= DSDataMakanan;
    DBControl.DataSource    :=
DSDataMakanan;
    //FIRST LOAD - CHECK THE
STATUS DATA IF EMPTY OR ANY
DATA IS IN

//==========================
============================
=====
    If  (vQuery.RecordCount  =
0) Then
      begin
        ShowMessage('Data
Makanan  Kosong,  Anda  Harus
Melakukan Penambahan Data !');
        ClearFieldData;
        SetKontrol(True);
        TBTambahClick(Nil);
        TBBatal.Enabled     :=
False;
      end
    else
      begin
        SetKontrol(False);
        vModeData           :=
BrowseData;
        ClearFieldData;
        ViewDetail;
      end;

//==========================
============================
=====
    SetStatusBar(vModeData);
    SetPropertyGridMakanan;
  finally
    FreeAndNil(vData);
  end;
end;
```

```
procedure
TFDataMakanan.SetKontrol(ABool
: Boolean);
begin
  PanelDetail.Enabled       :=
ABool;
  PanelDBGrid.Enabled  :=  Not
ABool;
  PanelCari.Enabled   :=   Not
ABool;
  TBTambah.Enabled    :=    Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBRefresh.Enabled   :=   Not
ABool;
  TBPilih.Enabled    :=    Not
ABool;
  If  (vPilihMakanan  =  True)
Then
    TBPilih.Visible := True
  Else
    TBPilih.Visible := False;
  If   FLoginUser.vHakAkses   =
'Staff' Then
    begin
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := False;
      TBHapus.Enabled       :=
False;
      TBSimpan.Enabled      :=
False;
      TBBatal.Enabled       :=
False;
    end;
end;

procedure
TFDataMakanan.SetPropertyGridM
akanan;
var
  vCount : Integer;
begin
  With DBGridDataMakanan do
```

```
    begin
      for vCount :=0 to 3 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
      Columns[0].Width := 70;
      Columns[1].Width := 200;
      Columns[2].Width := 60;
      Columns[3].Width := 120;
      Columns[0].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;

procedure
TFDataMakanan.SetStatusBar(AMo
deData: TModeData);
begin
  If (AModeData = InsertData)
Then
    StatusBar.Panels[0].Text
:=  'Insert Mode - Penambahan
Data'
  Else    If    (AModeData    =
UpdateData) Then
    StatusBar.Panels[0].Text
:= 'Update Mode - Pengubahan
Data'
  Else    If    (AModeData    =
BrowseData) Then
    StatusBar.Panels[0].Text
:= 'Browse Mode - Penelusuran
Data'
  Else    If    (AModeData    =
DeleteData) Then
    StatusBar.Panels[0].Text
:= 'Delete Mode - Penghapusan
Data'
  Else    If    (AModeData    =
SearchData) Then
    StatusBar.Panels[0].Text
:= 'Search Mode - Pencarian
Data'
  Else
    StatusBar.Panels[0].Text
:= 'Unknown Mode';
  StatusBar.Panels[1].Text  :=
'Data    Makanan    Ke-'    +
IntToStr(vQuery.RecNo)   +   '
Dari            '            +
IntToStr(vQuery.RecordCount) +
' Data';
end;

procedure
TFDataMakanan.ViewDetail;
begin
  EdIdMakanan.Text          :=
DBGridDataMakanan.Fields[0].Te
xt;
  EdNamaMakanan.Text        :=
DBGridDataMakanan.Fields[1].Te
xt;
  CBIdJenisMakanan.Text     :=
DBGridDataMakanan.Fields[2].Te
xt       +       '-'       +
DBGridDataMakanan.Fields[3].Te
xt
end;

procedure
TFDataMakanan.DBControlClick(S
ender: TObject;
  Button: TNavigateBtn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataMakanan.CBFieldCariClick
(Sender: TObject);
begin
  EdKataKunci.Clear;
  EdKataKunci.SetFocus;
end;
```

```
procedure
TFDataMakanan.CBFieldCariKeyPr
ess(Sender: TObject;
  var Key: Char);
begin
  Key := #0;
end;

procedure
TFDataMakanan.EdKataKunciKeyUp
(Sender:  TObject;  var  Key:
Word;
  Shift: TShiftState);
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    If (Trim(CBFieldCari.Text)
= '') Then
      Exit;
    vQuery                  :=
vData.GetDataSearchMakanan(CBF
ieldCari.Text,
EdKataKunci.Text);
    DSDataMakanan.DataSet   :=
vQuery;

DBGridDataMakanan.DataSource
:= DSDataMakanan;
    DBControl.DataSource    :=
DSDataMakanan;
    vModeData := SearchData;
    SetKontrol(False);
    ViewDetail;
    SetStatusBar(vModeData);
    SetPropertyGridMakanan;
    TBTambah.Enabled := False;
    If  (vQuery.RecordCount  =
0) Then
      begin
        TBUbah.Enabled      :=
False;
        TBHapus.Enabled     :=
False;
      end
    Else
      begin
        TBUbah.Enabled      :=
True;
        TBHapus.Enabled     :=
True;
        If
FLoginUser.vHakAkses = 'Staff'
Then
        begin
          TBUbah.Enabled    :=
False;
          TBHapus.Enabled   :=
False;
        end;
      end;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataMakanan.TBTambahClick(Se
nder: TObject);
begin
  ClearFieldData;
  vModeData := InsertData;
  SetKontrol(True);
  EdIdMakanan.Text :='Auto';
  EdNamaMakanan.SetFocus;
  SetStatusBar(vModeData);
end;

procedure
TFDataMakanan.TBUbahClick(Send
er: TObject);
begin
  If (vModeData = BrowseData)
Then
    vModeData := UpdateData;
  SetKontrol(True);
  EdIdMakanan.Enabled       :=
False;
  EdNamaMakanan.SetFocus;
  SetStatusBar(vModeData);
end;
```

```
procedure
TFDataMakanan.TBHapusClick(Sen
der: TObject);
var
  vData : TPermintaanMakanan;
begin
  vModeData := DeleteData;
  SetStatusBar(vModeData);
  If  MessageDlg('Anda  Yakin
Menghapus Data Makanan Dengan
ID. Makanan : ' +
      EdIdMakanan.Text + ' ?'
,   mtConfirmation,   [mbYes,
mbNo],0) = mrYes Then
    begin
      try
        vData               :=
TPermintaanMakanan.Create;
        If
vData.DeleteMakanan(StrToInt(E
dIdMakanan.Text)) Then
          ShowMessage('Data
Makanan Berhasil Dihapus !')
        Else
          ShowMessage('Data
Makanan Gagal Dihapus !');
        RefreshData;
      finally
        FreeAndNil(vData);
      end;
    end
  else
    begin
      vModeData := BrowseData;
      SetStatusBar(vModeData);
    end;
end;

procedure
TFDataMakanan.TBSimpanClick(Se
nder: TObject);
var
  vData : TPermintaanMakanan;
begin
  If   not   CekValiditasInput
Then
    begin
      ShowMessage('Terjadi
Kesalahan  Dalam  Penginputan
Data, Mohon Diperbaiki');
      Exit;
    end;
  If (vModeData = InsertData)
Then
    begin
      If(MessageDlg('Apakah
Anda  Yakin  Ingin  Menambahkan
Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData           :=
TPermintaanMakanan.Create;
            If
(vData.CreateMakanan(

EdNamaMakanan.Text,

StrToInt(GetIdJenisMakananFrom
Combo(CBIdJenisMakanan))))
Then

ShowMessage('Data      Makanan
Berhasil Ditambahkan !')
            Else

ShowMessage('Data      Makanan
Ditambahkan !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end
  Else   If   (vModeData   =
UpdateData)  or  (vModeData  =
SearchData) Then
    begin
      If(MessageDlg('Apakah
Anda Yakin Ingin Mengubah Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
```

```
          try
            vData           :=
TPermintaanMakanan.Create;
            If
(vData.UpdateMakanan(

StrToInt(EdIdMakanan.Text),

EdNamaMakanan.Text,

StrToInt(GetIdJenisMakananFrom
Combo(CBIdJenisMakanan))))
Then

ShowMessage('Data      Makanan
Berhasil Diubah !')
            Else

ShowMessage('Data      Makanan
Gagal Diubah !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end;
  RefreshData;
end;

procedure
TFDataMakanan.TBBatalClick(Sen
der: TObject);
begin
  If (vModeData = InsertData)
or (vModeData =   UpdateData)
Then
    begin
      vModeData := BrowseData;
      ViewDetail;
      SetKontrol(False);
      SetStatusBar(vModeData);
    end
  Else     If(vModeData     =
SearchData) Then
    begin
      ViewDetail;
      SetKontrol(False);
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := True;
      TBHapus.Enabled := True;
    end;
end;

procedure
TFDataMakanan.TBRefreshClick(S
ender: TObject);
begin
  RefreshData
end;

procedure
TFDataMakanan.TBKeluarClick(Se
nder: TObject);
begin
  vPilihMakanan := False;
  Close;
end;

function
TFDataMakanan.CekValiditasInpu
t: Boolean;
begin
  If
(Trim(EdIdMakanan.Text)='')
Then
    Result := False
  Else
    Result := True;
end;

procedure
TFDataMakanan.SetPilihMakanan(
AFormSender: TForm;
  APilihMakanan: Boolean);
begin
  vPilihMakanan             :=
APilihMakanan;
  vFormSender := AFormSender;
end;

procedure
TFDataMakanan.TBPilihClick(Sen
der: TObject);
begin
```

```
  If      (vFormSender      =
TForm(FDataDetailBahanMakanan)
) Then

FDataDetailBahanMakanan.EdIdMa
kanan.Text := EdIdMakanan.Text
+ '-' + EdNamaMakanan.Text
  Else   If   (vFormSender   =
TForm(FDataPermintaanMakanan))
Then

FDataPermintaanMakanan.EdIdMak
anan.Text  :=  EdIdMakanan.Text
+ '-' + EdNamaMakanan.Text;
  vPilihMakanan := False;
  Close;
end;

procedure
TFDataMakanan.FormShow(Sender:
TObject);
begin
  LoadDataJenisMakananToCombo;
RefreshData;
end;

end.

unit UDataKelas;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs,  ComCtrls,  DBCtrls,
Grids,  DBGrids,  ImgList,  DB,
StdCtrls,
  ExtCtrls,   XPMan,   ToolWin,
ADODB;

type
  TModeData   =   (BrowseData,
UpdateData,        InsertData,
DeleteData);
type
  TFDataKelas = class(TForm)
    ToolBar: TToolBar;
    TBTambah: TToolButton;
    TBUbah: TToolButton;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
    TBRefresh: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    XPStyle: TXPManifest;
    PanelDetail: TPanel;
    LblNamaKelas: TLabel;
    LblIdKelas: TLabel;
    EdIdKelas: TEdit;
    EdNamaKelas: TEdit;
    DSDataKelas: TDataSource;
    ImageList: TImageList;
    PanelDBGrid: TPanel;
    DBGridDataKelas: TDBGrid;
    DBControl: TDBNavigator;
    StatusBar: TStatusBar;
    procedure
DBGridDataKelasCellClick(Colum
n: TColumn);
    procedure
DBControlClick(Sender:
TObject;               Button:
TNavigateBtn);
    procedure
TBKeluarClick(Sender:
TObject);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
TBTambahClick(Sender:
TObject);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
TBBatalClick(Sender: TObject);
```

```
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
    procedure
SetPropertyGridKelas();
    procedure
ClearFieldData();
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewDetail();
    procedure RefreshData();
    procedure
SetStatusBar(AModeData       :
TModeData);
    function
CekValiditasInput() : Boolean;
  public
    { Public declarations }
  end;

var
  FDataKelas: TFDataKelas;
  vQuery : TADOQuery;
  vModeData : TModeData;

implementation

uses UIRI;

{$R *.dfm}

{ TFDataKelas }

procedure
TFDataKelas.ClearFieldData;
begin
  EdIdKelas.Clear;
  EdNamaKelas.Clear;
end;

procedure
TFDataKelas.RefreshData;
var
  vData : TIRI;
begin
  try
    vData := TIRI.Create;
    vQuery := vData.GetKelas;
    DSDataKelas.DataSet     :=
vQuery;
    DBGridDataKelas.DataSource
:= DSDataKelas;
    DBControl.DataSource    :=
DSDataKelas;
    //FIRST LOAD - CHECK THE
STATUS DATA IF EMPTY OR ANY
DATA IS IN

//============================
============================
=====
    If  (vQuery.RecordCount  =
0) Then
      begin
        ShowMessage('Data
Kelas   Kosong,   Anda   Harus
Melakukan Penambahan Data !');
        ClearFieldData;
        SetKontrol(True);
        TBTambahClick(Nil);
        TBBatal.Enabled     :=
False;
      end
    else
      begin
        SetKontrol(False);
        vModeData           :=
BrowseData;
        ClearFieldData;
        ViewDetail;
      end;

