

## LAMPIRAN

### **FormComponent.pas**

```
unit FormComponent;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ADODB, Vcl.ExtCtrls;
type
  TfrmComponent = class(TForm)
    lblCompany: TLabel;
    lblUserName: TLabel;
    lblOtoritas: TLabel;
    adoTerra: TADOConnection;
    timer: TTimer;
    lblDate: TLabel;
    lblTime: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure timerTimer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmComponent: TfrmComponent;
  waktuSekarang: TDateTime;
implementation
uses GlobalUnit;
{$R *.dfm}
procedure TfrmComponent.FormCreate(Sender: TObject);
begin
  adoTerra.Connected := False;
  adoTerra.Connected := True;
end;
procedure TfrmComponent.timerTimer(Sender: TObject);
begin
  waktuSekarang:=Now;
  lblDate.Caption := FormatDateTime('c',waktuSekarang);
  lblTime.Caption := FormatDateTime('hh:nn:ss.zzz',waktuSekarang);
end;
procedure TfrmComponent.FormClose(Sender: TObject;var Action: TCloseAction);
begin
  adoTerra.Connected := False; end; end.
```

## FormPassword.pas (AdminPool.exe)

```
unit FormPassword;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, DB, ADODB;
type
  TfrmPassword = class(TForm)
    btnOK: TSpeedButton;
    btnCancel: TSpeedButton;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    edtuser: TEdit;
    edtpass: TEdit;
    qryPass: TADOQuery;
    procedure btnOKClick(Sender: TObject);
    procedure btnCancelClick(Sender: TObject);
    procedure edtuserKeyPress(Sender: TObject; var Key: Char);
    procedure edtpassKeyPress(Sender: TObject; var Key: Char);
    procedure FormShow(Sender: TObject);
    function CekUserID : Boolean;
    function CekPassword(User, Password : string) : Boolean;
    procedure BersihkanData;
    function CekValid : Boolean;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmPassword: TfrmPassword;
implementation
uses
  FormComponent, FormDaftarSPK;
var
  UserName, Otoritas, temp : string;
  {$R *.dfm}
procedure TfrmPassword.btnOKClick(Sender: TObject);
begin
  if CekValid then
  begin
    temp := 'UPDATE Tbl_MstSystemUser SET LoginStat = ' + #39 + 'TRUE' + #39 +
      'WHERE IDSystemUser = ' + #39 + edtuser.Text + #39;
    with qryPass do
    begin
```

```

    Close;
    SQL.Clear;
    SQL.Add(temp);
    ExecSQL;
end;
frmComponent.lblUserName.Caption := UserName;
frmComponent.lblOtoritas.Caption := Otoritas;
frmDaftarSPK.ShowModal;
temp := 'UPDATE Tb1_MstSystemUser SET LoginStat = ' + #39 + 'FALSE' + #39 +
        'WHERE IDSystemUser = ' + #39 + edtuser.Text + #39;
with qryPass do
begin
    Close;
    SQL.Clear;
    SQL.Add(temp);
    ExecSQL;
end;
Close;
end;
end;
procedure TfrmPassword.btnCancelClick(Sender: TObject);
begin
    Close;
end;
procedure TfrmPassword.edtuserKeyPress(Sender: TObject; var Key: Char);
begin
    if key = #13 then
        edtpass.SetFocus;
end;
procedure TfrmPassword.edtpassKeyPress(Sender: TObject; var Key: Char);
begin
    if key = #13 then
        btnOK.Click;
end;
procedure TfrmPassword.FormShow(Sender: TObject);
begin
    frmComponent.adoTerra.Connected := True;
    BersihkanData;
end;
function TfrmPassword.CekPassword(User, Password : string) : Boolean;
begin
    temp := 'SELECT * FROM Tb1_MstSystemUser WHERE ' +
            'IDSystemUser = ' + #39 + UserName + #39 + ";
    with qryPass do
    begin
        close;
        sql.Clear;
        sql.Add(temp);

```

```

open;
if not locate('Password',Password,[]) then
begin
  showmessage('Wrong password, enter again !');
  edtPass.Clear;
  edtPass.SetFocus;
  CekPassword := false;
end
else
begin
  CekPassword := true;
  Otoritas := FieldByName('Rules').AsString;
end;
close;
end;
end;
function TfrmPassword.CekUserID;
begin
  temp := 'SELECT * FROM Tbl_MstSystemUser';
  with qryPass do
  begin
    close;
    sql.Clear;
    sql.Add(temp);
    open;
    if not locate('IDSystemUser',edtuser.Text,[]) then
    begin
      showmessage('Wrong or not registered user name, try again !');
      edtUser.Clear;
      edtUser.SetFocus;
      edtPass.Clear;
      CekUserID := false;
    end
    else
      CekUserID := true;
      UserName := fieldbyname('IDSystemUser').AsString;
    close;
  end;
end;
procedure TfrmPassword.BersihkanData;
begin
  edtUser.Clear;
  edtPass.Clear;
  edtUser.SetFocus;
end;
function TfrmPassword.CekValid : Boolean;
begin
  CekValid := true;

```

```

if edtUser.Text = " then
begin
  Showmessage('User name empty !');
  edtUser.Clear;
  CekValid := false;
  exit;
end;
if edtPass.Text = " then
begin
  Showmessage('Password still empty !');
  edtPass.Clear;
  CekValid := false;
  exit;
end;
if Not CekUserID then
begin
  CekValid := false;
  exit;
end;
if Not CekPassword(edtUser.Text, edtPass.Text) then
begin
  CekValid := false;
  exit; end;
temp := 'SELECT LoginStat FROM Tbl_MstSystemUser WHERE IDSystemUser = ' +
#39 + edtUser.Text + #39;
with qryPass do
begin
  close;
  sql.Clear;
  sql.Add(temp);
  open;
  if FieldByName('LoginStat').AsString = 'TRUE' then
  begin
    CekValid := false;
    ShowMessage('Your username already LOGIN');
  end; end; end;
procedure TfrmPassword.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
  qryPass.Close;
  frmComponent.adbTerra.Connected := False;
end;
end.

```

## **FormDaftarSPK.pas**

```
unit FormDaftarSPK;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
  Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Data.DB, Data.Win.ADODB, Vcl.Grids,
  AdvObj, BaseGrid, AdvGrid, DBAdvGrid, Vcl.StdCtrls, Vcl.ComCtrls, Vcl.Buttons,
  Vcl.ExtCtrls;
type
  TfrmDaftarSPK = class(TForm)
    GroupBox1: TGroupBox;
    btnAdd: TSpeedButton;
    btnEditDraft: TSpeedButton;
    btnConfirm: TSpeedButton;
    btnRefresh: TSpeedButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    btnCancel: TSpeedButton;
    dtpAwal: TDateTimePicker;
    cbStatus: TComboBox;
    dtpAkhir: TDateTimePicker;
    GroupBox2: TGroupBox;
    qryPerbaikan: TADOQuery;
    dsPerbaikan: TDataSource;
    GroupBox3: TGroupBox;
    Splitter1: TSplitter;
    grdDtlSPK: TDBAdvGrid;
    qryDtlSPK: TADOQuery;
    dsMsSPK: TDataSource;
    dsDtlSPK: TDataSource;
    qryDtlSPKNoSPK: TWideStringField;
    qryDtlSPKNoUrut: TSmallintField;
    qryDtlSPKItemsCode: TWideStringField;
    qryDtlSPKJmlDiminta: TBCDField;
    qryDtlSPKJmlDipakai: TBCDField;
    qryDtlSPKSelisihPemakaian: TBCDField;
    qryDtlSPKCost: TBCDField;
    qryDtlSPKIDSystemUser: TWideStringField;
    qryDtlSPKUpdateTime: TDateTimeField;
    qryMsSPK: TADOQuery;
    grdSPK: TDBAdvGrid;
    DBAdvGrid1: TDBAdvGrid;
    qryPerbaikanNoMintaPerbaikan: TWideStringField;
    qryPerbaikanTglMintaPerbaikan: TDateTimeField;
    qryPerbaikanDeskripsiPerbaikan: TWideStringField;
    qryPerbaikanDeskripsiSparepart: TWideStringField;
```

```

qryPerbaikanIDSystemUser: TWideStringField;
qryPerbaikanUpdateTime: TDateTimeField;
Splitter2: TSplitter;
qryMsSPKNoSPK: TWideStringField;
qryMsSPKTglSPK: TDateTimeField;
qryMsSPKNoMintaPerbaikan: TWideStringField;
qryMsSPKIDMekanik: TWideStringField;
qryMsSPKKeteranganSPK: TWideStringField;
qryMsSPKIDSystemUser: TWideStringField;
qryMsSPKUpdateTime: TDateTimeField;
qryMsSPKStatusSPK: TWideStringField;
qryMsSPKName: TWideStringField;
qryDtlSPKItemsName: TWideStringField;
qryDtlSPKUOM: TWideStringField;
btnTutupSPK: TSpeedButton;
qryPerbaikanNoPolisi: TWideStringField;
btnAmbil: TSpeedButton;
qryDtlSPKCostPrice: TBCDField;
qryDtlSPKMerek: TWideStringField;
chkEksternal: TCheckBox;
qryMsSPKSupplierCode: TWideStringField;
qryMsSPKHarga: TBCDField;
procedure btnCancelClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure btnRefreshClick(Sender: TObject);
procedure RefreshMsSPK;
procedure RefreshDtlSPK;
procedure DBAdvGrid1Click(Sender: TObject);
procedure cbStatusChange(Sender: TObject);
procedure dtpAwalChange(Sender: TObject);
procedure dtpAkhirChange(Sender: TObject);
procedure btnAddClick(Sender: TObject);
procedure grdSPKClick(Sender: TObject);
procedure btnConfirmClick(Sender: TObject);
procedure grdSPKDb1Click(Sender: TObject);
procedure btnDocumentClick(Sender: TObject);
procedure btnTutupSPKClick(Sender: TObject);
procedure btnEditDraftClick(Sender: TObject);
procedure btnAmbilClick(Sender: TObject);
procedure chkEksternalClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  frmDaftarSPK: TfrmDaftarSPK;
implementation