//============================
============================
=====
    SetStatusBar(vModeData);
    SetPropertyGridKelas;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataKelas.SetKontrol(ABool:
Boolean);
```

```
begin
  PanelDetail.Enabled       :=
ABool;
  PanelDBGrid.Enabled  :=  Not
ABool;
  TBTambah.Enabled    :=    Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBRefresh.Enabled   :=   Not
ABool;
end;

procedure
TFDataKelas.SetPropertyGridKel
as;
var
  vCount : Integer;
begin
  With DBGridDataKelas do
    begin
      for vCount :=0 to 1 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
      Columns[0].Width := 100;
      Columns[1].Width := 250;
      Columns[0].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;

procedure
TFDataKelas.SetStatusBar(AMode
Data: TModeData);
begin
  If (AModeData = InsertData)
Then
    StatusBar.Panels[0].Text
:=  'Insert Mode - Penambahan
Data'
  Else    If    (AModeData    =
UpdateData) Then
    StatusBar.Panels[0].Text
:= 'Update Mode - Pengubahan
Data'
  Else    If    (AModeData    =
BrowseData) Then
    StatusBar.Panels[0].Text
:= 'Browse Mode - Penelusuran
Data'
  Else    If    (AModeData    =
DeleteData) Then
    StatusBar.Panels[0].Text
:= 'Delete Mode - Penghapusan
Data'
 Else
    StatusBar.Panels[0].Text
:= 'Unknown Mode';
  StatusBar.Panels[1].Text  :=
'Data     Kelas     Ke-'     +
IntToStr(vQuery.RecNo)   +   '
Dari            '            +
IntToStr(vQuery.RecordCount) +
' Data';
end;

procedure
TFDataKelas.ViewDetail;
begin
  EdIdKelas.Text            :=
DBGridDataKelas.Fields[0].Text
;
  EdNamaKelas.Text          :=
DBGridDataKelas.Fields[1].Text
;
end;

procedure
TFDataKelas.DBGridDataKelasCel
lClick(Column: TColumn);
begin
  ViewDetail;
SetStatusBar(vModeData);
```

```
end;

procedure
TFDataKelas.DBControlClick(Sen
der: TObject;
  Button: TNavigateBtn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataKelas.TBKeluarClick(Send
er: TObject);
begin
  Close;
end;

procedure
TFDataKelas.TBRefreshClick(Sen
der: TObject);
begin
  RefreshData;
end;

procedure
TFDataKelas.TBTambahClick(Send
er: TObject);
begin
  ClearFieldData;
  vModeData := InsertData;
  SetKontrol(True);
  EdIdKelas.Enabled := True;
  EdIdKelas.SetFocus;
  SetStatusBar(vModeData);
end;

procedure
TFDataKelas.TBUbahClick(Sender
: TObject);
begin
  If (vModeData = BrowseData)
Then
    vModeData := UpdateData;
  SetKontrol(True);
  EdIdKelas.Enabled := False;
  SetStatusBar(vModeData);
end;

procedure
TFDataKelas.TBHapusClick(Sende
r: TObject);
var
  vData : TIRI;
begin
  vModeData := DeleteData;
  SetStatusBar(vModeData);
  If  MessageDlg('Anda  Yakin
Menghapus  Data  Kelas  Dengan
ID. Kelas : ' +
      EdIdKelas.Text + ' ?' ,
mtConfirmation,        [mbYes,
mbNo],0) = mrYes Then
    begin
      try
        vData := TIRI.Create;
        If
vData.DeleteKelas(EdIdKelas.Te
xt) Then
          ShowMessage('Data
Kelas Berhasil Dihapus !')
        Else
          ShowMessage('Data
Kelas Gagal Dihapus !');
        RefreshData;
      finally
        FreeAndNil(vData);
      end;
    end
  else
    begin
      vModeData := BrowseData;
      SetStatusBar(vModeData);
    end;
end;

procedure
TFDataKelas.TBSimpanClick(Send
er: TObject);
var
  vData : TIRI;
begin
  If   not   CekValiditasInput
Then
    begin
```

```
      ShowMessage('Terjadi
Kesalahan  Dalam  Penginputan
Data, Mohon Diperbaiki');
      Exit;
    end;
  If (vModeData = InsertData)
Then
    begin
      If(MessageDlg('Apakah
Anda  Yakin  Ingin  Menambahkan
Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData          :=
TIRI.Create;
            If
(vData.CreateKelas(

EdIdKelas.Text,

EdNamaKelas.Text)) Then

ShowMessage('Data        Kelas
Berhasil Ditambahkan !')
            Else

ShowMessage('Data  Kelas  Gagal
Ditambahkan !');
            RefreshData;
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end
  Else   If   (vModeData   =
UpdateData) Then
    begin
      If(MessageDlg('Apakah
Anda Yakin Ingin Mengubah Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData          :=
TIRI.Create;
            If
(vData.UpdateKelas(

EdIdKelas.Text,

EdNamaKelas.Text)) Then

ShowMessage('Data        Kelas
Berhasil Diubah !')
            Else

ShowMessage('Data  Kelas  Gagal
Diubah !');
            RefreshData;
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end;
end;

procedure
TFDataKelas.TBBatalClick(Sende
r: TObject);
begin
  If (vModeData = InsertData)
or (vModeData =  UpdateData)
Then
    begin
      vModeData := BrowseData;
      ViewDetail;
      SetKontrol(False);
      SetStatusBar(vModeData);
    end;
end;

function
TFDataKelas.CekValiditasInput:
Boolean;
begin
  If(Trim(EdIdKelas.Text)   =
'') or (Trim(EdNamaKelas.Text)
= '') Then
    Result := False
```

```
  Else
    Result := True;
end;

procedure
TFDataKelas.FormShow(Sender:
TObject);
begin
  RefreshData;
end;

end.

unit UDataJenisMakanan;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs, ComCtrls, StdCtrls,
DBCtrls,  Grids,  DBGrids,  DB,
ImgList, XPMan,
  ExtCtrls, ToolWin, ADODB;

type
  TModeData   =   (BrowseData,
UpdateData,        InsertData,
DeleteData, SearchData);
type
  TFDataJenisMakanan         =
class(TForm)
    ToolBar: TToolBar;
    TBTambah: TToolButton;
    TBUbah: TToolButton;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
    TBRefresh: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    PanelDetail: TPanel;
    LblIdJenisMakanan: TLabel;
    LblNamaJenisMakanan:
TLabel;
    EdIdJenisMakanan: TEdit;
    EdNamaJenisMakanan: TEdit;
    XPStyle: TXPManifest;
    ImageList: TImageList;
    DSDataJenisMakanan:
TDataSource;
    PanelDBGrid: TPanel;
    DBGridDataJenisMakanan:
TDBGrid;
    DBControl: TDBNavigator;
    StatusBar: TStatusBar;
    PanelCari: TPanel;
    LblFieldCari: TLabel;
    LblKataKunci: TLabel;
    CBFieldCari: TComboBox;
    EdKataKunci: TEdit;
    procedure
TBTambahClick(Sender:
TObject);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
TBBatalClick(Sender: TObject);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
TBKeluarClick(Sender:
TObject);
    procedure
DBGridDataJenisMakananCellClic
k(Column: TColumn);
    procedure
DBControlClick(Sender:
TObject;               Button:
TNavigateBtn);
    procedure
CBFieldCariClick(Sender:
TObject);
    procedure
CBFieldCariKeyPress(Sender:
TObject; var Key: Char);
```

```
    procedure
EdKataKunciKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
    procedure
SetPropertyGridJenisMakanan();
    procedure
ClearFieldData();
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewDetail();
    procedure RefreshData();
    procedure
SetStatusBar(AModeData       :
TModeData);
    function
CekValiditasInput() : Boolean;
  public
    { Public declarations }
  end;

var
  FDataJenisMakanan:
TFDataJenisMakanan;
  vQuery : TADOQuery;
  vModeData : TModeData;

implementation

uses       UPermintaanMakanan,
UDataPasien, ULoginUser;

{$R *.dfm}

{ TFDataJenisMakanan }

procedure
TFDataJenisMakanan.ClearFieldD
ata;
begin
  EdIdJenisMakanan.Clear;
  EdNamaJenisMakanan.Clear;
  EdKataKunci.Clear;
end;

procedure
TFDataJenisMakanan.RefreshData
;
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    vQuery                  :=
vData.GetJenisMakanan;
    DSDataJenisMakanan.DataSet
:= vQuery;

DBGridDataJenisMakanan.DataSou
rce := DSDataJenisMakanan;
    DBControl.DataSource    :=
DSDataJenisMakanan;
    //FIRST LOAD - CHECK THE
STATUS DATA IF EMPTY OR ANY
DATA IS IN

//==========================
==========================
=====
    If  (vQuery.RecordCount  =
0) Then
      begin
        ShowMessage('Data
Jenis  Makanan  Kosong,  Anda
Harus  Melakukan  Penambahan
Data !');
        ClearFieldData;
        SetKontrol(True);
        TBTambahClick(Nil);
        TBBatal.Enabled     :=
False;
      end
    else
      begin
        SetKontrol(False);
        vModeData           :=
BrowseData;
        ClearFieldData;
        ViewDetail;
      end;
```

```
//============================
============================
=====
    SetStatusBar(vModeData);

SetPropertyGridJenisMakanan;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataJenisMakanan.SetKontrol(
ABool: Boolean);
begin
  PanelDetail.Enabled       :=
ABool;
  PanelDBGrid.Enabled  :=  Not
ABool;
  PanelCari.Enabled   :=   Not
ABool;
  TBTambah.Enabled    :=    Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBRefresh.Enabled   :=   Not
ABool;
  If   FLoginUser.vHakAkses   =
'Staff' Then
    begin
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := False;
      TBHapus.Enabled       :=
False;
      TBSimpan.Enabled      :=
False;
      TBBatal.Enabled       :=
False;
    end;
end;

procedure
TFDataJenisMakanan.SetPropertyGridJenisMakanan;
var
  vCount : Integer;
begin
  With  DBGridDataJenisMakanan
do
    begin
      for vCount :=0 to 1 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
      Columns[0].Width := 90;
      Columns[1].Width := 260;
      Columns[0].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;

procedure
TFDataJenisMakanan.SetStatusBa
r(AModeData: TModeData);
begin
  If (AModeData = InsertData)
Then
    StatusBar.Panels[0].Text
:=  'Insert Mode - Penambahan
Data'
  Else    If    (AModeData    =
UpdateData) Then
    StatusBar.Panels[0].Text
:= 'Update Mode - Pengubahan
Data'
  Else    If    (AModeData    =
BrowseData) Then
    StatusBar.Panels[0].Text
:= 'Browse Mode - Penelusuran
Data'
```

```
  Else    If    (AModeData    =
DeleteData) Then
    StatusBar.Panels[0].Text
:= 'Delete Mode - Penghapusan
Data'
  Else    If    (AModeData    =
SearchData) Then
    StatusBar.Panels[0].Text
:= 'Search Mode - Pencarian
Data'
  Else
    StatusBar.Panels[0].Text
:= 'Unknown Mode';
  StatusBar.Panels[1].Text  :=
'Data Jenis Makanan Ke-' +
IntToStr(vQuery.RecNo)   +   '
Dari            '            +
IntToStr(vQuery.RecordCount) +
' Data';
end;

procedure
TFDataJenisMakanan.ViewDetail;
begin
  EdIdJenisMakanan.Text     :=
DBGridDataJenisMakanan.Fields[
0].Text;
  EdNamaJenisMakanan.Text   :=
DBGridDataJenisMakanan.Fields[
1].Text;
end;

procedure
TFDataJenisMakanan.TBTambahCli
ck(Sender: TObject);
begin
  ClearFieldData;
  vModeData := InsertData;
  SetKontrol(True);
  EdIdJenisMakanan.Text     :=
'Auto';
  EdNamaJenisMakanan.SetFocus;
  SetStatusBar(vModeData);
end;

procedure
TFDataJenisMakanan.TBUbahClick
(Sender: TObject);
begin
  If (vModeData = BrowseData)
Then
    vModeData := UpdateData;
  SetKontrol(True);
  EdIdJenisMakanan.Enabled  :=
False;
  EdNamaJenisMakanan.SetFocus;
  SetStatusBar(vModeData);
end;

procedure
TFDataJenisMakanan.TBHapusClic
k(Sender: TObject);
var
  vData : TPermintaanMakanan;
begin
  vModeData := DeleteData;
  SetStatusBar(vModeData);
  If  MessageDlg('Anda  Yakin
Menghapus Data Jenis Makanan
Dengan ID. Jenis Makanan : ' +
      EdIdJenisMakanan.Text  +
' ?' , mtConfirmation, [mbYes,
mbNo],0) = mrYes Then
    begin
      try
        vData               :=
TPermintaanMakanan.Create;
        If
vData.DeleteJenisMakanan(StrTo
Int(EdIdJenisMakanan.Text))
Then
          ShowMessage('Data
Jenis Makanan Berhasil Dihapus
!')
        Else
          ShowMessage('Data
Jenis  Makanan  Gagal  Dihapus
!');
        RefreshData;
      finally
        FreeAndNil(vData);
      end;
    end
  else
    begin
      vModeData := BrowseData;
```