```

```

Uses FormComponent, FormInputSPK, FormUseItems, FormUseItemsTires,
FormInputSPKEksternal;
var temp:string;
{$R *.dfm}
procedure TfrmDaftarSPK.btnAddClick(Sender: TObject);
begin
  if not qryPerbaikan.IsEmpty then
    begin
      if chkEksternal.Checked then
        begin
          frmInputSPKEksternal.txtStatus.Caption := 'NEW';
          frmInputSPKEksternal.ShowModal;
        end
      else
        begin
          frmInputSPK.txtStatus.Caption := 'NEW';
          frmInputSPK.ShowModal;
        end;
      btnRefresh.Click;
    end;end;
procedure TfrmDaftarSPK.btnAmbilClick(Sender: TObject);
begin
  if not(chkEksternal.Checked) and not (qryDtlSPK.IsEmpty) then
    begin
      if qryDtlSPK.FieldByName('ItemsName').AsString <> 'Ban' then
        begin
          frmUseItems.txtNoSPK.Caption := qryMsSPK.FieldByName('NoSPK').AsString;
          frmUseItems.lblNoUrut.Caption := qryDtlSPK.FieldByName('NoUrut').AsString;
          frmUseItems.lblHarga.Caption := qryDtlSPK.FieldByName('CostPrice').AsString;
          frmUseItems.txtKodeBarang.Caption :=
qryDtlSPK.FieldByName('ItemsCode').AsString;
          frmUseItems.txtNamaBarang.Caption :=
qryDtlSPK.FieldByName('ItemsName').AsString;
          frmUseItems.txtMerek.Caption := qryDtlSPK.FieldByName('Merek').AsString;
          frmUseItems.txtJmlSPK.Caption := qryDtlSPK.FieldByName('JmlDiminta').AsString;
          frmUseItems.txtJmlSdhKeluar.Caption :=
qryDtlSPK.FieldByName('JmlDipakai').AsString;
          frmUseItems.edtJmlAmbil.Text :=
qryDtlSPK.FieldByName('SelisihPemakaian').AsString;
          frmUseItems.ShowModal;
          RefreshDtlSPK;
        end
      else
        begin
          frmUseItemsTires.txtNoSPK.Caption := qryMsSPK.FieldByName('NoSPK').AsString;
          frmUseItemsTires.lblNoUrut.Caption := qryDtlSPK.FieldByName('NoUrut').AsString;
          frmUseItemsTires.lblNoPolisi.Caption :=
qryPerbaikan.FieldByName('NoPolisi').AsString;

```



```

    frmUseItemsTires.txtKodeBarang.Caption :=
qryDtSPK.FieldByName('ItemsCode').AsString;
    frmUseItemsTires.txtNamaBarang.Caption :=
qryDtSPK.FieldByName('ItemsName').AsString;
    frmUseItemsTires.txtMerek.Caption := qryDtSPK.FieldByName('Merek').AsString;
    frmUseItemsTires.txtJmlSPK.Caption :=
qryDtSPK.FieldByName('JmlDiminta').AsString;
    frmUseItemsTires.txtJmlSdhKeluar.Caption :=
qryDtSPK.FieldByName('JmlDipakai').AsString;
    frmUseItemsTires.ShowModal;
    RefreshDtSPK;
  end;
end;end;
procedure TfrmDaftarSPK.btnCancelClick(Sender: TObject);
begin
  qryPerbaikan.Close;
  qryMsSPK.Close;
  qryDtSPK.Close;
  Close;
end;
procedure TfrmDaftarSPK.btnConfirmClick(Sender: TObject);
begin
  if not(qryMsSPK.IsEmpty) then
  begin
    if chkEksternal.Checked then
    begin
      frmInputSPKEksternal.txtStatus.Caption := 'CONFIRM';
      frmInputSPKEksternal.ShowModal;
    end
    else
    begin
      frmInputSPK.txtStatus.Caption := 'CONFIRM';
      frmInputSPK.ShowModal;
    end;
    btnRefresh.Click;
  end
  else
  ShowMessage('Tidak ada SPK yang mau dikonfirmasi !');
end;
procedure TfrmDaftarSPK.btnDocumentClick(Sender: TObject);
begin
  frmInputSPK.txtStatus.Caption := 'CLOSE';
  frmInputSPK.ShowModal;
  btnRefresh.Click;
end;
procedure TfrmDaftarSPK.btnEditDraftClick(Sender: TObject);
begin
  if not(qryMsSPK.IsEmpty) then

```

```

begin
  if chkEksternal.Checked then
    begin
      frmInputSPKEksternal.txtStatus.Caption := 'EDIT';
      frmInputSPKEksternal.ShowModal;
    end
  else
    begin
      frmInputSPK.txtStatus.Caption := 'EDIT';
      frmInputSPK.ShowModal;
    end;
    btnRefresh.Click;
  end
else
  ShowMessage('Tidak ada SPK yang mau dirubah !');
end;
procedure TfrmDaftarSPK.btnRefreshClick(Sender: TObject);
begin
  if cbStatus.Text = 'OPEN' then
    temp:= 'SELECT DISTINCT (Tbl_MsMintaPerbaikan.NoMintaPerbaikan) AS No, ' +
      'Tbl_MsMintaPerbaikan.*, Tbl_MsSPK.StatusSPK FROM ' +
      'Tbl_MsMintaPerbaikan LEFT OUTER JOIN Tbl_MsSPK ON ' +
      'dbo.Tbl_MsMintaPerbaikan.NoMintaPerbaikan = ' +
      'dbo.Tbl_MsSPK.NoMintaPerbaikan WHERE (TglMintaPerbaikan >= CONVERT' +
      '(DATETIME, ' + QuotedStr(FormatDateTime('yyyy-mm-dd',dtpAwal.Date) +
      '00:00:00') + ', 102)) AND (TglMintaPerbaikan <= CONVERT' +
      '(DATETIME, ' + QuotedStr(FormatDateTime('yyyy-mm-dd',dtpAkhir.Date) +
      '00:00:00') + ', 102)) AND (StatusSPK = ' +
      QuotedStr(cbStatus.Text) + ' OR StatusSPK IS NULL) ORDER BY
Tbl_MsMintaPerbaikan.NoMintaPerbaikan'
  else
    temp:= 'SELECT DISTINCT (Tbl_MsMintaPerbaikan.NoMintaPerbaikan) AS No, ' +
      'Tbl_MsMintaPerbaikan.*, Tbl_MsSPK.StatusSPK FROM ' +
      'Tbl_MsMintaPerbaikan INNER JOIN Tbl_MsSPK ON ' +
      'dbo.Tbl_MsMintaPerbaikan.NoMintaPerbaikan = ' +
      'dbo.Tbl_MsSPK.NoMintaPerbaikan WHERE (TglMintaPerbaikan >= CONVERT' +
      '(DATETIME, ' + QuotedStr(FormatDateTime('yyyy-mm-dd',dtpAwal.Date) +
      '00:00:00') + ', 102)) AND (TglMintaPerbaikan <= CONVERT' +
      '(DATETIME, ' + QuotedStr(FormatDateTime('yyyy-mm-dd',dtpAkhir.Date) +
      '00:00:00') + ', 102)) AND StatusSPK = ' +
      QuotedStr(cbStatus.Text) + ' ORDER BY
Tbl_MsMintaPerbaikan.NoMintaPerbaikan';
    with qryPerbaikan do
      begin
        Close;
        SQL.Clear;
        SQL.Add(temp);
        Open;
      end
    end;
end;

```

```

end;
RefreshMsSPK;
end;
procedure TfrmDaftarSPK.btnTutupSPKClick(Sender: TObject);
begin
if not(qryMsSPK.IsEmpty) then
begin
if chkEksternal.Checked then
begin
frmInputSPKEksternal.txtStatus.Caption := 'CLOSE';
frmInputSPKEksternal.ShowModal;
end
else
begin
frmInputSPK.txtStatus.Caption := 'CLOSE';
frmInputSPK.ShowModal;
end;
end;
btnRefresh.Click;
end
else
ShowMessage('Tidak ada SPK yang mau ditutup !');
end;
procedure TfrmDaftarSPK.cbStatusChange(Sender: TObject);
begin
if cbStatus.Text = 'OPEN' then
begin
btnConfirm.Enabled := True;
btnEditDraft.Enabled := True;
btnTutupSPK.Enabled := False;
btnAmbil.Enabled := False;
end
else
begin
btnConfirm.Enabled := False;
btnEditDraft.Enabled := False;
btnTutupSPK.Enabled := True;
btnAmbil.Enabled := True;
end;
end;
if cbStatus.Text = 'CLOSE' then
begin
btnTutupSPK.Enabled := False;
btnAmbil.Enabled := False;
end;
end;
btnRefresh.Click;
end;
procedure TfrmDaftarSPK.chkEksternalClick(Sender: TObject);
begin
RefreshMsSPK;

```

```

end;
procedure TfrmDaftarSPK.DBAdvGrid1Click(Sender: TObject);
begin
    RefreshMsSPK;
end;
procedure TfrmDaftarSPK.dtpAkhirChange(Sender: TObject);
begin
    if dtpAwal.Date > dtpAkhir.Date then
        dtpAwal.Date := dtpAkhir.Date;
        btnRefresh.Click;
end;
procedure TfrmDaftarSPK.dtpAwalChange(Sender: TObject);
begin
    if dtpAwal.Date > dtpAkhir.Date then
        dtpAwal.Date := dtpAkhir.Date;
        btnRefresh.Click;
end;
procedure TfrmDaftarSPK.FormShow(Sender: TObject);
begin
    dtpAwal.Date := Date - StrToInt(FormatDateTime('dd',Date)) + 1;
    dtpAkhir.Date := Date;
    cbStatus.ItemIndex := 0;
    chkEksternal.Checked := False;
    btnRefresh.Click;
end;
procedure TfrmDaftarSPK.grdSPKClick(Sender: TObject);
begin
    RefreshDtlSPK;
end;
procedure TfrmDaftarSPK.grdSPKDb1Click(Sender: TObject);
begin
    if not(qryMsSPK.IsEmpty) then
        begin
            if chkEksternal.Checked then
                begin
                    frmInputSPKEksternal.txtStatus.Caption := 'VIEW';
                    frmInputSPKEksternal.ShowModal;
                end
            else
                begin
                    frmInputSPK.txtStatus.Caption := 'VIEW';
                    frmInputSPK.ShowModal;
                end;
            end;
            btnRefresh.Click;
        end
    else
        ShowMessage('Tidak ada SPK yang mau dilihat !');
end;
end;

```

```

procedure TfrmDaftarSPK.RefreshDtlSPK;
begin
  with qryDtlSPK do
    begin
      Parameters[0].Value := qryMsSPK.Fields[0].Value;
      Close;
      Open;
    end;
end;
procedure TfrmDaftarSPK.RefreshMsSPK;
begin
  with qryMsSPK do
    begin
      Parameters[0].Value := qryPerbaikan.Fields[0].Value;
      Parameters[1].Value := cbStatus.Text;
      if not chkEksternal.Checked then
        begin
          Parameters.Items[2].Value := 'INTERNAL';
        end
      else
        begin
          Parameters.Items[2].Value := 'EKSTERNAL'
        end;
      Close;
      Open;
    end;
  RefreshDtlSPK;
end;
end.

```

## **FormInputSPK.pas**

```
unit FormInputSPK;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
  Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.Buttons, Vcl.DBCtrls,
  AdvCombo, ColCombo, Vcl.DBLookup, Data.DB, Data.Win.ADODB, Vcl.ComCtrls,
  Vcl.Grids, AdvObj, BaseGrid, AdvGrid, DBAdvGrid, TaskDialog;
type
  TfrmInputSPK = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    grpAction: TGroupBox;
    btnCancel: TSpeedButton;
    btnSaveConfirm: TSpeedButton;
    btnSaveDraft: TSpeedButton;
    btnPrintShipper: TSpeedButton;
    btnPrintBuyer: TSpeedButton;
    txtStatus: TStaticText;
    dbtxtNoPermintaan: TDBText;
    Label1: TLabel;
    Label2: TLabel;
    dbtxtTglMinta: TDBText;
    Label3: TLabel;
    dbtxtDeskPerbaikan: TDBText;
    Label4: TLabel;
    dbtxtDeskSparepart: TDBText;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    txtNoSPK: TStaticText;
    dtpSPK: TDateTimePicker;
    txtStatusSPK: TStaticText;
    cbNIK: TComboBox;
    cbName: TComboBox;
    edtKetSPK: TEdit;
    grdDtlSPK: TDBAdvGrid;
    qrySQL: TADOQuery;
    qryTempDtlSPK: TADOQuery;
    dsDtlSPK: TDataSource;
    qryTempDtlSPKIDSystemUser: TWideStringField;
    qryTempDtlSPKNoUrut: TSmallintField;
    qryTempDtlSPKItemsCode: TWideStringField;
    qryTempDtlSPKJmlDiminta: TBCDField;
```

```

qryTempDtlSPKJmlDipakai: TBCDField;
qryTempDtlSPKSelisihPemakaian: TBCDField;
qryTempDtlSPKCost: TBCDField;
qryTempDtlSPKItemsName: TWideStringField;
grpDetail: TGroupBox;
Label10: TLabel;
cbItemsCode: TComboBox;
cbItemsName: TComboBox;
Label11: TLabel;
edtJmlDiminta: TEdit;
qryTempDtlSPKUOM: TWideStringField;
lblUOM: TLabel;
btnSubmit: TButton;
btnEditPay: TButton;
btnDeletePay: TButton;
qryMstSPK: TADOQuery;
cbUOM: TComboBox;
lblStatusDetail: TLabel;
btnTutupSPK: TSpeedButton;
Label12: TLabel;
cbMerek: TComboBox;
qryTempDtlSPKMerek: TWideStringField;
InputKetTutup: TAdvInputTaskDialog;
Label13: TLabel;
dbtxtNoPolisi: TDBText;
procedure btnCancelClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure BersihkanDtlSPK;
procedure BersihkanSPK;
procedure StatusReadOnly(Status:Boolean);
procedure BersihkanTabelTemp;
procedure IsiComboBox;
procedure OpenQrySQL;
procedure OpenQryTemp;
procedure SaveSeluruh;
procedure BuatNoSPK;
procedure cbNIKChange(Sender: TObject);
procedure cbItemsCodeChange(Sender: TObject);
procedure cbItemsNameChange(Sender: TObject);
procedure cbNameChange(Sender: TObject);
procedure btnSubmitClick(Sender: TObject);
procedure btnSaveDraftClick(Sender: TObject);
procedure btnDeletePayClick(Sender: TObject);
procedure btnEditPayClick(Sender: TObject);
procedure btnSaveConfirmClick(Sender: TObject);
procedure ViewSPK;
function Cek : Boolean;
procedure btnTutupSPKClick(Sender: TObject);

```

```

procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure UpdateQtySPKTambah;
procedure IsiComboMerek;
procedure TutupSPK;
procedure edtJmlDimintaKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  frmInputSPK: TfrmInputSPK;
implementation
uses FormDaftarSPK, FormComponent, GlobalUnit;
var temp, NoSPK : string;
    NoUrut,i : Integer;
    TotSelisih : Single;
{$R *.dfm}
function TfrmInputSPK.Cek : Boolean;
begin
  Cek := True;
  if cbNIK.Text = " then
  begin
    Showmessage('NIK Required!');
    cbNIK.SetFocus;
    Cek := false;
    exit;
  end
  else if cbName.Text = " then
  begin
    Showmessage('Nama Required!');
    cbName.SetFocus;
    Cek := false;
    exit;
  end
  else if cbItemsCode.Text = " then
  begin
    Showmessage('Items Code Required!');
    cbItemsCode.SetFocus;
    Cek := false;
    exit;
  end
  else if cbItemsName.Text = " then
  begin
    Showmessage('Items Name Required!');
    cbItemsName.SetFocus;
    Cek := false;
    exit;
  end;