```
      SetStatusBar(vModeData);
    end;
end;

procedure
TFDataJenisMakanan.TBSimpanCli
ck(Sender: TObject);
var
  vData : TPermintaanMakanan;
begin
  If   not   CekValiditasInput
Then
    begin
      ShowMessage('Terjadi
Kesalahan  Dalam  Penginputan
Data, Mohon Diperbaiki');
      Exit;
    end;
  If (vModeData = InsertData)
Then
    begin
      If(MessageDlg('Apakah
Anda  Yakin  Ingin  Menambahkan
Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData           :=
TPermintaanMakanan.Create;
            If
(vData.CreateJenisMakanan(

EdNamaJenisMakanan.Text)) Then

ShowMessage('Data        Jenis
Makanan  Berhasil  Ditambahkan
!')
            Else

ShowMessage('Data        Jenis
Makanan Ditambahkan !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end
  Else   If   (vModeData   =
UpdateData)  or  (vModeData  =
SearchData) Then
    begin
      If(MessageDlg('Apakah
Anda Yakin Ingin Mengubah Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData           :=
TPermintaanMakanan.Create;
            If
(vData.UpdateJenisMakanan(

StrToInt(EdIdJenisMakanan.Text
),

EdNamaJenisMakanan.Text)) Then

ShowMessage('Data        Jenis
Makanan Berhasil Diubah !')
            Else

ShowMessage('Data        Jenis
Makanan Gagal Diubah !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end;
  RefreshData;
end;

procedure
TFDataJenisMakanan.TBBatalClic
k(Sender: TObject);
begin
  If (vModeData = InsertData)
or (vModeData =  UpdateData)
Then
    begin
      vModeData := BrowseData;
      ViewDetail;
      SetKontrol(False);
```

```
      SetStatusBar(vModeData);
    end
  Else      If(vModeData      =
SearchData) Then
    begin
      ViewDetail;
      SetKontrol(False);
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := True;
      TBHapus.Enabled := True;
    end;
end;
procedure
TFDataJenisMakanan.TBRefreshCl
ick(Sender: TObject);
begin
  RefreshData;
end;

procedure
TFDataJenisMakanan.TBKeluarCli
ck(Sender: TObject);
begin
  Close;
end;

procedure
TFDataJenisMakanan.DBGridDataJ
enisMakananCellClick(
  Column: TColumn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataJenisMakanan.DBControlCl
ick(Sender: TObject;
  Button: TNavigateBtn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataJenisMakanan.CBFieldCari
Click(Sender: TObject);
begin
  EdKataKunci.Clear;
  EdKataKunci.SetFocus;
end;

procedure
TFDataJenisMakanan.CBFieldCari
KeyPress(Sender: TObject;
  var Key: Char);
begin
  Key := #0;
end;

procedure
TFDataJenisMakanan.EdKataKunci
KeyUp(Sender: TObject;
  var   Key:   Word;   Shift:
TShiftState);
var
  vData : TPermintaanMakanan;
begin
  try
    vData                    :=
TPermintaanMakanan.Create;
    If (Trim(CBFieldCari.Text)
= '') Then
      Exit;
    vQuery                   :=
vData.GetDataSearchJenisMakana
n(CBFieldCari.Text,
EdKataKunci.Text);
    DSDataJenisMakanan.DataSet
:= vQuery;

DBGridDataJenisMakanan.DataSou
rce := DSDataJenisMakanan;
    DBControl.DataSource    :=
DSDataJenisMakanan;
    vModeData := SearchData;
    SetKontrol(False);
    ViewDetail;
    SetStatusBar(vModeData);

SetPropertyGridJenisMakanan;
    TBTambah.Enabled := False;
    If  (vQuery.RecordCount  =
0) Then
      begin
```

```
        TBUbah.Enabled      :=
False;
        TBHapus.Enabled     :=
False;
      end
    Else
      begin
        TBUbah.Enabled      :=
True;
        TBHapus.Enabled     :=
True;
        If
FLoginUser.vHakAkses = 'Staff'
Then
        begin
          TBUbah.Enabled    :=
False;
          TBHapus.Enabled   :=
False;
        end;
      end;
  finally
    FreeAndNil(vData);
  end;
end;

function
TFDataJenisMakanan.CekValidita
sInput: Boolean;
begin

If(Trim(EdIdJenisMakanan.Text)
=            '')            or
(Trim(EdNamaJenisMakanan.Text)
= '') Then
    Result := False
  Else
    Result := True;
end;
procedure
TFDataJenisMakanan.FormShow(Se
nder: TObject);
begin
  RefreshData;
end;

end.
```

```
unit UDataIRI;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs,  ComCtrls,  DBCtrls,
Grids,   DBGrids,   StdCtrls,
ExtCtrls, XPMan,
  ImgList, DB, ToolWin, ADODB,
StrUtils;

type
  TModeData   =   (BrowseData,
UpdateData,        InsertData,
DeleteData, SearchData);
type
  TFDataIRI = class(TForm)
    ToolBar: TToolBar;
    TBTambah: TToolButton;
    TBUbah: TToolButton;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
    TBRefresh: TToolButton;
    TBCetak: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    DSDataIRI: TDataSource;
    ImageList: TImageList;
    XPStyle: TXPManifest;
    PanelDetail: TPanel;
    LblNoMedRec: TLabel;
    LblNamaPasien: TLabel;
    LblTglMasuk: TLabel;
    LblCaraBayar: TLabel;
    EdNoDokIRI: TEdit;
    EdNoMedRec: TEdit;
    EdCaraBayar: TEdit;
    DTTglMasuk:
TDateTimePicker;
    PanelDBGrid: TPanel;
    DBGridDataIRI: TDBGrid;
    DBControl: TDBNavigator;
```

```
    PanelCari: TPanel;
    LblFieldCari: TLabel;
    LblKataKunci: TLabel;
    CBFieldCari: TComboBox;
    EdKataKunci: TEdit;
    StatusBar: TStatusBar;
    LblTglKeluar: TLabel;
    DTTglKeluar:
TDateTimePicker;
    EdCaraMasuk: TEdit;
    LblCaraMasuk: TLabel;
    CBIdRuang: TComboBox;
    LblIdRuang: TLabel;
    BtnPilihPasien: TButton;
    LblStatusID: TLabel;
    TBPilih: TToolButton;
    procedure
BtnPilihPasienClick(Sender:
TObject);
    procedure
TBTambahClick(Sender:
TObject);
    procedure
TBKeluarClick(Sender:
TObject);
    procedure
TBBatalClick(Sender: TObject);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
DBGridDataIRICellClick(Column:
TColumn);
    procedure
DBControlClick(Sender:
TObject;               Button:
TNavigateBtn);
    procedure
CBFieldCariKeyPress(Sender:
TObject; var Key: Char);
    procedure
CBFieldCariClick(Sender:
TObject);
    procedure
EdKataKunciKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure
EdNoDokIRIKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure
EdNoDokIRIKeyPress(Sender:
TObject; var Key: Char);
    procedure
TBPilihClick(Sender: TObject);
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
    procedure
SetPropertyGridIRI();
    procedure
ClearFieldData();
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewDetail();
    procedure RefreshData();
    procedure
SetStatusBar(AModeData       :
TModeData);
    procedure
LoadDataRuangToCombo();
    function
GetIDRuangFromCombo(ACombo
:TComboBox) : String;
    function
GetIDPasienFromEd(AEd : TEdit)
: String;
    function
CekValiditasInput() : Boolean;
  public
    { Public declarations }
    procedure
SetPilihIRI(APilihIRI        :
Boolean);
  end;
```

```
var
  FDataIRI: TFDataIRI;
  vQuery : TADOQuery;
  vModeData : TModeData;
  vStatusID : Boolean;
  vPilihIRI : Boolean;

implementation

uses    UIRI,    UDataPasien,
UDataPermintaanMakanan;

{$R *.dfm}

{ TFDataIRI }

procedure
TFDataIRI.ClearFieldData;
begin
  EdNoDokIRI.Clear;
  EdNoMedRec.Clear;
  CBIdRuang.Text := '';
  DTTglMasuk.Date := Date;
  DTTglKeluar.Date := Date;
  EdCaraMasuk.Clear;
  EdCaraBayar.Clear;
  EdKataKunci.Clear;
end;

function
TFDataIRI.GetIDPasienFromEd(AE
d: TEdit): String;
var
  vPos : integer;
begin
  if AEd.Text <> '' then
    begin
      vPos    :=    Pos('-',
AEd.Text);
      Result               :=
LeftStr(AEd.Text,vPos-1);
    end
  else
    Result := '';
end;

function
TFDataIRI.GetIDRuangFromCombo(
ACombo: TComboBox): String;
var
  vPos : integer;
begin
  if ACombo.Text <> '' then
    begin
      vPos     :=     Pos('-
',ACombo.Text);
      Result               :=
LeftStr(ACombo.Text,vPos-1);
    end
  else
    Result := '';
end;

procedure
TFDataIRI.LoadDataRuangToCombo
;
var
  vData : TIRI;
  vQueryTemp : TADOQuery;
begin
  try
    vData := TIRI.Create;
    vQueryTemp             :=
vData.GetRuang;
    CBIdRuang.Clear;
    While  Not  vQueryTemp.Eof
Do
      begin

CBIdRuang.Items.Add(vQueryTemp
.Fields[0].Text   +   '-'   +
vQueryTemp.Fields[1].Text);
        vQueryTemp.Next;
      end;
  finally
    FreeAndNil(vQueryTemp);
    FreeAndNil(vData);
  end;
end;

procedure
TFDataIRI.RefreshData;
var
  vData : TIRI;
```

```
begin
  try
    vData := TIRI.Create;
    vQuery := vData.GetIRI;
    DSDataIRI.DataSet       :=
vQuery;
    DBGridDataIRI.DataSource
:= DSDataIRI;
    DBControl.DataSource    :=
DSDataIRI;
    //FIRST LOAD - CHECK THE
STATUS DATA IF EMPTY OR ANY
DATA IS IN

//============================
============================
=====
    If  (vQuery.RecordCount  =
0) Then
      begin
        ShowMessage('Data  IRI
Kosong, Anda Harus Melakukan
Penambahan Data !');
        ClearFieldData;
        SetKontrol(True);
        TBTambahClick(Nil);
        TBBatal.Enabled     :=
False;
      end
    else
      begin
        SetKontrol(False);
        CBFieldCari.ItemIndex
:= 0;
        vModeData           :=
BrowseData;
        ClearFieldData;
        ViewDetail;
      end;

//============================
============================
=====
    SetStatusBar(vModeData);
    SetPropertyGridIRI;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataIRI.SetKontrol(ABool:
Boolean);
begin
  PanelDetail.Enabled       :=
ABool;
  PanelDBGrid.Enabled  :=  Not
ABool;
  PanelCari.Enabled   :=   Not
ABool;
  TBTambah.Enabled    :=    Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBRefresh.Enabled   :=   Not
ABool;
  TBCetak.Enabled    :=    Not
ABool;
  TBPilih.Enabled    :=    Not
ABool;
  If (vPilihIRI = True) Then
    TBPilih.Visible := True
  Else
    TBPilih.Visible := False;
  LblStatusID.Visible       :=
False;
end;

procedure
TFDataIRI.SetPropertyGridIRI;
var
  vCount : Integer;
begin
  With DBGridDataIRI do
    begin
      for vCount :=0 to 8 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
```