```



```

end
else if cbMerek.Text = " then
begin
  Showmessage('Merk Required!');
  cbMerek.SetFocus;
  Cek := false;
  exit;
end
else if edtJmlDiminta.Text = " then
begin
  Showmessage('Jumlah Dibutuhkan Required!');
  edtJmlDiminta.SetFocus;
  Cek := false;
  exit;
end
else if StrToInt(edtJmlDiminta.Text) <= 0 then
begin
  Showmessage('Jumlah Dibutuhkan harus lebih dari nol!');
  edtJmlDiminta.SetFocus;
  Cek := false;
  exit; end end;
procedure TfrmInputSPK.BersihkanDtlSPK;
begin
  cbItemsCode.ItemIndex := -1;
  cbItemsName.ItemIndex := -1;
  cbMerek.Items.Clear;
  edtJmlDiminta.Clear;
  lblUOM.Caption := 'UOM';
end;
procedure TfrmInputSPK.BersihkanSPK;
begin
  txtNoSPK.Caption := 'SPK BARU';
  dtpSPK.Date := Date;
  cbNIK.ItemIndex := -1;
  cbName.ItemIndex := -1;
  edtKetSPK.Clear;
  BersihkanDtlSPK;
end;
procedure TfrmInputSPK.BersihkanTabelTemp;
begin
  temp := 'DELETE FROM Tbl_TempDtlSPK WHERE IDSystemUser = ' +
    QuotedStr(frmComponent.lblUserName.Caption);
  with qrySQL do
  begin
    Close;
    SQL.Clear;
    SQL.Add(temp);
    ExecSQL;
  end;
end;

```

```

end;
end;
procedure TfrmInputSPK.btnCancelClick(Sender: TObject);
begin
    Close;
end;
procedure TfrmInputSPK.btnDeletePayClick(Sender: TObject);
var Jawab : Integer;
begin
    if not(qryTempDtlSPK.IsEmpty) then
        begin
            Jawab := MessageDlgPos('Sparepart ini mau
dihapus?',mtCustom,mbOKCancel,0,Left+300,Top+430);
            if Jawab = mrOK then
                qryTempDtlSPK.Delete;
                OpenQryTemp;
            end;
        end;
end;
procedure TfrmInputSPK.btnEditPayClick(Sender: TObject);
begin
    if not(qryTempDtlSPK.IsEmpty) then
        with qryTempDtlSPK do
            begin
                NoUrut := FieldByName('NoUrut').AsInteger;
                edtJmlDiminta.Text := FieldByName('JmlDiminta').AsString;
                i := 0;
                repeat
                    cbItemsCode.ItemIndex := i;
                    i := i + 1;
                until (cbItemsCode.Text = FieldByName('ItemsCode').AsString) or (i >
cbItemsCode.Items.Count);
                if (i > cbItemsCode.Items.Count) then
                    cbItemsCode.ItemIndex := -1;
                    cbItemsName.ItemIndex := cbItemsCode.ItemIndex;
                    cbUOM.ItemIndex := cbItemsCode.ItemIndex;
                    lblUOM.Caption := cbUOM.Text;
                    IsiComboMerek;
                    i := 0;
                    repeat
                        cbMerek.ItemIndex := i;
                        i := i + 1;
                    until (cbMerek.Text = FieldByName('Merek').AsString) or (i > cbMerek.Items.Count);
                    if (i > cbMerek.Items.Count) then
                        cbMerek.ItemIndex := -1;
                    qryTempDtlSPK.Delete;
                    btnEditPay.Enabled := False;
                    btnDeletePay.Enabled := False;
                    grdDtlSPK.Enabled := False;

```

```

    btnSubmit.Enabled := True;
end;
end;
procedure TfrmInputSPK.btnSaveConfirmClick(Sender: TObject);
begin
    if not frmComponent.adoTerra.InTransaction then
        with qryMstSPK do
            begin
                frmComponent.adoTerra.BeginTrans;
            try
                temp := 'UPDATE Tbl_MsSPK SET StatusSPK = ' + QuotedStr('CONFIRM') +
                    ', IDSystemConfirm = ' + QuotedStr(frmComponent.lblUserName.Caption ) +
                    ', ConfirmTime = ' + QuotedStr(FormatDateTime('yyyy-mm-dd',Date) + '' +
FormatDateTime('hh:mm',Time)) +
                    ' WHERE NoSPK = ' + QuotedStr(NoSPK);
                Close;
                SQL.Clear;
                SQL.Add(temp);
                ExecSQL;
                qryTempDtlSPK.First;
                while not(qryTempDtlSPK.Eof) do
                    begin
                        UpdateQtySPKTambah;
                        qryTempDtlSPK.Next;
                    end;
                frmComponent.adoTerra.CommitTrans;
                btnSaveConfirm.Enabled := False;
                ShowMessage('SPK CONFIRM');
            Except
                frmComponent.adoTerra.RollbackTrans;
            end; // end try
        end; //end if ado.InTrans / with qry
    end;
    procedure TfrmInputSPK.btnSaveDraftClick(Sender: TObject);
    begin
        if not qryTempDtlSPK.Eof then
            begin
                SaveSeluruh;
                btnSaveDraft.Enabled := False;
            end
        else
            Showmessage('Tidak ada data');
        end;
    procedure TfrmInputSPK.btnSubmitClick(Sender: TObject);
    begin
        if Cek then
            with qryTempDtlSPK do
                begin

```

```

Close;
Open;
if {(lblStatusDetail.Caption = 'NEW') and } (not(IsEmpty)) then
begin
    Last;
    NoUrut := FieldByName('NoUrut').AsInteger + 1;
end;
if {(lblStatusDetail.Caption = 'NEW') and } (IsEmpty) then
    NoUrut := 1;
Insert;
FieldByName('IDSystemUser').AsString := frmComponent.lblUserName.Caption;
FieldByName('NoUrut').AsInteger := NoUrut;
FieldByName('ItemsCode').AsString := cbItemsCode.Text;
FieldByName('ItemsName').AsString := cbItemsName.Text;
FieldByName('Merek').AsString := cbMerek.Text;
FieldByName('UOM').AsString := lblUOM.Caption;
FieldByName('JmlDiminta').AsFloat := StrToFloat(edtJmlDiminta.Text);
{if edtJmlDipakai.Text <> " then
    FieldByName('JmlDipakai').AsFloat := StrToFloat(edtJmlDipakai.Text);
if edtCost.Text <> " then
    FieldByName('Cost').AsCurrency := StrToCurr(edtCost.Text); }
Post;
Close;
Open;
BersihkanDtlSPK;
btnEditPay.Enabled := True;
btnDeletePay.Enabled := True;
grdDtlSPK.Enabled := True;
end;
end;
procedure TfrmInputSPK.btnTutupSPKClick(Sender: TObject);
var Jawab : integer;
begin
    if TotSelisih <> 0 then
        begin
            Jawab := MessageDlgPos('SPK ini benar mau ditutup? Ada Barang yang belum diterima
!',mtCustom,mbOKCancel,0,Left+300,Top+430);
            if Jawab = mrOK then
                TutupSPK
            else
                ShowMessage('Masih ada Items yang belum diterima !');
        end
    else
        TutupSPK;
end;
end;
procedure TfrmInputSPK.BuatNoSPK;
var TempNoSPK : integer;
begin

```

```

temp := 'SELECT NoSPK FROM Tbl_MsSPK ORDER BY NoSPK';
OpenQrySQL;
if not(qrySQL.IsEmpty) then
begin
    qrySQL.Last;
    TempNoSPK := qrySQL.FieldByName('NoSPK').AsInteger + 1;
end
else
    TempNoSPK := 1;
NoSPK := TambahNoRibuan(TempNoSPK);
end;
procedure TfrmInputSPK.cbItemsCodeChange(Sender: TObject);
begin
    cbItemsName.ItemIndex := cbItemsCode.ItemIndex;
    cbUOM.ItemIndex := cbItemsCode.ItemIndex;
    lblUOM.Caption := cbUOM.Text;
    IsiComboMerek;
end;
procedure TfrmInputSPK.cbItemsNameChange(Sender: TObject);
begin
    cbItemsCode.ItemIndex := cbItemsName.ItemIndex;
    cbUOM.ItemIndex := cbItemsCode.ItemIndex;
    lblUOM.Caption := cbUOM.Text;
    IsiComboMerek;
end;
procedure TfrmInputSPK.cbNameChange(Sender: TObject);
begin
    cbNIK.ItemIndex := cbName.ItemIndex;
end;
procedure TfrmInputSPK.cbNIKChange(Sender: TObject);
begin
    cbName.ItemIndex := cbNIK.ItemIndex;
end;
procedure TfrmInputSPK.edtJmlDimintaKeyPress(Sender: TObject; var Key: Char);
begin
    if not (Key in ['0'..'9','.', '#27, #13, #8, #9']) then
        begin
            edtJmlDiminta.Clear;
            edtJmlDiminta.SetFocus;
            Key := #0;
        end end;
procedure TfrmInputSPK.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    BersihkanTabelTemp;
end;
procedure TfrmInputSPK.FormShow(Sender: TObject);
begin
    BersihkanTabelTemp;

```

```

IsiComboBox;
if txtStatus.Caption = 'NEW' then
begin
  NoSPK := 'NEW';
  BersihkanSPK;
  StatusReadOnly(FALSE);
  btnSaveDraft.Enabled := True;
  btnSaveConfirm.Enabled := False;
  btnTutupSPK.Enabled := False;
  txtStatusSPK.Caption := 'OPEN';
end;
if txtStatus.Caption = 'CONFIRM' then
begin
  NoSPK := frmDaftarSPK.qryMsSPK.FieldByName('NoSPK').AsString;
  ViewSPK;
  StatusReadOnly(TRUE);
  btnSaveConfirm.Enabled := True;
  btnSaveDraft.Enabled := False;
  btnTutupSPK.Enabled := False;
  txtStatusSPK.Caption := 'TO CLOSE';
end;
if txtStatus.Caption = 'CLOSE' then
begin
  NoSPK := frmDaftarSPK.qryMsSPK.FieldByName('NoSPK').AsString;
  ViewSPK;
  StatusReadOnly(TRUE);
  btnSaveConfirm.Enabled := False;
  btnSaveDraft.Enabled := False;
  btnTutupSPK.Enabled := True;
  txtStatusSPK.Caption := 'TO CONFIRM';
end;
if txtStatus.Caption = 'VIEW' then
begin
  NoSPK := frmDaftarSPK.qryMsSPK.FieldByName('NoSPK').AsString;
  ViewSPK;
  StatusReadOnly(TRUE);
  btnSaveConfirm.Enabled := False;
  btnSaveDraft.Enabled := False;
  btnTutupSPK.Enabled := False;
  txtStatusSPK.Caption :=
frmDaftarSPK.qryMsSPK.FieldByName('StatusSPK').AsString;
end;
if txtStatus.Caption = 'EDIT' then
begin
  NoSPK := frmDaftarSPK.qryMsSPK.FieldByName('NoSPK').AsString;
  ViewSPK;
  StatusReadOnly(FALSE);
  btnSaveDraft.Enabled := True;

```

```

    btnSaveConfirm.Enabled := False;
    btnTutupSPK.Enabled := False;
    txtStatusSPK.Caption :=
frmDaftarSPK.qryMsSPK.FieldByName('StatusSPK').AsString;
    end;
    txtNoSPK.Caption := NoSPK;
    lblStatusDetail.Caption := 'NEW';
end;
procedure TfrmInputSPK.IsiComboBox;
begin
    temp := 'SELECT EmployeeCode, EmployeeName FROM Tbl_MstEmployee WHERE
EmployeeGroup = ' +
        QuotedStr('Mekanik') + ' ORDER BY EmployeeName';
    OpenQrySQL;
    cbNIK.Items.Clear;
    cbName.Items.Clear;
    while not(qrySQL.Eof) do
    begin
        cbNIK.Items.Add(qrySQL.FieldByName('EmployeeCode').AsString);
        cbName.Items.Add(qrySQL.FieldByName('EmployeeName').AsString);
        qrySQL.Next;
    end;
    temp := 'SELECT DISTINCT ItemsCode, ItemsName, UOM FROM Tbl_MstItems
ORDER BY ItemsCode';
    OpenQrySQL;
    cbItemsCode.Items.Clear;
    cbItemsName.Items.Clear;
    while not(qrySQL.Eof) do
    begin
        cbItemsCode.Items.Add(qrySQL.FieldByName('ItemsCode').AsString);
        cbItemsName.Items.Add(qrySQL.FieldByName('ItemsName').AsString);
        cbUOM.Items.Add(qrySQL.FieldByName('UOM').AsString);
        qrySQL.Next;
    end;
end;
procedure TfrmInputSPK.IsiComboMerek;
begin
    temp := 'SELECT Merek FROM Tbl_MstItems +
        'WHERE ItemsCode = ' + QuotedStr(cbItemsCode.Text) +
        ' ORDER BY Merek';
    OpenQrySQL;
    cbMerek.Items.Clear;
    while not(qrySQL.Eof) do
    begin
        cbMerek.Items.Add(qrySQL.FieldByName('Merek').AsString);
        qrySQL.Next;
    end;
end;
end;

```

```

procedure TfrmInputSPK.OpenQrySQL;
begin
  with qrySQL do
  begin
    Close;
    SQL.Clear;
    SQL.Add(temp);
    Open;
  end;
end;

procedure TfrmInputSPK.OpenQryTemp;
begin
  temp := 'SELECT * FROM Tbl_TempDtSPK WHERE IDSystemUser = ' +
    QuotedStr(frmComponent.IblUserName.Caption);
  with qryTempDtSPK do
  begin
    Close;
    SQL.Clear;
    SQL.Add(temp);
    Open;
  end;
end;

procedure TfrmInputSPK.SaveSeluruh;
begin
  temp := 'SELECT * FROM Tbl_MsSPK WHERE NoSPK = ' + QuotedStr(NoSPK);
  with qryMstSPK do
  begin
    Close;
    SQL.Clear;
    SQL.Add(temp);
    Open;
    if not frmComponent.adoTerra.InTransaction then
    begin
      frmComponent.adoTerra.BeginTrans;
      try
        if txtStatus.Caption = 'EDIT' then
          Edit
        else
          Insert;
        If txtStatus.Caption = 'NEW' then
          BuatNoSPK;
        FieldByName('NoSPK').AsString := NoSPK;
        FieldByName('StatusSPK').AsString := 'OPEN';
        FieldByName('TglSPK').AsDateTime := dtpSPK.Date;
        FieldByName('NoMintaPerbaikan').AsString := dbtxtNoPermintaan.Caption;
        FieldByName('IDMekanik').AsString := cbNIK.Text;
        FieldByName('KeteranganSPK').AsString := edtKetSPK.Text;
        FieldByName('JenisSPK').AsString := 'INTERNAL';
      except
      end;
    end;
  end;
end;

```



```

FieldByName('IDSystemUser').AsString := frmComponent.lblUserName.Caption;
FieldByName('UpdateTime').AsDateTime := Date;
Post;
Refresh;
if txtStatus.Caption = 'EDIT' then
with qrySQL do
begin
temp := 'DELETE FROM Tbl_DtlSPK WHERE NoSPK = ' + QuotedStr(NoSPK);
Close;
SQL.Clear;
SQL.Add(temp);
ExecSQL;
end;
temp := 'SELECT * FROM Tbl_DtlSPK WHERE NoSPK = ' + QuotedStr(NoSPK);
OpenQrySQL;
qryTempDtlSPK.First;
while not(qryTempDtlSPK.Eof) do
begin
qrySQL.Insert;
qrySQL.FieldByName('NoSPK').AsString := NoSPK;
qrySQL.FieldByName('NoUrut').AsInteger :=
qryTempDtlSPK.FieldByName('NoUrut').AsInteger;
qrySQL.FieldByName('ItemsCode').AsString
:=qryTempDtlSPK.FieldByName('ItemsCode').AsString;
qrySQL.FieldByName('Merek').AsString
:=qryTempDtlSPK.FieldByName('Merek').AsString;
qrySQL.FieldByName('JmlDiminta').AsFloat :=
qryTempDtlSPK.FieldByName('JmlDiminta').AsFloat;
{qrySQL.FieldByName('JmlDipakai').AsFloat :=
qryTempDtlSPK.FieldByName('JmlDipakai').AsFloat;
qrySQL.FieldByName('Cost').AsCurrency :=
qryTempDtlSPK.FieldByName('Cost').AsCurrency;}
qrySQL.FieldByName('IDSystemUser').AsString :=
frmComponent.lblUserName.Caption;
qrySQL.FieldByName('UpdateTime').AsDateTime := Date;
qrySQL.Post;
qryTempDtlSPK.Next;
end;
frmComponent.adbTerra.CommitTrans;
ShowMessage('SAVE FINISH');
Except
frmComponent.adbTerra.RollbackTrans;
end; end; end; end;
procedure TfrmInputSPK.StatusReadOnly(Status: Boolean);
begin
dtpSPK.Enabled := Not(Status);
cbNIK.Enabled := Not(Status);
cbName.Enabled := Not(Status);