```
        end;
      Columns[0].Width := 90;
      Columns[0].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;

procedure
TFDataIRI.SetStatusBar(AModeDa
ta: TModeData);
begin
  If (AModeData = InsertData)
Then
    StatusBar.Panels[0].Text
:=  'Insert Mode - Penambahan
Data'
  Else    If    (AModeData    =
UpdateData) Then
    StatusBar.Panels[0].Text
:= 'Update Mode - Pengubahan
Data'
  Else    If    (AModeData    =
BrowseData) Then
    StatusBar.Panels[0].Text
:= 'Browse Mode - Penelusuran
Data'
  Else    If    (AModeData    =
DeleteData) Then
    StatusBar.Panels[0].Text
:= 'Delete Mode - Penghapusan
Data'
  Else    If    (AModeData    =
SearchData) Then
    StatusBar.Panels[0].Text
:= 'Search Mode - Pencarian
Data'
  Else
    StatusBar.Panels[0].Text
:= 'Unknown Mode';
  StatusBar.Panels[1].Text  :=
'Status Data IRI : Data Ke-' +
IntToStr(vQuery.RecNo)   +   '
Dari           '           +
IntToStr(vQuery.RecordCount) +
' Data';
  StatusBar.Panels[2].Text  :=
'IRI-RS. HasanSadikin';
end;

procedure
TFDataIRI.ViewDetail;
begin
  EdNoDokIRI.Text           :=
DBGridDataIRI.Fields[0].Text;
  EdNoMedRec.Text           :=
DBGridDataIRI.Fields[1].Text +
'-'                          +
DBGridDataIRI.Fields[2].Text;
  CBIdRuang.Text            :=
DBGridDataIRI.Fields[3].Text +
'-'                          +
DBGridDataIRI.Fields[4].Text;
  If (vQuery.RecordCount = 0)
Then
    begin
      DTTglMasuk.Date := Date;
      DTTglKeluar.Date      :=
Date;
    end
  else
    begin
      DTTglMasuk.Date       :=
DBGridDataIRI.Fields[5].Value;
      DTTglKeluar.Date      :=
DBGridDataIRI.Fields[6].Value;
    end;
  EdCaraMasuk.Text          :=
DBGridDataIRI.Fields[7].Text;
  EdCaraBayar.Text          :=
DBGridDataIRI.Fields[8].Text;
end;

procedure
TFDataIRI.BtnPilihPasienClick(
Sender: TObject);
begin

FDataPasien.SetPilihPasien(Tru
e);
  FDataPasien.ShowModal;
end;
```

```
procedure
TFDataIRI.TBTambahClick(Sender
: TObject);
begin
  ClearFieldData;
  vModeData := InsertData;
  SetKontrol(True);
  EdNoMedRec.Enabled := True;
  EdNoDokIRI.Enabled := True;
  BtnPilihPasien.Enabled    :=
True;
  EdNoDokIRI.SetFocus;
  LblStatusID.Visible := True;
  LblStatusID.Font.Color    :=
clBlack;
  LblStatusID.Caption       :=
'Masukkan No. Dok. IRI';
  SetStatusBar(vModeData);
end;

procedure
TFDataIRI.TBKeluarClick(Sender
: TObject);
begin
  vPilihIRI := False;
  Close;
end;

procedure
TFDataIRI.TBBatalClick(Sender:
TObject);
begin
  If (vModeData = InsertData)
or (vModeData =   UpdateData)
Then
    begin
      vModeData := BrowseData;
      ViewDetail;
      SetKontrol(False);
      SetStatusBar(vModeData);
    end
  Else     If(vModeData     =
SearchData) Then
    begin
      ViewDetail;
      SetKontrol(False);
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := True;
      TBHapus.Enabled := True;
    end;
end;

procedure
TFDataIRI.TBRefreshClick(Sende
r: TObject);
begin
  RefreshData;
end;

procedure
TFDataIRI.TBUbahClick(Sender:
TObject);
begin
  If (vModeData = BrowseData)
Then
    vModeData := UpdateData;
  SetKontrol(True);
  EdNoMedRec.Enabled := False;
  EdNoDokIRI.Enabled := False;
  BtnPilihPasien.Enabled    :=
False;
  vStatusID := True;
  SetStatusBar(vModeData);
end;

procedure
TFDataIRI.TBHapusClick(Sender:
TObject);
var
  vData : TIRI;
begin
  vModeData := DeleteData;
  SetStatusBar(vModeData);
  If  MessageDlg('Anda  Yakin
Menghapus Data IRI Dengan No.
Dok. IRI : ' +
      EdNoDokIRI.Text + ' ?' ,
mtConfirmation,        [mbYes,
mbNo],0) = mrYes Then
    begin
      try
        vData := TIRI.Create;
```

```
        If
vData.DeleteIRI(EdNoDokIRI.Tex
t) Then
          ShowMessage('Data
IRI Berhasil Dihapus !')
        Else
          ShowMessage('Data
IRI Gagal Dihapus !');
        RefreshData;
      finally
        FreeAndNil(vData);
      end;
    end
  else
    begin
      vModeData := BrowseData;
      SetStatusBar(vModeData);
    end;
end;

procedure
TFDataIRI.TBSimpanClick(Sender
: TObject);
var
  vData : TIRI;
begin
  If   not   CekValiditasInput
Then
    begin
      ShowMessage('Terjadi
Kesalahan  Dalam  Penginputan
Data, Mohon Diperbaiki');
      Exit;
    end;
  If (vModeData = InsertData)
Then
    begin
      If(MessageDlg('Apakah
Anda  Yakin  Ingin  Menambahkan
Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData          :=
TIRI.Create;
            If
(vData.CreateIRI(
EdNoDokIRI.Text,

GetIDPasienFromEd(EdNoMedRec),

GetIDRuangFromCombo(CBIdRuang)
,

DTTglMasuk.Date,

DTTglKeluar.Date,

EdCaraMasuk.Text,

EdCaraBayar.Text)) Then

ShowMessage('Data IRI Berhasil
Ditambahkan !')
            Else

ShowMessage('Data  IRI  Gagal
Ditambahkan !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end
  Else   If   (vModeData   =
UpdateData)  or  (vModeData  =
SearchData) Then
    begin
      If(MessageDlg('Apakah
Anda Yakin Ingin Mengubah Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData          :=
TIRI.Create;
            If
(vData.UpdateIRI(

EdNoDokIRI.Text,

GetIDPasienFromEd(EdNoMedRec),
```

```
GetIDRuangFromCombo(CBIdRuang)
,

DTTglMasuk.Date,

DTTglKeluar.Date,

EdCaraMasuk.Text,

EdCaraBayar.Text)) Then

ShowMessage('Data IRI Berhasil
Diubah !')
           Else

ShowMessage('Data  IRI  Gagal
Diubah !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end;
  RefreshData;
end;

procedure
TFDataIRI.DBGridDataIRICellCli
ck(Column: TColumn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataIRI.DBControlClick(Sende
r:      TObject;      Button:
TNavigateBtn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataIRI.CBFieldCariKeyPress(
Sender:  TObject;  var  Key:
Char);
begin
  Key := #0;
end;

procedure
TFDataIRI.CBFieldCariClick(Sen
der: TObject);
begin
  EdKataKunci.Clear;
  EdKataKunci.SetFocus;
end;

procedure
TFDataIRI.EdKataKunciKeyUp(Sen
der: TObject; var Key: Word;
  Shift: TShiftState);
var
  vData : TIRI;
begin
  try
    vData := TIRI.Create;
    If (Trim(CBFieldCari.Text)
= '') Then
      Exit;
    vQuery                  :=
vData.GetDataSearchIRI(CBField
Cari.Text, EdKataKunci.Text);
    DSDataIRI.DataSet       :=
vQuery;
    DBGridDataIRI.DataSource
:= DSDataIRI;
    DBControl.DataSource    :=
DSDataIRI;
    vModeData := SearchData;
    SetKontrol(False);
    ViewDetail;
    SetStatusBar(vModeData);
    SetPropertyGridIRI;
    TBTambah.Enabled := False;
    If  (vQuery.RecordCount  =
0) Then
      begin
        TBUbah.Enabled      :=
False;
        TBHapus.Enabled     :=
False;
```

```
      end
    Else
      begin
        TBUbah.Enabled      :=
True;
        TBHapus.Enabled     :=
True;
      end;
  finally
    FreeAndNil(vData);
  end;
end;

function
TFDataIRI.CekValiditasInput:
Boolean;
begin
  If(vStatusID  =  False)  or
(Trim(EdNoMedRec.Text)  =  '')
Then
    Result := False
  Else
    Result := True;
end;

procedure
TFDataIRI.EdNoDokIRIKeyUp(Send
er: TObject; var Key: Word;
  Shift: TShiftState);
var
  vData : TIRI;
  vQueryTemp : TADOQuery;
begin
  vStatusID := False;
  If  Trim(EdNoDokIRI.Text)  =
'' Then
    begin
      LblStatusID.Caption   :=
'Masukkan No. Dok IRI';
      Exit;
    end;
  If
Length(Trim(EdNoDokIRI.Text))<
10 Then
    begin
      LblStatusID.Font.Color
:= ClBlue;
      LblStatusID.Caption   :=
'No.  Dok  IRI  (10  Karakter
Angka)';
      Exit;
    end;
  try
    vData := TIRI.Create;
    vQueryTemp              :=
vDAta.GetDataSearchIRI('no_dok
_iri',EdNoDokIRI.Text);
    If (vQueryTemp.RecordCount
>0) Then
      begin
        LblStatusID.Font.Color
:= clRed;
        LblStatusID.Caption :=
'Tidak Tersedia';
        vStatusID := False;
      end
    Else
      begin
        LblStatusID.Font.Color
:= clGreen;
        LblStatusID.Caption :=
'Tersedia';
        vStatusID := True;
      end;
  finally
    FreeAndNil(vQueryTemp);
    FreeAndNil(vData);
  end;
end;

procedure
TFDataIRI.EdNoDokIRIKeyPress(S
ender:   TObject;   var   Key:
Char);
begin
  If  (Key=#8)  or  ((Key>=#48)
and (Key<=#57)) Then
    Exit
  Else
    Key := #0
end;

procedure
TFDataIRI.TBPilihClick(Sender:
TObject);
```

```
begin
  vPilihIRI := False;

FDataPermintaanMakanan.EdNoDok
IRI.Text := EdNoDokIRI.Text +
'-(' + EdNoMedRec.Text + ')';
  Close;
end;

procedure
TFDataIRI.SetPilihIRI(APilihIR
I: Boolean);
begin
  vPilihIRI := APilihIRI;
end;

procedure
TFDataIRI.FormShow(Sender:
TObject);
begin
  LoadDataRuangToCombo;
RefreshData;
end;

end.

unit UDataDetaiBahanMakanan;

interface

uses
  Windows, Messages, SysUtils,
Variants, Classes, Graphics,
Controls, Forms,
  Dialogs, DB, ImgList, XPMan,
ComCtrls, DBCtrls, Grids,
DBGrids, StdCtrls,
  ExtCtrls, ToolWin, ADODB,
StrUtils;

type
  TModeData = (BrowseData,
UpdateData, InsertData,
DeleteData, SearchData);
type
  TFDataDetailBahanMakanan =
class(TForm)
    ToolBar: TToolBar;
    TBTambah: TToolButton;
    TBUbah: TToolButton;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
    TBRefresh: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    PanelDetail: TPanel;
    LblIdDetail: TLabel;
    LblIdBahan: TLabel;
    LblIdMakanan: TLabel;
    EdIdDetail: TEdit;
    EdJumlahBahan: TEdit;
    PanelDBGrid: TPanel;
    PanelCari: TPanel;
    LblFieldCari: TLabel;
    LblKataKunci: TLabel;
    CBFieldCari: TComboBox;
    EdKataKunci: TEdit;
    StatusBar: TStatusBar;
    XPStyle: TXPManifest;
    ImageList: TImageList;
    DSDataDetail: TDataSource;
    LblGram: TLabel;
    LblJumlahBahan: TLabel;
    EdIdBahan: TEdit;
    BtnPilihBahan: TButton;
    EdIdMakanan: TEdit;
    BtnPilihMakanan: TButton;
    DBGridDetailBahan:
TDBGrid;
    DBControlDetail:
TDBNavigator;
    DSDataMaster: TDataSource;
    PanelMakanan: TPanel;
    DBGridDataMakanan:
TDBGrid;
    DBControl: TDBNavigator;
    procedure
TBKeluarClick(Sender:
TObject);
    procedure
BtnPilihMakananClick(Sender:
TObject);
```

```
    procedure
BtnPilihBahanClick(Sender:
TObject);
    procedure
DBGridDataMakananCellClick(Col
umn: TColumn);
    procedure
TBTambahClick(Sender:
TObject);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
TBBatalClick(Sender: TObject);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
CBFieldCariClick(Sender:
TObject);
    procedure
CBFieldCariKeyPress(Sender:
TObject; var Key: Char);
    procedure
EdKataKunciKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure FormShow(Sender:
TObject);
    procedure
DBGridDetailBahanCellClick(Col
umn: TColumn);
    procedure
DBControlClick(Sender:
TObject;               Button:
TNavigateBtn);
    procedure
DBControlDetailClick(Sender:
TObject;               Button:
TNavigateBtn);
  private
    { Private declarations }
    procedure
SetPropertyGridMaster();
    procedure
SetPropertyGridDetail();
    procedure
ClearFieldData();
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewMaster();
    procedure ViewDetail();
    procedure RefreshData();
    procedure
RefreshDataDetail();
    procedure
SetStatusBar(AModeData       :
TModeData);
    function GetIdFromEd(AEd :
TEdit) : String;
    function
CekValiditasInput() : Boolean;
  public
    { Public declarations }
  end;

var
  FDataDetailBahanMakanan:
TFDataDetailBahanMakanan;
  vQueryMaster : TADOQuery;
  vQueryDetail : TADOQuery;
  vModeData : TModeData;


implementation

uses UDataMakanan, UDataBahan,
UPermintaanMakanan,
ULoginUser;

{$R *.dfm}

function
TFDataDetailBahanMakanan.CekVa
liditasInput: Boolean;
begin
  If  (Trim(EdIdBahan.Text)  =
'') or (Trim(EdIdMakanan.Text)
= '') Then
    Result := False
  Else
    Result := True;
```

```
end;

procedure
TFDataDetailBahanMakanan.Clear
FieldData;
begin
  EdIdDetail.Clear;
  EdIdBahan.Clear;
  EdIdMakanan.Clear;
  EdJumlahBahan.Text := '0';
  EdKataKunci.Clear;
end;

function
TFDataDetailBahanMakanan.GetId
FromEd(AEd: TEdit): String;
var
  vPos : integer;
begin
  if AEd.Text <> '' then
    begin
      vPos      :=      Pos('-
',AEd.Text);
      Result                :=
LeftStr(AEd.Text,vPos-1);
    end
  else
    Result := '0';
end;

procedure
TFDataDetailBahanMakanan.Refre
shData;
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    vQueryMaster            :=
vData.GetDetailBahanMakanan('M
aster',0);
    DSDataMaster.DataSet    :=
vQueryMaster;

DBGridDataMakanan.DataSource
:= DSDataMaster;
    DBControl.DataSource    :=
DSDataMaster;
    //FIRST LOAD - CHECK THE
STATUS DATA IF EMPTY OR ANY
DATA IS IN

//==========================
==========================
=====
    If
(vQueryMaster.RecordCount = 0)
Then
      begin
        ShowMessage('Data
Detail Bahan Makanan Kosong,
Anda      Harus      Melakukan
Penambahan Data !');
        ClearFieldData;
        SetKontrol(True);
        TBTambahClick(Nil);
        TBBatal.Enabled     :=
False;
      end
    else
      begin
        SetKontrol(False);
        vModeData           :=
BrowseData;
        ClearFieldData;
        ViewMaster;
        RefreshDataDetail;
        ViewDetail;
      end;

//==========================
==========================
=====

    SetStatusBar(vModeData);
    SetPropertyGridMaster;
    SetPropertyGridDetail;
  finally
    FreeAndNil(vData);
  end;
end;
```

```
procedure
TFDataDetailBahanMakanan.SetKo
ntrol(ABool: Boolean);
begin
  PanelMakanan.Enabled  :=  Not
ABool;
  PanelDetail.Enabled       :=
ABool;
  PanelDBGrid.Enabled  :=  Not
ABool;
  PanelCari.Enabled   :=   Not
ABool;
  TBTambah.Enabled    :=    Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBRefresh.Enabled   :=   Not
ABool;
  BtnPilihBahan.Enabled     :=
ABool;
  BtnPilihMakanan.Enabled   :=
ABool;
  If   FLoginUser.vHakAkses   =
'Staff' Then
    begin
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := False;
      TBHapus.Enabled       :=
False;
      TBSimpan.Enabled      :=
False;
      TBBatal.Enabled       :=
False;
    end;
end;

procedure
TFDataDetailBahanMakanan.SetPr
opertyGridDetail;
var
  vCount : Integer;
begin
  With DBGridDetailBahan do
    begin
      for vCount :=0 to 2 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
      Columns[0].Width := 80;
      Columns[1].Width := 80;
      Columns[2].Width := 210;
      Columns[3].Visible    :=
False;
      Columns[4].Visible    :=
False;
      Columns[0].Alignment  :=
taCenter;
      Columns[1].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;

procedure
TFDataDetailBahanMakanan.SetSt
atusBar(AModeData: TModeData);
begin
  If (AModeData = InsertData)
Then
    StatusBar.Panels[0].Text
:=  'Insert Mode - Penambahan
Data'
  Else    If    (AModeData    =
UpdateData) Then
    StatusBar.Panels[0].Text
:= 'Update Mode - Pengubahan
Data'
  Else    If    (AModeData    =
BrowseData) Then
    StatusBar.Panels[0].Text
:= 'Browse Mode - Penelusuran
Data'
  Else    If    (AModeData    =
DeleteData) Then
```

```
    StatusBar.Panels[0].Text
:= 'Delete Mode - Penghapusan
Data'
  Else    If    (AModeData    =
SearchData) Then
    StatusBar.Panels[0].Text
:= 'Search Mode - Pencarian
Data'
  Else
    StatusBar.Panels[0].Text
:= 'Unknown Mode';
  StatusBar.Panels[1].Text  :=
'Data     Detail     Ke-'     +
IntToStr(vQueryDetail.RecNo) +
'        Dari        '        +
IntToStr(vQueryDetail.RecordCo
unt) + ' Data';
end;

procedure
TFDataDetailBahanMakanan.TBKel
uarClick(Sender: TObject);
begin
  Close;
end;

procedure
TFDataDetailBahanMakanan.ViewD
etail;
begin
  If(vQueryDetail.RecordCount
<> 0) then
    begin
      EdIdDetail.Text       :=
DBGridDetailBahan.Fields[0].Te
xt;
      EdIdBahan.Text        :=
DBGridDetailBahan.Fields[1].Te
xt        +        '-'        +
DBGridDetailBahan.Fields[2].Te
xt;
    end
  Else
    begin
      EdIdDetail.Text := '';
      EdIdBahan.Text := '';
    end;
end;

procedure
TFDataDetailBahanMakanan.BtnPi
lihMakananClick(Sender:
TObject);
begin

FDataMakanan.SetPilihMakanan(T
Form(FDataDetailBahanMakanan),
True);
  FDataMakanan.ShowModal;
end;

procedure
TFDataDetailBahanMakanan.BtnPi
lihBahanClick(Sender:
TObject);
begin

FDataBahan.SetPilihBahan(True)
;
  FDataBahan.ShowModal;
end;

procedure
TFDataDetailBahanMakanan.DBGri
dDataMakananCellClick(
  Column: TColumn);
begin
  ViewMaster;
  RefreshDataDetail;
  ViewDetail;
  SetStatusBar(vModeData);
end;

procedure
TFDataDetailBahanMakanan.TBTam
bahClick(Sender: TObject);
begin
  ClearFieldData;
  vModeData := InsertData;
  SetKontrol(True);
  EdIdDetail.Text := 'Auto';
  EdIdMakanan.SetFocus;
  SetStatusBar(vModeData);
end;
```

```
procedure
TFDataDetailBahanMakanan.TBUbahClick(Sender: TObject);
begin
  If (vModeData = BrowseData) Then
    vModeData := UpdateData;
  SetKontrol(True);
  SetStatusBar(vModeData);
end;

procedure
TFDataDetailBahanMakanan.TBHapusClick(Sender: TObject);
var
  vData : TPermintaanMakanan;
begin
  vModeData := DeleteData;
  SetStatusBar(vModeData);
  If  MessageDlg('Anda  Yakin Menghapus  Data  Detai  Bahan-Makanan Dengan ID. Detail : ' +
      EdIdMakanan.Text + ' ?' ,   mtConfirmation,   [mbYes, mbNo],0) = mrYes Then
    begin
      try
        vData               := TPermintaanMakanan.Create;
        If vData.DeleteDetailBahanMakanan (StrToInt(EdIdDetail.Text)) Then
          ShowMessage('Data Detail  Bahan-Makanan  Berhasil Dihapus !')
        Else
          ShowMessage('Data Detail   Bahan-Makanan   Gagal Dihapus !');
        RefreshData;
      finally
        FreeAndNil(vData);
      end;
    end
  else
    begin
      vModeData := BrowseData;
      SetStatusBar(vModeData);
    end;
end;

procedure
TFDataDetailBahanMakanan.TBSimpanClick(Sender: TObject);
var
  vData : TPermintaanMakanan;
begin
  If   not   CekValiditasInput Then
    begin
      ShowMessage('Terjadi Kesalahan  Dalam  Penginputan Data, Mohon Diperbaiki');
      Exit;
    end;
  If (vModeData = InsertData) Then
    begin
      If(MessageDlg('Apakah Anda  Yakin  Ingin  Menambahkan Data
Ini?',mtConfirmation,[mbYes,mbNo],0)= mrYes) Then
        begin
          try
            vData           := TPermintaanMakanan.Create;
            If (vData.CreateDetailBahanMakanan(

StrToInt(GetIdFromEd(EdIdBahan)),

StrToInt(GetIdFromEd(EdIdMakanan)))) Then

ShowMessage('Data       Detail Bahan-Makanan         Berhasil Ditambahkan !')
            Else

ShowMessage('Data       Detail
```

```
Bahan-Makanan            Gagal
Ditambahkan !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end
  Else   If   (vModeData   =
UpdateData)  or  (vModeData  =
SearchData) Then
    begin
      If(MessageDlg('Apakah
Anda Yakin Ingin Mengubah Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData          :=
TPermintaanMakanan.Create;
            If
(vData.UpdateDetailBahanMakana
n(

StrToInt(EdIdDetail.Text),

StrToInt(GetIdFromEd(EdIdBahan
)),

StrToInt(GetIdFromEd(EdIdMakan
an)))) Then

ShowMessage('Data       Detail
Bahan-Makanan  Berhasil  Diubah
!')
            Else

ShowMessage('Data       Detail
Bahan-Makanan   Gagal   Diubah
!');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end;
  RefreshData;
end;

procedure
TFDataDetailBahanMakanan.TBBat
alClick(Sender: TObject);
begin
  If (vModeData = InsertData)
or (vModeData =  UpdateData)
Then
    begin
      vModeData := BrowseData;
      ViewMaster;
      RefreshDataDetail;
      ViewDetail;
      SetKontrol(False);
      SetStatusBar(vModeData);
    end
  Else     If(vModeData     =
SearchData) Then
    begin
      ViewMaster;
      RefreshDataDetail;
      ViewDetail;
      SetKontrol(False);
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := True;
      TBHapus.Enabled := True;
    end;
end;

procedure
TFDataDetailBahanMakanan.TBRef
reshClick(Sender: TObject);
begin
  RefreshData;
end;

procedure
TFDataDetailBahanMakanan.CBFie
ldCariClick(Sender: TObject);
begin
  EdKataKunci.Clear;
  EdKataKunci.SetFocus;
end;
```

```
procedure
TFDataDetailBahanMakanan.CBFie
ldCariKeyPress(Sender:
TObject;
  var Key: Char);
begin
  Key := #0;
end;

procedure
TFDataDetailBahanMakanan.EdKat
aKunciKeyUp(Sender: TObject;
  var   Key:   Word;   Shift:
TShiftState);
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    If (Trim(CBFieldCari.Text)
= '') Then
      begin
        FreeAndNil(vData);
        Exit;
      end
    Else
If(Trim(CBFieldCari.Text)    =
'Nama Makanan')Then
      begin
        vQueryMaster        :=
vData.GetDataSearchDetailBahan
Makanan(CBFieldCari.Text,
EdKataKunci.Text,0);
        DSDataMaster.DataSet
:= vQueryMaster;

DBGridDataMakanan.DataSource
:= DSDataMaster;
        DBControl.DataSource
:= DSDataMaster;
        SetPropertyGridMaster;
        ViewMaster;
        RefreshDataDetail;
        ViewDetail;
      end
    Else
If(Trim(CBFieldCari.Text)    =
'Nama Bahan') Then
      begin
        vQueryDetail        :=
vData.GetDataSearchDetailBahan
Makanan(CBFieldCari.Text,
EdKataKunci.Text,StrToInt(GetI
dFromEd(EdIdMakanan)));
        DSDataDetail.DataSet
:= vQueryDetail;

DBGridDetailBahan.DataSource
:= DSDataDetail;

DBControlDetail.DataSource  :=
DSDataDetail;
        ViewDetail;
      end;
    vModeData := SearchData;
    SetKontrol(False);
    SetStatusBar(vModeData);
    TBTambah.Enabled := False;
    If
(vQueryMaster.RecordCount = 0)
or (vQueryDetail.RecordCount =
0) Then
      begin
        TBUbah.Enabled      :=
False;
        TBHapus.Enabled     :=
False;
      end
    Else
      begin
        TBUbah.Enabled      :=
True;
        TBHapus.Enabled     :=
True;
        If
FLoginUser.vHakAkses = 'Staff'
Then
          begin
            TBUbah.Enabled  :=
False;
            TBHapus.Enabled :=
False;
          end;
```