```

```

edtKetSPK.ReadOnly := (Status);
cbItemsCode.Enabled := Not(Status);
cbItemsName.Enabled := Not(Status);
edtJmlDiminta.ReadOnly := (Status);
{ edtJmlDipakai.ReadOnly := (Status);
edtSelisih.ReadOnly := (Status);
edtCost.ReadOnly := (Status);}
grpDetail.Enabled := Not(Status);
end;
procedure TfrmInputSPK.TutupSPK;
var TotTerima : String;
begin
if not frmComponent.adoTerra.InTransaction then
with qryMstSPK do
begin
frmComponent.adoTerra.BeginTrans;
try
InputKetTutup.Execute;
temp := 'UPDATE Tbl_MsSPK SET StatusSPK = ' + QuotedStr('CLOSE') +
', KeteranganTutupSPK = ' + QuotedStr(InputKetTutup.InputText) +
', IDSystemClose = ' + QuotedStr(frmComponent.lblUserName.Caption) +
', CloseTime = ' + QuotedStr(FormatDateTime('yyyy-mm-dd',Date) + '' +
FormatDateTime('hh:mm',Time)) +
' WHERE NoSPK = ' + QuotedStr(NoSPK);
Close;
SQL.Clear;
SQL.Add(temp);
ExecSQL;
temp := 'UPDATE Tbl_MstVehicle SET StatusKendaraan = ' + QuotedStr('SELESAI') +
', NoDokAkhir = ' + QuotedStr(NoSPK) + ' WHERE NoPolisi = ' +
QuotedStr(dbtxtNoPolisi.Caption);
Close;
SQL.Clear;
SQL.Add(temp);
ExecSQL;
frmComponent.adoTerra.CommitTrans;
btnTutupSPK.Enabled := False;
ShowMessage('SPK CLOSE');
Except
frmComponent.adoTerra.RollbackTrans;
end; end; end;
procedure TfrmInputSPK.UpdateQtySPKTambah;
begin
temp := 'UPDATE Tbl_MstItems SET QtyForSPK = QtyForSPK + ' +
qryTempDtlSPK.FieldByName('JmlDiminta').AsString + ' WHERE ItemsCode = ' +
QuotedStr(qryTempDtlSPK.FieldByName('ItemsCode').AsString) +
' AND Merek = ' + QuotedStr(qryTempDtlSPK.FieldByName('Merek').AsString);
with qrySQL do

```

```

begin
  Close;
  SQL.Clear;  SQL.Add(temp);  ExecSQL;
end; end;
procedure TfrmInputSPK.ViewSPK;
begin
  with frmDaftarSPK.qryMsSPK do
  begin
    txtNoSPK.Caption := NoSPK;
   .dtpSPK.Date := FieldByName('TglSPK').AsDateTime;
    i := 0;
    repeat
      cbNIK.ItemIndex := i;
      i := i + 1;
    until (cbNIK.Text = FieldByName('IDMekanik').AsString) or (i > cbNIK.Items.Count);
    if (i > cbNIK.Items.Count) then
      cbNIK.ItemIndex := -1;
      cbName.ItemIndex := cbNIK.ItemIndex;
      edtKetSPK.Text := FieldByName('KeteranganSPK').AsString;
    end;
    OpenQryTemp;  TotSelisih := 0;
    with frmDaftarSPK.qryDtlSPK do
    begin
      First;
      while not(Eof) do  begin
        qryTempDtlSPK.Insert;
        qryTempDtlSPK.FieldByName('IDSystemUser').AsString :=
frComponent.lblUserName.Caption;
        qryTempDtlSPK.FieldByName('NoUrut').AsInteger :=
FieldByName('NoUrut').AsInteger;
        qryTempDtlSPK.FieldByName('ItemsCode').AsString :=
FieldByName('ItemsCode').AsString;
        qryTempDtlSPK.FieldByName('ItemsName').AsString :=
FieldByName('ItemsName').AsString;
        qryTempDtlSPK.FieldByName('Merek').AsString := FieldByName('Merek').AsString;
        qryTempDtlSPK.FieldByName('UOM').AsString := FieldByName('UOM').AsString;
        qryTempDtlSPK.FieldByName('JmlDiminta').AsFloat :=
FieldByName('JmlDiminta').AsFloat;
        qryTempDtlSPK.FieldByName('JmlDipakai').AsFloat :=
FieldByName('JmlDipakai').AsFloat;
        qryTempDtlSPK.FieldByName('Cost').AsCurrency :=
FieldByName('Cost').AsCurrency;
        qryTempDtlSPK.Post;
        TotSelisih := TotSelisih + FieldByName('SelisihPemakaian').AsFloat;;
        Next;  end;  end;  end;  OpenQryTemp;end;end.

```

## **FormUseItems.pas**

```
unit FormUseItems;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
  Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ComCtrls, Vcl.Buttons,
  Vcl.Grids, AdvObj, BaseGrid, AdvGrid, DBAdvGrid, Data.DB, Data.Win.ADODB;
type
  TfrmUseItems = class(TForm)
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    txtNoSPK: TStaticText;
    txtKodeBarang: TStaticText;
    txtJmlSPK: TStaticText;
    txtJmlSdhKeluar: TStaticText;
    dtpAmbil: TDateTimePicker;
    edtJmlAmbil: TEdit;
    btnClose: TSpeedButton;
    lblNoUrut: TLabel;
    qrySQL: TADOQuery;
    lblHarga: TLabel;
    txtNamaBarang: TStaticText;
    DBAdvGrid1: TDBAdvGrid;
    qryGrid: TADOQuery;
    dsGrid: TDataSource;
    qryGridNoSPK: TWideStringField;
    qryGridNoUrut: TSmallintField;
    qryGridNoUrutAmbil: TSmallintField;
    qryGridTglAmbil: TDateTimeField;
    qryGridJumlahAmbil: TBCDField;
    qryGridIDWarehouse: TWideStringField;
    qryGridIDSystemUser: TWideStringField;
    qryGridUpdateTime: TDateTimeField;
    qryGridItemsCode: TWideStringField;
    qryGridItemsName: TWideStringField;
    btnAmbil: TSpeedButton;
    txtMerek: TStaticText;
    Label7: TLabel;
    procedure btnCloseClick(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure btnAmbilClick(Sender: TObject);
  private
```

```

    { Private declarations }
Public
    { Public declarations }
end;
var
    frmUseItems: TfrmUseItems;
implementation
    {$R *.dfm}
Uses FormComponent;
var temp : string;
procedure TfrmUseItems.btnCloseClick(Sender: TObject);
begin
    Close;end;
procedure TfrmUseItems.btnAmbilClick(Sender: TObject);
var NoUrutAmbil : integer;
begin
    temp := 'SELECT * FROM Tbl_MstItems WHERE ItemsCode = ' +
    QuotedStr(txtKodeBarang.Caption) +
        ' AND Merek = ' + QuotedStr(txtMerek.Caption);
    with qrySQL do
        begin
            Close;
            SQL.Clear;
            SQL.Add(temp);
            Open; end;
    temp := 'SELECT * FROM Tbl_UseItems WHERE NoSPK = ' +
    QuotedStr(txtNoSPK.Caption) +
        ' AND NoUrut = ' + lblNoUrut.Caption + ' ORDER BY NoUrutAmbil';
    if (StrToFloat(edtJmlAmbil.Text) <= qrySQL.FieldByName('QtyStock').AsFloat) and
        (StrToFloat(edtJmlAmbil.Text) <= StrToFloat(txtJmlSPK.Caption) -
    StrToFloat(txtJmlSdhKeluar.Caption)) then
        with qrySQL do
            begin
                Close;
                SQL.Clear;
                SQL.Add(temp);
                Open;
                if not(IsEmpty) then
                    begin
                        Last;
                        NoUrutAmbil := FieldByName('NoUrutAmbil').AsInteger + 1;
                    end
                else
                    NoUrutAmbil := 1;
                if not frmComponent.adoTerra.InTransaction then
                    begin
                        frmComponent.adoTerra.BeginTrans;
                    try