```
      end;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataDetailBahanMakanan.FormS
how(Sender: TObject);
begin
  RefreshData;
end;

procedure
TFDataDetailBahanMakanan.ViewM
aster;
begin
  If(vQueryMaster.RecordCount
<> 0) then
    EdIdMakanan.Text        :=
DBGridDataMakanan.Fields[0].Te
xt       +       '-'       +
DBGridDataMakanan.Fields[1].Te
xt
  else
    EdIdMakanan.Text := '';
end;

procedure
TFDataDetailBahanMakanan.Refre
shDataDetail;
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
    vQueryDetail            :=
vData.GetDetailBahanMakanan('D
etail',
StrToInt(GetIdFromEd(EdIdMakan
an)));
    DSDataDetail.DataSet    :=
vQueryDetail;

DBGridDetailBahan.DataSource
:= DSDataDetail;
    DBControlDetail.DataSource
:= DSDataDetail;
    SetPropertyGridDetail;
  finally
    FreeAndNil(vData);
  end;
end;


procedure
TFDataDetailBahanMakanan.SetPr
opertyGridMaster;
var
  vCount : Integer;
begin
  With DBGridDataMakanan do
    begin
      for vCount :=0 to 2 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
      Columns[0].Width := 80;
      Columns[1].Width := 150;
      Columns[2].Width := 150;
      Columns[0].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;

procedure
TFDataDetailBahanMakanan.DBGri
dDetailBahanCellClick(
  Column: TColumn);
begin
  ViewDetail;
  SetStatusBar(vModeData);
end;
```

```
procedure
TFDataDetailBahanMakanan.DBCon
trolClick(Sender: TObject;
  Button: TNavigateBtn);
begin
  ViewMaster;
  RefreshDataDetail;
  ViewDetail;
  SetStatusBar(vModeData);
end;

procedure
TFDataDetailBahanMakanan.DBCon
trolDetailClick(Sender:
TObject;
  Button: TNavigateBtn);
begin
  ViewDetail;
  SetStatusBar(vModeData);
end;

end.

unit UDataBahan;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs,    ComCtrls,    DB,
ImgList,    XPMan,    ToolWin,
StdCtrls, DBCtrls, Grids,
  DBGrids, ExtCtrls, ADODB;

type
  TModeData   =   (BrowseData,
UpdateData,        InsertData,
DeleteData, SearchData);
type
  TFDataBahan = class(TForm)
    PanelDBGrid: TPanel;
    DBGridDataBahan: TDBGrid;
    DBControl: TDBNavigator;
    PanelCari: TPanel;
    LblFieldCari: TLabel;
    LblKataKunci: TLabel;
    CBFieldCari: TComboBox;
    EdKataKunci: TEdit;
    PanelDetail: TPanel;
    LblIdBahan: TLabel;
    LblNamaBahan: TLabel;
    EdIdBahan: TEdit;
    EdNamaBahan: TEdit;
    ToolBar: TToolBar;
    TBTambah: TToolButton;
    TBUbah: TToolButton;
    TBHapus: TToolButton;
    TBSeparator1: TToolButton;
    TBSimpan: TToolButton;
    TBBatal: TToolButton;
    TBSeparator2: TToolButton;
    TBRefresh: TToolButton;
    TBSeparator3: TToolButton;
    TBKeluar: TToolButton;
    XPStyle: TXPManifest;
    ImageList: TImageList;
    DSDataBahan: TDataSource;
    StatusBar: TStatusBar;
    LblStok: TLabel;
    EdStok: TEdit;
    LblGram: TLabel;
    TBPilih: TToolButton;
    procedure
TBKeluarClick(Sender:
TObject);
    procedure
DBGridDataBahanCellClick(Colum
n: TColumn);
    procedure
DBControlClick(Sender:
TObject;               Button:
TNavigateBtn);
    procedure
TBRefreshClick(Sender:
TObject);
    procedure
TBTambahClick(Sender:
TObject);
    procedure
TBUbahClick(Sender: TObject);
    procedure
TBHapusClick(Sender: TObject);
```

```
    procedure
TBSimpanClick(Sender:
TObject);
    procedure
TBBatalClick(Sender: TObject);
    procedure
CBFieldCariClick(Sender:
TObject);
    procedure
CBFieldCariKeyPress(Sender:
TObject; var Key: Char);
    procedure
EdKataKunciKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure FormShow(Sender:
TObject);
    procedure
TBPilihClick(Sender: TObject);
  private
    { Private declarations }
    procedure
SetPropertyGridBahan();
    procedure
ClearFieldData();
    procedure SetKontrol(ABool
: Boolean);
    procedure ViewDetail();
    procedure RefreshData();
    procedure
SetStatusBar(AModeData       :
TModeData);
    function
CekValiditasInput() : Boolean;
  public
    { Public declarations }
    procedure
SetPilihBahan(APilihBahan    :
Boolean);
  end;

var
  FDataBahan: TFDataBahan;
  vQuery : TADOQuery;
  vModeData : TModeData;
  vPilihBahan : Boolean;

implementation

uses       UPermintaanMakanan,
UDataDetaiBahanMakanan,
ULoginUser;


{$R *.dfm}

{ TFDataBahan }

procedure
TFDataBahan.ClearFieldData;
begin
  EdIdBahan.Clear;
  EdNamaBahan.Clear;
  EdStok.Text := '0';
  EdKataKunci.Clear;
end;

procedure
TFDataBahan.RefreshData;
var
  vData : TPermintaanMakanan;
begin
  try
    vData                    :=
TPermintaanMakanan.Create;
    vQuery                   :=
vData.GetBahanMakanan;
    DSDataBahan.DataSet      :=
vQuery;
    DBGridDataBahan.DataSource
:= DSDataBahan;
    DBControl.DataSource     :=
DSDataBahan;
    //FIRST LOAD - CHECK THE
STATUS DATA IF EMPTY OR ANY
DATA IS IN

//============================
============================
=====
    If  (vQuery.RecordCount  =
0) Then
      begin
        ShowMessage('Data
Bahan  Makanan  Kosong,  Anda
```

```
Harus   Melakukan   Penambahan
Data !');
        ClearFieldData;
        SetKontrol(True);
        TBTambahClick(Nil);
        TBBatal.Enabled     :=
False;
      end
    else
      begin
        SetKontrol(False);
        vModeData           :=
BrowseData;
        ClearFieldData;
        ViewDetail;
      end;

//==========================
============================
=====
    SetStatusBar(vModeData);
    SetPropertyGridBahan;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDataBahan.SetKontrol(ABool:
Boolean);
begin
  PanelDetail.Enabled       :=
ABool;
  PanelDBGrid.Enabled  :=  Not
ABool;
  PanelCari.Enabled   :=   Not
ABool;
  TBTambah.Enabled    :=    Not
ABool;
  TBUbah.Enabled := Not ABool;
  TBHapus.Enabled    :=    Not
ABool;
  TBSimpan.Enabled := ABool;
  TBBatal.Enabled := ABool;
  TBRefresh.Enabled   :=   Not
ABool;
  TBPilih.Enabled    :=    Not
ABool;
  If (vPilihBahan = True) Then
    TBPilih.Visible := True
  Else
    TBPilih.Visible := False;
  If   FLoginUser.vHakAkses   =
'Staff' Then
    begin
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := False;
      TBHapus.Enabled       :=
False;
      TBSimpan.Enabled      :=
False;
      TBBatal.Enabled       :=
False;
    end;
end;

procedure
TFDataBahan.SetPropertyGridBah
an;
var
  vCount : Integer;
begin
  With DBGridDataBahan do
    begin
      for vCount :=0 to 1 do
        begin

Columns[vCount].Title.Alignmen
t := taCenter;

Columns[vCount].Title.Font.Sty
le := [fsBold];
        end;
      Columns[0].Width := 75;
      Columns[1].Width := 200;
      Columns[0].Alignment  :=
taCenter;
      Columns[0].Font.Color :=
clWhite;
      Columns[0].Color      :=
clWindowFrame;
    end;
end;
```

```
procedure
TFDataBahan.SetStatusBar(AMode
Data: TModeData);
begin
  If (AModeData = InsertData)
Then
    StatusBar.Panels[0].Text
:=  'Insert Mode - Penambahan
Data'
  Else    If    (AModeData    =
UpdateData) Then
    StatusBar.Panels[0].Text
:= 'Update Mode - Pengubahan
Data'
  Else    If    (AModeData    =
BrowseData) Then
    StatusBar.Panels[0].Text
:= 'Browse Mode - Penelusuran
Data'
  Else    If    (AModeData    =
DeleteData) Then
    StatusBar.Panels[0].Text
:= 'Delete Mode - Penghapusan
Data'
  Else    If    (AModeData    =
SearchData) Then
    StatusBar.Panels[0].Text
:= 'Search Mode - Pencarian
Data'
  Else
    StatusBar.Panels[0].Text
:= 'Unknown Mode';
  StatusBar.Panels[1].Text  :=
'Data     Bahan     Ke-'     +
IntToStr(vQuery.RecNo)   +   '
Dari            '            +
IntToStr(vQuery.RecordCount) +
' Data';
end;

procedure
TFDataBahan.ViewDetail;
begin
  EdIdBahan.Text            :=
DBGridDataBahan.Fields[0].Text
;
  EdNamaBahan.Text          :=
DBGridDataBahan.Fields[1].Text
;
  {
  If (vQuery.RecordCount = 0)
Then
    EdStok.Text := '0'
  Else
    EdStok.Text             :=
DBGridDataBahan.Fields[2].Text
;     }
end;

procedure
TFDataBahan.TBKeluarClick(Send
er: TObject);
begin
  vPilihBahan := False;
  Close;
end;                  procedure
TFDataBahan.DBGridDataBahanCel
lClick(Column: TColumn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;



procedure
TFDataBahan.DBControlClick(Sen
der: TObject;
  Button: TNavigateBtn);
begin
  ViewDetail;
SetStatusBar(vModeData);
end;

procedure
TFDataBahan.TBRefreshClick(Sen
der: TObject);
begin
  RefreshData;
end;

procedure
TFDataBahan.TBTambahClick(Send
er: TObject);
```

```
begin
  ClearFieldData;
  vModeData := InsertData;
  SetKontrol(True);
  EdIdBahan.Text := 'Auto';
  EdNamaBahan.SetFocus;
  SetStatusBar(vModeData);
end;

procedure
TFDataBahan.TBUbahClick(Sender
: TObject);
begin
  If (vModeData = BrowseData)
Then
    vModeData := UpdateData;
  SetKontrol(True);
  SetStatusBar(vModeData);
end;

procedure
TFDataBahan.TBHapusClick(Sende
r: TObject);
var
  vData : TPermintaanMakanan;
begin
  vModeData := DeleteData;
  SetStatusBar(vModeData);
  If  MessageDlg('Anda  Yakin
Menghapus  Data  Bahan  Dengan
ID. Bahan : ' +
      EdIdBahan.Text + ' ?' ,
mtConfirmation,        [mbYes,
mbNo],0) = mrYes Then
    begin
      try
        vData               :=
TPermintaanMakanan.Create;
        If
vData.DeleteBahanMakanan(StrTo
Int(EdIdBahan.Text)) Then
          ShowMessage('Data
Bahan Makanan Berhasil Dihapus
!')
        Else
          ShowMessage('Data
Bahan  Makanan  Gagal  Dihapus
!');
        RefreshData;
      finally
        FreeAndNil(vData);
      end;
    end
  else
    begin
      vModeData := BrowseData;
      SetStatusBar(vModeData);
    end;
end;

procedure
TFDataBahan.TBSimpanClick(Send
er: TObject);
var
  vData : TPermintaanMakanan;
begin
  If   not   CekValiditasInput
Then
    begin
      ShowMessage('Terjadi
Kesalahan  Dalam  Penginputan
Data, Mohon Diperbaiki');
      Exit;
    end;
  If (vModeData = InsertData)
Then
    begin
      If(MessageDlg('Apakah
Anda  Yakin  Ingin  Menambahkan
Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData          :=
TPermintaanMakanan.Create;
            If
(vData.CreateBahanMakanan(

EdNamaBahan.Text)) Then

ShowMessage('Data        Bahan
Berhasil Ditambahkan !')
            Else
```

```
ShowMessage('Data Bahan Gagal
Ditambahkan !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end
  Else   If   (vModeData   =
UpdateData) or (vModeData =
SearchData) Then
    begin
      If(MessageDlg('Apakah
Anda Yakin Ingin Mengubah Data
Ini?',mtConfirmation,[mbYes,mb
No],0)= mrYes) Then
        begin
          try
            vData          :=
TPermintaanMakanan.Create;
            If
(vData.UpdateBahanMakanan(

StrToInt(EdIdBahan.Text),

EdNamaBahan.Text)) Then

ShowMessage('Data        Bahan
Berhasil Diubah !')
            Else

ShowMessage('Data Bahan Gagal
Diubah !');
          finally
            FreeAndNil(vData);
          end;
        end
      Else
        Exit;
    end;
  RefreshData;
end;

procedure
TFDataBahan.TBBatalClick(Sende
r: TObject);
begin
  If (vModeData = InsertData)
or (vModeData =  UpdateData)
Then
    begin
      vModeData := BrowseData;
      ViewDetail;
      SetKontrol(False);
      SetStatusBar(vModeData);
    end
  Else     If(vModeData     =
SearchData) Then
    begin
      ViewDetail;
      SetKontrol(False);
      TBTambah.Enabled      :=
False;
      TBUbah.Enabled := True;
      TBHapus.Enabled := True;
    end;
end;

procedure
TFDataBahan.CBFieldCariClick(S
ender: TObject);
begin
  EdKataKunci.Clear;
  EdKataKunci.SetFocus;
end;

procedure
TFDataBahan.CBFieldCariKeyPres
s(Sender:  TObject;  var  Key:
Char);
begin
  Key := #0;
end;

procedure
TFDataBahan.EdKataKunciKeyUp(S
ender: TObject; var Key: Word;
  Shift: TShiftState);
var
  vData : TPermintaanMakanan;
begin
  try
    vData                   :=
TPermintaanMakanan.Create;
```

```
    If (Trim(CBFieldCari.Text)
= '') Then
      Exit;
    vQuery                   :=
vData.GetDataSearchBahanMakana
n(CBFieldCari.Text,
EdKataKunci.Text);
    DSDataBahan.DataSet      :=
vQuery;
    DBGridDataBahan.DataSource
:= DSDataBahan;
    DBControl.DataSource     :=
DSDataBahan;
    vModeData := SearchData;
    SetKontrol(False);
    ViewDetail;
    SetStatusBar(vModeData);
    SetPropertyGridBahan;
    TBTambah.Enabled := False;
    If  (vQuery.RecordCount  =
0) Then
      begin
        TBUbah.Enabled       :=
False;
        TBHapus.Enabled      :=
False;
      end
    Else
      begin
        TBUbah.Enabled       :=
True;
        TBHapus.Enabled      :=
True;
        If
FLoginUser.vHakAkses = 'Staff'
Then
          begin
            TBUbah.Enabled   :=
False;
            TBHapus.Enabled :=
False;
          end;
      end;
  finally
    FreeAndNil(vData);
  end;
end;
```

```
procedure
TFDataBahan.SetPilihBahan(APil
ihBahan: Boolean);
begin
  vPilihBahan := APilihBahan;
end;

procedure
TFDataBahan.FormShow(Sender:
TObject);
begin
  RefreshData;
end;

procedure
TFDataBahan.TBPilihClick(Sende
r: TObject);
begin

FDataDetailBahanMakanan.EdIdBa
han.Text  :=  EdIdBahan.Text  +
'-' + EdNamaBahan.Text;
  vPilihBahan := False;
  Close;
end;

function
TFDataBahan.CekValiditasInput:
Boolean;
begin
  If(Trim(EdIdBahan.Text)    =
'') or (Trim(EdNamaBahan.Text)
= '') Then
    Result := False
  Else
    Result := True;
end;

end.

unit UDaftarUser;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
```

```
  Dialogs, ExtCtrls, StdCtrls,
Buttons, ADODB, StrUtils;

type
  TFDaftarUser = class(TForm)
    PanelDaftar: TPanel;
    LblUserName: TLabel;
    LblPassword: TLabel;
    EdUserName: TEdit;
    EdPassword1: TEdit;
    LblStatusUserName: TLabel;
    LblStatusPassword: TLabel;
    EdPassword2: TEdit;
    BtnDaftar: TBitBtn;
    BtnBatal: TBitBtn;
    procedure
BtnDaftarClick(Sender:
TObject);
    procedure
BtnBatalClick(Sender:
TObject);
    procedure
EdUserNameKeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure
EdPassword2KeyUp(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
    function
CekValidasiInput(AUserValid,
APasswordValid  :  Boolean)  :
Boolean;
  public
    { Public declarations }
  end;

var
  FDaftarUser: TFDaftarUser;
  vUserValid, vPasswordValid :
Boolean;

implementation

uses UPegawai, UDataPegawai;

{$R *.dfm}

procedure
TFDaftarUser.BtnDaftarClick(Se
nder: TObject);
var
  vData : TPegawai;
begin
  try
    vData := TPegawai.Create;
    If
(CekValidasiInput(vUserValid,
vPasswordValid) = False) Then
      begin
        ShowMessage('Terjadi
Kesalahan  Dalam  Penginputan
Data !');
        FreeAndNil(vData);
        Exit;
      end;
    If MessageDlg('Apakah Anda
Telah  Memasukkan  Data  Dengan
Benar                    ?',
mtConfirmation,[mbYes,mbNo],0)
= mrYes Then
      If (vData.CreateUser(

FDataPegawai.EdNIP.Text,

Trim(EdUserName.Text),

Trim(EdPassword1.Text),'Staff'
)) Then
          begin

ShowMessage('Pendaftaran
Berhasil !');

FDataPegawai.RefreshData;
            Close;
          end
      Else

ShowMessage('Pendaftaran Gagal
!');
  finally
```

```
    FreeAndNil(vData);
  end;
end;

function
TFDaftarUser.CekValidasiInput(
AUserValid, APasswordValid :
Boolean) : Boolean;
begin
  If (AUserValid = True) and
(APasswordValid = True) Then
    Result := True
  Else
    Result := False;
end;

procedure
TFDaftarUser.BtnBatalClick(Sen
der: TObject);
begin
  Close;
end;

procedure
TFDaftarUser.EdUserNameKeyUp(S
ender: TObject; var Key: Word;
  Shift: TShiftState);
var
  vData : TPegawai;
  vQueryTemp : TADOQuery;
begin
  try
    vData := TPegawai.Create;
    If
(Length(Trim(EdUserName.Text))
=0) Then
      begin

LblStatusUserName.Caption   :=
'Masukkan UserName Anda' ;
        vUserValid := False;

LblStatusUserName.Font.Color
:= clBlue;
        Exit;
      end;

    vQueryTemp             :=
vData.GetDataFinalSearchUser('
username', EdUserName.Text);
    If (vQueryTemp.RecordCount
>0) Then
      begin

LblStatusUserName.Caption   :=
'Username Sudah Terpakai !';

LblStatusUserName.Font.Color
:= clRed;
        vUserValid := False;
      end
    Else
      begin

LblStatusUserName.Caption   :=
'Username Tersedia';

LblStatusUserName.Font.Color
:= clGreen;
        vUserValid := True;
      end;
  finally
    FreeAndNil(vData);
  end;
end;

procedure
TFDaftarUser.EdPassword2KeyUp(
Sender:   TObject;   var   Key:
Word;
  Shift: TShiftState);
begin
  vPasswordValid := False;
  If  Length(EdPassword2.Text)
= 0 Then
    begin

LblStatusPassword.Caption   :=
'Masukkan  Password  Anda  1x
Lagi :' ;

LblStatusPassword.Font.Color
:= clBlue;
      vPasswordValid := False;
      Exit;
```

```
    end;
  If    (EdPassword2.Text    =
EdPassword1.Text) Then
    begin

LblStatusPassword.Caption   :=
'Password Sesuai' ;

LblStatusPassword.Font.Color
:= clGreen;
      vPasswordValid := True;
    end
  Else
    begin

LblStatusPassword.Caption   :=
'Password Tidak Sama' ;

LblStatusPassword.Font.Color
:= clRed;
      vPasswordValid := False;
    end;
end;

procedure
TFDaftarUser.FormShow(Sender:
TObject);
begin
  EdUserName.Clear;
  EdPassword1.Clear;
  EdPassword2.Clear;
  LblStatusUserName.Caption :=
'Masukkan UserName Anda';
  LblStatusPassword.Caption :=
'Masukkan  Password  Anda  1x
Lagi :';
  LblStatusPassword.Font.Color
:= clBlack;
  LblStatusUserName.Font.Color
:= clBlack;
  vUserValid := False;
  vPasswordValid := False;
end;

end.
```

```
program PrjSIGRS;

uses
  Forms,
  UIRI in 'Classes\UIRI.pas',
  UPegawai                  in
'Classes\UPegawai.pas',
  UPermintaanMakanan        in
'Classes\UPermintaanMakanan.pa
s',
  USys in 'Classes\USys.pas',
  UDataPasien               in
'Forms\UDataPasien.pas'
{FDataPasien},
  UDataRuang                in
'Forms\UDataRuang.pas'
{FDataRuang},
  UDataKelas                in
'Forms\UDataKelas.pas'
{FDataKelas},
  UDataIRI                  in
'Forms\UDataIRI.pas'
{FDataIRI},
  UDataBahan                in
'Forms\UDataBahan.pas'
{FDataBahan},
  UDataJenisMakanan         in
'Forms\UDataJenisMakanan.pas'
{FDataJenisMakanan},
  UDataPegawai              in
'Forms\UDataPegawai.pas'
{FDataPegawai},
  UDataMakanan              in
'Forms\UDataMakanan.pas'
{FDataMakanan},
  UDataDetaiBahanMakanan    in
'Forms\UDataDetaiBahanMakanan.
pas'
{FDataDetailBahanMakanan},
  UDaftarUser               in
'Forms\UDaftarUser.pas'
{FDaftarUser},
  UDataPermintaanMakanan    in
'Forms\UDataPermintaanMakanan.
pas' {FDataPermintaanMakanan},
  UMenuUtama                in
'Forms\UMenuUtama.pas'
{FUtama},
```

```
  UDataUser                in
'Forms\UDataUser.pas'
{FDataUser},
  ULoginUser               in
'Forms\ULoginUser.pas'
{FLoginUser},
  UProfileUser             in
'Forms\UProfileUser.pas'
{FProfileUser},
  UGantiUserName           in
'Forms\UGantiUserName.pas'
{FGantiUserName},
  UGantiPassword           in
'Forms\UGantiPassword.pas'
{FGantiPassword},
  FLapPasien               in
'Report\FLapPasien.pas'
{FLaporanPasien},
  UPasienReport            in
'Report\UPasienReport.pas'
{ReportPasien: TQuickRep},
  UDMReport                in
'Report\UDMReport.pas'
{DataModuleReport:
TDataModule},
  UIRIReport               in
'Report\UIRIReport.pas'
{ReportIRI: TQuickRep},
  UPermintaanMakananReport  in
'Report\UPermintaanMakananRepo
rt.pas'
{ReportPermintaanMakanan:
TQuickRep};

{$R *.res}

begin
  Application.Initialize;

Application.CreateForm(TFUtama
, FUtama);

Application.CreateForm(TFDataP
asien, FDataPasien);

Application.CreateForm(TFDataR
uang, FDataRuang);
Application.CreateForm(TFDataK
elas, FDataKelas);

Application.CreateForm(TFDataI
RI, FDataIRI);

Application.CreateForm(TFDataB
ahan, FDataBahan);

Application.CreateForm(TFDataJ
enisMakanan,
FDataJenisMakanan);

Application.CreateForm(TFDataP
egawai, FDataPegawai);

Application.CreateForm(TFDataM
akanan, FDataMakanan);

Application.CreateForm(TFDataD
etailBahanMakanan,
FDataDetailBahanMakanan);

Application.CreateForm(TFDafta
rUser, FDaftarUser);

Application.CreateForm(TFDataP
ermintaanMakanan,
FDataPermintaanMakanan);

Application.CreateForm(TFDataU
ser, FDataUser);

Application.CreateForm(TFLogin
User, FLoginUser);

Application.CreateForm(TFProfi
leUser, FProfileUser);

Application.CreateForm(TFGanti
UserName, FGantiUserName);

Application.CreateForm(TFGanti
Password, FGantiPassword);

Application.CreateForm(TFLapor
anPasien, FLaporanPasien);
```

```
Application.CreateForm(TReport
Pasien, ReportPasien);

Application.CreateForm(TDataMo
duleReport, DataModuleReport);

Application.CreateForm(TReport
IRI, ReportIRI);

Application.CreateForm(TReport
PermintaanMakanan,
ReportPermintaanMakanan);
  Application.Run;
end.