```

```

Insert;
FieldByName('NoSPK').AsString := txtNoSPK.Caption;
FieldByName('NoUrut').AsString := lblNoUrut.Caption;
FieldByName('NoUrutAmbil').AsInteger := NoUrutAmbil;
FieldByName('TglAmbil').AsDateTime := dtpAmbil.Date;
FieldByName('JumlahAmbil').AsFloat := StrToFloat(edtJmlAmbil.Text);
FieldByName('CostPrice').AsCurrency := StrToFloat(lblHarga.Caption);
FieldByName('IDWarehouse').AsString := 'BARU';
FieldByName('IDSystemUser').AsString := frmComponent.lblUserName.Caption;
FieldByName('UpdateTime').AsDateTime := Date;
Post;
temp := 'UPDATE Tbl_DtlSPK SET JmlDipakai = JmlDipakai + ' +
      edtJmlAmbil.Text + ' WHERE NoSPK = ' + QuotedStr(txtNoSPK.Caption) +
      ' AND NoUrut = ' + lblNoUrut.Caption;
Close;SQL.Clear;SQL.Add(temp); ExecSQL;
temp := 'UPDATE Tbl_MstItems SET QtyStock = QtyStock - ' + edtJmlAmbil.Text +
      ', QtyForSPK = QtyForSPK - ' + edtJmlAmbil.Text +
      ' WHERE ItemsCode = ' + QuotedStr(txtKodeBarang.Caption) +
      ' AND Merek = ' + QuotedStr(txtMerek.Caption);
Close;
SQL.Clear;
SQL.Add(temp);
ExecSQL;
frmComponent.adoTerra.CommitTrans;
ShowMessage('PENGAMBILAN BARANG SELESAI');
btnAmbil.Enabled := False;
qryGrid.Close; qryGrid.Open;
Except
  frmComponent.adoTerra.RollbackTrans;end; end; end //end with qry
else
  ShowMessage('Jumlah Barang di Stok tidak cukup atau' + #13 + 'Jumlah Diambil lebih
  besar dari yang dibutuhkan');end;
procedure TfrmUseItems.FormShow(Sender: TObject);
begin
  with qryGrid do
    begin
      Close;
      Parameters[0].Value := txtNoSPK.Caption;
      Parameters[1].Value := lblNoUrut.Caption;
      Open;
    end;
    dtpAmbil.Date := Date;
    edtJmlAmbil.SetFocus;
    btnAmbil.Enabled := True;
  end;end.

```

### **FormPassword.pas (AdminHeadOffice.exe)**

```
unit FormPassword;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, DB, ADODB;
type
  TfrmPassword = class(TForm)
    btnOK: TSpeedButton;
    btnCancel: TSpeedButton;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    edtuser: TEdit;
    edtpass: TEdit;
    qryPass: TADOQuery;
    procedure btnOKClick(Sender: TObject);
    procedure btnCancelClick(Sender: TObject);
    procedure edtuserKeyPress(Sender: TObject; var Key: Char);
    procedure edtpassKeyPress(Sender: TObject; var Key: Char);
    procedure FormShow(Sender: TObject);
    function CekUserID : Boolean;
    function CekPassword(User, Password : string) : Boolean;
    procedure BersihkanData;
    function CekValid : Boolean;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmPassword: TfrmPassword;
implementation
uses
  FormRekomendasiPR, FormComponent;
var
  UserName, Otoritas, temp : string;
{$R *.dfm}
procedure TfrmPassword.btnOKClick(Sender: TObject);
begin
  if CekValid then
  begin
    temp := 'UPDATE Tbl_MstSystemUser SET LoginStat = ' + #39 + 'TRUE' + #39 +
      'WHERE IDSystemUser = ' + #39 + edtuser.Text + #39;
    with qryPass do
    begin
      Close;
```

```

    SQL.Clear;
    SQL.Add(temp);
    ExecSQL;
end;
frmRekomendasiPR.ShowModal;
temp := 'UPDATE Tbl_MstSystemUser SET LoginStat = ' + #39 + 'FALSE' + #39 +
        'WHERE IDSystemUser = ' + #39 + edtuser.Text + #39;
with qryPass do
begin
    Close;
    SQL.Clear;
    SQL.Add(temp);
    ExecSQL; end; Close; end;end;
procedure TfrmPassword.btnCancelClick(Sender: TObject);
begin
    Close;end;
procedure TfrmPassword.edtuserKeyPress(Sender: TObject; var Key: Char);
begin
    if key = #13 then
        edtpass.SetFocus;end;
procedure TfrmPassword.edtpassKeyPress(Sender: TObject; var Key: Char);
begin
    if key = #13 then
        btnOK.Click;end;
procedure TfrmPassword.FormShow(Sender: TObject);
begin
    frmComponent.adoterra.Connected := True;
    BersihkanData;
end;
function TfrmPassword.CekPassword(User, Password : string) : Boolean;
begin
    temp := 'SELECT * FROM Tbl_MstSystemUser WHERE ' +
            'IDSystemUser = ' + #39 + UserName + #39 + ";
    with qryPass do
    begin
        close;
        sql.Clear;
        sql.Add(temp);
        open;
        if not locate('Password',Password,[]) then
            begin
                showmessage('Wrong password, enter again !');
                edtPass.Clear; edtPass.SetFocus;
                CekPassword := false;
            end
        else
            begin
                CekPassword := true;
            end
        end
    end;
end;

```



```

    Otoritas := FieldByName('Rules').AsString;
end;
close;
end;
end;
function TfrmPassword.CekUserID;
begin
temp := 'SELECT * FROM Tbl_MstSystemUser';
with qryPass do
begin
close;
sql.Clear;
sql.Add(temp);
open;
if not locate('IDSystemUser',edtuser.Text,[]) then
begin
showmessage('Wrong or not registered user name, try again !');
edtUser.Clear;
edtUser.SetFocus;
edtPass.Clear;
CekUserID := false;
end
else
CekUserID := true;
UserName := fieldbyname('IDSystemUser').AsString;
close;
end;end;
procedure TfrmPassword.BersihkanData;
begin
edtUser.Clear;
edtPass.Clear;
edtUser.SetFocus;
end;
function TfrmPassword.CekValid : Boolean;
begin
CekValid := true;
if edtUser.Text = " then
begin
Showmessage('User name empty !');
edtUser.Clear;
CekValid := false;
exit; end;
if edtPass.Text = " then
begin
Showmessage('Password still empty !');
edtPass.Clear;
CekValid := false;
exit;

```

```

end;
if Not CekUserID then
begin
  CekValid := false;
  exit; end;
if Not CekPassword(edtUser.Text, edtpass.Text) then
begin
  CekValid := false;
  exit; end;
temp := 'SELECT LoginStat FROM Tbl_MstSystemUser WHERE IDSystemUser = ' +
#39 + edtuser.Text + #39;
with qryPass do
begin
  close;
  sql.Clear;
  sql.Add(temp);
  open;
  if FieldByName('LoginStat').AsString = 'TRUE' then
  begin
    CekValid := false;
    ShowMessage('Your username already LOGIN');
  end; end;end;
procedure TfrmPassword.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
  qryPass.Close;
  frmComponent.adoTerra.Connected := False;
end; end.

```

## **FormRekomendasiPR.pas**

```
unit FormRekomendasiPR;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
  Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ComCtrls, Vcl.Buttons,
  Vcl.ExtCtrls, Data.DB, Data.Win.ADODB, Vcl.Grids, AdvObj, BaseGrid, AdvGrid,
  DBAdvGrid;
type
  TfrmRekomendasiPR = class(TForm)
    GroupBox1: TGroupBox;
    btnPR: TSpeedButton;
    btnBuatPO: TSpeedButton;
    btnConfirm: TSpeedButton;
    Label1: TLabel;
    Label3: TLabel;
    btnCancel: TSpeedButton;
    dtpAwal: TDateTimePicker;
    dtpAkhir: TDateTimePicker;
    GroupBox2: TGroupBox;
    Splitter1: TSplitter;
    GroupBox3: TGroupBox;
    grdRekPR: TDBAdvGrid;
    grdSPK: TDBAdvGrid;
    qryPR: TADOQuery;
    dsPR: TDataSource;
    qryDtlSPK: TADOQuery;
    dsDtlSPK: TDataSource;
    qryDtlSPKNoSPK: TWideStringField;
    qryDtlSPKNoUrut: TSmallintField;
    qryDtlSPKItemsCode: TWideStringField;
    qryDtlSPKJmlDiminta: TBCDField;
    qryDtlSPKJmlDipakai: TBCDField;
    qryDtlSPKSelisihPemakaian: TBCDField;
    qryDtlSPKCost: TBCDField;
    qryDtlSPKIDSystemUser: TWideStringField;
    qryDtlSPKUpdateTime: TDateTimeField;
    qryDtlSPKItemsName: TWideStringField;
    qryDtlSPKUOM: TWideStringField;
    qryPRItemsCode: TWideStringField;
    qryPRItemsName: TWideStringField;
    qryPRUOM: TWideStringField;
    qryPRJmlButuh: TFMTBCDField;
    qryPRQtyStock: TBCDField;
    qryPRQtyOnOrder: TBCDField;
    qryPRJmlPR: TFMTBCDField;
    qryGroupItems: TADOQuery;
```

```

qryPRMerek: TWideStringField;
procedure btnPRClick(Sender: TObject);
procedure btnCancelClick(Sender: TObject);
procedure RefreshDtlSPK;
procedure FormShow(Sender: TObject);
procedure grdRekPRCheckBoxClick(Sender: TObject; ACol, ARow: Integer;
  State: Boolean);
procedure grdRekPRGetCellColor(Sender: TObject; ARow, ACol: Integer;
  AState: TGridDrawState; ABrush: TBrush; AFont: TFont);
procedure grdRekPRClickCell(Sender: TObject; ARow, ACol: Integer);
procedure btnBuatPOClick(Sender: TObject);
procedure btnConfirmClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  frmRekomendasiPR: TfrmRekomendasiPR;
implementation
uses FormComponent, FormInputPO, FormDaftarPO;
var temp:string;
{$R *.dfm}
procedure TfrmRekomendasiPR.btnBuatPOClick(Sender: TObject);
var
  i,j,NoUrut: integer;
  state: boolean;
  bookmark:string;
begin
  with frmInputPO do
  begin
    BersihkanTabelTemp;
    OpenQryTemp;
    grdRekPR.BeginUpdate;
    qryPR.DisableControls;
    j := 0;
    NoUrut := 1;
    qryPR.First;
    for i := 1 to grdRekPR.RowCount - 1 do
    begin
      if grdRekPR.GetCheckBoxState(1,i,state) then
      begin
        if state then
        begin
          qryPR.MoveBy(i - 1 - j);
          qryDtl.Insert;
          qryDtl.FieldName('IDSystemUser').AsString :=
frmComponent.lblUserName.Caption;