unit FLapPasien;

interface

uses
  Windows, Messages, SysUtils,
Variants,  Classes,  Graphics,
Controls, Forms,
  Dialogs,  ExtCtrls,  QRPrntr,
Spin, Buttons, StdCtrls;

type
  TFLaporanPasien           =
class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    Edit1: TEdit;
    Label2: TLabel;
    Edit2: TEdit;
    SpeedButton1:
TSpeedButton;
    SpeedButton2:
TSpeedButton;
    SpeedButton3:
TSpeedButton;
    SpeedButton4:
TSpeedButton;
    SpeedButton5:
TSpeedButton;
    SpinEdit1: TSpinEdit;
    SpeedButton6:
TSpeedButton;
    SpeedButton7:
TSpeedButton;
    QRPreview1: TQRPreview;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FLaporanPasien:
TFLaporanPasien;

implementation

{$R *.dfm}

end.
```

LAMPIRAN B

HASIL KUISIONER

Kuesioner Sistem Informasi Instalasi Gizi

Di RSUP dr.Hasan Sadikin Bandung

Nama : Sunya

Jabatan : Staff

1. Menurut anda, apakah tampilan aplikasi ini bagus ?

   ☐ Sangat Bagus ☒ Bagus ☐ Cukup Bagus
   ☐ Biasa-biasa Saja ☐ Kurang Bagus
   ☐ Tidak Bagus ☐ Sangat Tidak Bagus

2. Menurut anda, apakah bahasa yang digunakan mudah dimengerti ?

   ☒ Sangat Mudah ☐ Mudah ☐ Cukup Mudah
   ☐ Biasa-biasa Saja ☐ Kurang Mudah
   ☐ Tidak Mudah ☐ Sangat Tidak Mudah

3. Menurut anda, apakah aplikasi ini dapat membantu mengolah data permintaan makanan bagi pasien di Instalasi Gizi RSHS dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☒ Membantu ☐ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

4. Menurut Anda, apakah aplikasi ini dapat membantu memberikan informasi yang berguna bagi Instalasi Gizi RSUP dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☒ Membantu ☐ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

5. Menurut Anda, apakah aplikasi ini dapat membantu memberikan gambaran pengambilan keputusan bagi Instalasi Gizi RSUP dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☐ Membantu ☒ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

6. Halaman aplikasi sudah terstruktur dengan baik?

   ☐ Sangat Baik ☐ Ragu-ragu ☐ Sangat tidak Baik
   ☒ Baik ☐ Tidak Baik

Kuesioner Sistem Informasi Instalasi Gizi

Di RSUP dr.Hasan Sadikin Bandung

Nama : Ida

Jabatan : Administrasi

1. Menurut anda, apakah tampilan aplikasi ini bagus ?

   ☐ Sangat Bagus ☑ Bagus ☐ Cukup Bagus
   ☐ Biasa-biasa Saja ☐ Kurang Bagus
   ☐ Tidak Bagus ☐ Sangat Tidak Bagus

2. Menurut anda, apakah bahasa yang digunakan mudah dimengerti ?

   ☐ Sangat Mudah ☑ Mudah ☐ Cukup Mudah
   ☐ Biasa-biasa Saja ☐ Kurang Mudah
   ☐ Tidak Mudah ☐ Sangat Tidak Mudah

3. Menurut anda, apakah aplikasi ini dapat membantu mengolah data permintaan makanan bagi pasien di Instalasi Gizi RSHS dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☑ Membantu ☐ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

4. Menurut Anda, apakah aplikasi ini dapat membantu memberikan informasi yang berguna bagi Instalasi Gizi RSUP dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☑ Membantu ☐ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

5. Menurut Anda, apakah aplikasi ini dapat membantu memberikan gambaran pengambilan keputusan bagi Instalasi Gizi RSUP dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☐ Membantu ☑ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

6. Halaman aplikasi sudah terstruktur dengan baik?

   ☐ Sangat Baik ☐ Ragu-ragu ☐ Sangat tidak Baik
   ☑ Baik ☐ Tidak Baik

Kuesioner Sistem Informasi Instalasi Gizi

Di RSUP dr.Hasan Sadikin Bandung

Nama : Yeni H.

Jabatan : Staff

1. Menurut anda, apakah tampilan aplikasi ini bagus ?

   ☐ Sangat Bagus ☑ Bagus ☐ Cukup Bagus
   ☐ Biasa-biasa Saja ☐ Kurang Bagus
   ☐ Tidak Bagus ☐ Sangat Tidak Bagus

2. Menurut anda, apakah bahasa yang digunakan mudah dimengerti ?

   ☐ Sangat Mudah ☑ Mudah ☐ Cukup Mudah
   ☐ Biasa-biasa Saja ☐ Kurang Mudah
   ☐ Tidak Mudah ☐ Sangat Tidak Mudah

3. Menurut anda, apakah aplikasi ini dapat membantu mengolah data permintaan makanan bagi pasien di Instalasi Gizi RSHS dr.Hasan Sadikin Bandung ?

   ☑ Sangat Membantu ☐ Membantu ☐ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

4. Menurut Anda, apakah aplikasi ini dapat membantu memberikan informasi yang berguna bagi Instalasi Gizi RSUP dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☑ Membantu ☐ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

5. Menurut Anda, apakah aplikasi ini dapat membantu memberikan gambaran pengambilan keputusan bagi Instalasi Gizi RSUP dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☐ Membantu ☑ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

6. Halaman aplikasi sudah terstruktur dengan baik?

   ☐ Sangat Baik ☐ Ragu-ragu ☐ Sangat tidak Baik
   ☑ Baik ☐ Tidak Baik

Kuesioner Sistem Informasi Instalasi Gizi

Di RSUP dr.Hasan Sadikin Bandung

Nama : Djoko

Jabatan : Administrasi

1. Menurut anda, apakah tampilan aplikasi ini bagus ?

   ☐ Sangat Bagus ☐ Bagus ☑ Cukup Bagus
   ☐ Biasa-biasa Saja ☐ Kurang Bagus
   ☐ Tidak Bagus ☐ Sangat Tidak Bagus

2. Menurut anda, apakah bahasa yang digunakan mudah dimengerti ?

   ☐ Sangat Mudah ☑ Mudah ☐ Cukup Mudah
   ☐ Biasa-biasa Saja ☐ Kurang Mudah
   ☐ Tidak Mudah ☐ Sangat Tidak Mudah

3. Menurut anda, apakah aplikasi ini dapat membantu mengolah data permintaan makanan bagi pasien di Instalasi Gizi RSHS dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☐ Membantu ☑ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

4. Menurut Anda, apakah aplikasi ini dapat membantu memberikan informasi yang berguna bagi Instalasi Gizi RSUP dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☐ Membantu ☑ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

5. Menurut Anda, apakah aplikasi ini dapat membantu memberikan gambaran pengambilan keputusan bagi Instalasi Gizi RSUP dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☑ Membantu ☐ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

6. Halaman aplikasi sudah terstruktur dengan baik?

   ☐ Sangat Baik ☑ Ragu-ragu ☐ Sangat tidak Baik
   ☐ Baik ☐ Tidak Baik

Kuesioner Sistem Informasi Instalasi Gizi

Di RSUP dr.Hasan Sadikin Bandung

Nama : Indah

Jabatan : Administrasi

1. Menurut anda, apakah tampilan aplikasi ini bagus ?

   ☐ Sangat Bagus ☐ Bagus ☐ Cukup Bagus
   ☑ Biasa-biasa Saja ☐ Kurang Bagus
   ☐ Tidak Bagus ☐ Sangat Tidak Bagus

2. Menurut anda, apakah bahasa yang digunakan mudah dimengerti ?

   ☐ Sangat Mudah ☑ Mudah ☐ Cukup Mudah
   ☐ Biasa-biasa Saja ☐ Kurang Mudah
   ☐ Tidak Mudah ☐ Sangat Tidak Mudah

3. Menurut anda, apakah aplikasi ini dapat membantu mengolah data permintaan makanan bagi pasien di Instalasi Gizi RSHS dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☐ Membantu ☐ Cukup Membantu
   ☑ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

4. Menurut Anda, apakah aplikasi ini dapat membantu memberikan informasi yang berguna bagi Instalasi Gizi RSUP dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☐ Membantu ☑ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

5. Menurut Anda, apakah aplikasi ini dapat membantu memberikan gambaran pengambilan keputusan bagi Instalasi Gizi RSUP dr.Hasan Sadikin Bandung ?

   ☐ Sangat Membantu ☑ Membantu ☐ Cukup Membantu
   ☐ Biasa-biasa Saja ☐ Kurang Membantu
   ☐ Tidak Membantu ☐ Sangat Tdk Membantu

6. Halaman aplikasi sudah terstruktur dengan baik?

   ☐ Sangat Baik ☐ Ragu-ragu ☐ Sangat tidak Baik
   ☑ Baik ☐ Tidak Baik

LAMPIRAN C

SURAT PENELITIAN dan BERKAS DARI PERUSAHAAN

DEPARTEMEN KESEHATAN
DIREKTORAT JENDERAL BINA PELAYANAN MEDIK
RSUP Dr. HASAN SADIKIN BANDUNG
Jalan Pasteur No. 38 , Bandung 40161
Telepon : (022) 2034953 (hunting) Faksimile : (022) 2032216, 2032533
Surat Elektronik : humas@rshs.or.id , perjan_rshs@yahoo.com
SMS hotline : 081220050547

Nomor : LB.02.01/C0 22/ 5794 /V/2010
Hal : Ijin Penelitian

7 Mei 2010

Yang terhormat,
Ketua Jurusan Teknik Informatika
Universitas Komputer Indonesia
Jl. Dipati Ukur No 112
Bandung

Menindak lanjuti surat Saudara nomor : 188/LP– SI / TIF--FTIK/UNIKOM / III /2010 , tanggal 25 Maret, 2010 perihal tersebut di atas dengan ini disampaikan bahwa pada prinsipnya kami dapat memberikan ijin kepada mahasiswa Saudara tersebut di bawah ini :

Nama : Ade Jamaludin
NIM : 10105120

untuk melaksanakan penelitian tentang" Sistem Informasi Instalasi Gizi di RSUP Dr. Hasan Sadikin Bandung."

Kegiatan tersebut dapat dilaksanakan dengan ketentuan sebagai berikut :
1. Tidak mengganggu pelayanan di RSUP Dr. Hasan Sadikin
2. Mematuhi prosedur / ketentuan yang telah ditentukan oleh RSUP Dr. Hasan Sadikin
3. Hasil dari kegiatan digunakan hanya untuk tujuan akademik (tidak dipublikasikan)
4. Menyerahkan laporan hasil kegiatan kepada RSUP Dr. Hasan Sadikin melalui Bagian Pendidikan & Penelitian yang disetujui oleh Instalasi Gizi, Instalasi Sistem Informasi Rumah Sakit, serta diketahui oleh Bagian Pendidikan & Penelitian RSUP Dr. Hasan Sadikin sejumlah 2 (dua) exemplar paling lambat satu bulan setelah selesai pelaksanaan.
5. Kegiatan dilaksanakan tanggal 10 Mei s.d. 10 Juni 2010.
6. Untuk pelaksanaannya dilaksanakan berdasarkan kesepakatan Saudara dengan unit terkait
7. Bersedia mempresentasikan hasil penelitian sesuai kebutuhan RSUP Dr.Hasan Sadikin dengan waktu ditentukan kemudian berdasarkan kesepakatan bersama
8. Membawa pas photo masing – masing satu lembar ukuran 2x 3 (hitam putih/ berwarna)

Untuk memperoleh keterangan lebih lanjut sebelum melaksanakan kegiatan, kami harap peserta yang bersangkutan dapat menghubungi Ka. Bag. Pendidikan & Penelitian melalui Ka. Sub.Bag. Pendidikan & Penelitian Keperawatan Non Medik RSUP Dr. Hasan Sadikin Bandung guna penyelesaian administrasi.

Atas perhatian dan kerjasamanya kami ucapkan terima kasih.

a.n. Direktur Utama
Direktur SDM & Pendidikan

DEPARTEMEN KESEHATAN REPUBLIK INDONESIA RSUP Dr. HASAN SADIKIN BANDUNG

Dr. Agus Hadian Rahim, dr Sp. OT(K) , M. Epid, M.H Kes
NIP. 19600812 198812 1 001

Tembusan :
1. Direktur Utama RSHS (sebagai laporan)
2. Kepala Instalasi Gizi RSHS
3. Kepala Instalasi Sistem Informasi RSHS
4. Kepala Bagian Perbendaharaan & Mobilisasi Dana RSHS
5. Yang Bersangkutan

DEPARTEMEN KESEHATAN
RUMAH SAKIT UMUM PUSAT DR. HASAN SADIKIN
JALAN PASTEUR NO. 38
B A N D U N G

DAFTAR PERMINTAAN MAKANAN

TANGGAL : 18-05-10 JUMLAH PASIEN : ____
RUANGAN : Kemuning 2 Ba/cpk JUMLAH YANG DIMINTA : ____
KAPASITAS : ____

| NO. URUT | NAMA | NO. MED. REC | KELAS PERAWATAN | | | | JENIS MAKANAN | | | |