```

```

qryDtl.FieldByName('NoUrut').AsInteger := NoUrut;
NoUrut := NoUrut + 1;
qryGroupItems.Close;
qryGroupItems.Parameters[0].Value := qryPR.FieldByName('ItemsCode').AsString;
qryGroupItems.Open;
qryDtl.FieldByName('ItemsGroupCode').AsString :=
qryGroupItems.FieldByName('ItemsGroupCode').AsString;
qryDtl.FieldByName('ItemsGroup').AsString :=
qryGroupItems.FieldByName('ItemsGroup').AsString;
qryDtl.FieldByName('ItemsCode').AsString :=
qryPR.FieldByName('ItemsCode').AsString;
qryDtl.FieldByName('ItemsName').AsString :=
qryPR.FieldByName('ItemsName').AsString;
qryDtl.FieldByName('Merek').AsString := qryPR.FieldByName('Merek').AsString;
qryDtl.FieldByName('Jumlah').AsFloat := qryPR.FieldByName('JmlPR').AsFloat;
qryDtl.FieldByName('JmlPR').AsFloat := qryPR.FieldByName('JmlPR').AsFloat;
qryDtl.Post;
j := i - 1;
end;
end;
end;
qryPR.EnableControls;
grdRekPR.EndUpdate;
txtJenisPO.Caption := 'PR';
txtStatus.Caption := 'NEW';
frmInputPO.ShowModal;
end;
end;
procedure TfrmRekomendasiPR.btnCancelClick(Sender: TObject);
begin
qryPR.Close;
qryDtlSPK.Close;
Close;
end;
procedure TfrmRekomendasiPR.btnConfirmClick(Sender: TObject);
begin
frmDaftarPO.ShowModal;
end;
procedure TfrmRekomendasiPR.btnPRClick(Sender: TObject);
begin
temp := 'SELECT dbo.Tbl_MstItems.ItemsCode, dbo.Tbl_MstItems.ItemsName, ' +
'dbo.Tbl_MstItems.UOM, dbo.Tbl_MstItems.Merek, ' +
'SUM(dbo.Tbl_DtlSPK.SelisihPemakaian) AS JmlButuh, ' +
'dbo.Tbl_MstItems.QtyStock, dbo.Tbl_MstItems.QtyOnOrder, ' +
'SUM(dbo.Tbl_DtlSPK.SelisihPemakaian) - dbo.Tbl_MstItems.QtyStock - ' +
'dbo.Tbl_MstItems.QtyOnOrder AS JmlPR FROM dbo.Tbl_MstItems INNER JOIN '
+
'dbo.Tbl_DtlSPK ON (dbo.Tbl_MstItems.ItemsCode = dbo.Tbl_DtlSPK.' +

```

```

'ItemsCode) AND (dbo.Tbl_MstItems.Merek = dbo.Tbl_DtlSPK.' +
'Merek) INNER JOIN dbo.Tbl_MsSPK ON dbo.Tbl_DtlSPK.NoSPK = dbo.' +
'Tbl_MsSPK.NoSPK WHERE (dbo.Tbl_MsSPK.StatusSPK = N' +
QuotedStr('CONFIRM') + ') AND (dbo.Tbl_MstItems.ItemsGroupCode = N' +
QuotedStr('SP') + ') GROUP BY dbo.Tbl_MstItems.ItemsCode, dbo.' +
'Tbl_MstItems.ItemsName, dbo.Tbl_MstItems.Merek, dbo.Tbl_MstItems.UOM,
dbo.Tbl_MstItems.' +
'QtyStock, dbo.Tbl_MstItems.QtyOnOrder HAVING (SUM(dbo.Tbl_DtlSPK.' +
'SelisihPemakaian) - dbo.Tbl_MstItems.QtyStock - dbo.Tbl_MstItems.' +
'QtyOnOrder <> 0)');
with qryPR do
begin
  Close;
  SQL.Clear;
  SQL.Add(temp);
  Open;
end;
RefreshDtlSPK;
end;
procedure TfrmRekomendasiPR.grdRekPRCheckBoxClick(Sender: TObject; ACol,
  ARow: Integer; State: Boolean);
begin
  grdRekPR.RepaintRow(ARow);
end;
procedure TfrmRekomendasiPR.grdRekPRClickCell(Sender: TObject; ARow,
  ACol: Integer);
begin
  RefreshDtlSPK;
end;
procedure TfrmRekomendasiPR.grdRekPRGetCellColor(Sender: TObject; ARow,
  ACol: Integer; AState: TGridDrawState; ABrush: TBrush; AFont: TFont);
var
  state: boolean;
begin
  if grdRekPR.GetCheckBoxState(1,ARow,state) then
  begin
    if state then
    begin
      ABrush.Color := clRed;
      AFont.Color := clWhite;
      AFont.Style := [fsBold];
    end
    else
    begin
      ABrush.Color := clWindow;
      AFont.Color := clWindowText;
      AFont.Style := [];
    end;
  end;
end;

```

```
end;
end;
procedure TfrmRekomendasiPR.FormShow(Sender: TObject);
begin
    dtpAwal.Date := Date - StrToInt(FormatDateTime('dd',Date)) + 1;
    dtpAkhir.Date := Date;
    btnPR.Click;
end;
procedure TfrmRekomendasiPR.RefreshDtlSPK;
begin
    with qryDtlSPK do
        begin
            Parameters[0].Value := qryPR.Fields[0].Value;
            Parameters[1].Value := qryPR.Fields[2].Value;
            Close; Open; end;end;end.
```

## **FormInputPO.pas**

```
unit FormInputPO;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
  Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ComCtrls, Vcl.Buttons,
  Vcl.Grids, AdvObj, BaseGrid, AdvGrid, DBAdvGrid, Data.DB,
  Data.Win.ADODB, AnsiStrings,
  TaskDialog, frxClass, frxDBSet;
type
  TfrmInputPO = class(TForm)
    GroupBox1: TGroupBox;
    Label1: TLabel;
    Label2: TLabel;
    Label15: TLabel;
    Label16: TLabel;
    Label17: TLabel;
    lblStatus: TLabel;
    cbSupplierName: TComboBox;
    dtpTgl: TDateTimePicker;
    cbSupplierCode: TComboBox;
    txtAlamat: TStaticText;
    txtTelepon: TStaticText;
    grpDetail: TGroupBox;
    Label24: TLabel;
    Label25: TLabel;
    Label26: TLabel;
    Label28: TLabel;
    Label30: TLabel;
    Label32: TLabel;
    lblStatusDtl: TLabel;
    Label3: TLabel;
    edtJml: TEdit;
    edtHarga: TEdit;
    cbItemsGroup: TComboBox;
    btnSaveDtl: TButton;
    btnEditDtl: TButton;
    btnDeleteDtl: TButton;
    edtKeteranganDtl: TEdit;
    edtMerek: TEdit;
    cbItemsName: TComboBox;
    txtSubTotal: TStaticText;
    cbItemsCode: TComboBox;
    grdDtl: TDBAdvGrid;
    GroupBox3: TGroupBox;
    btnClose: TSpeedButton;
    btnSave: TSpeedButton;
```



```

Label42: TLabel;
txtTotal: TStaticText;
btnSaveConfirm: TSpeedButton;
qrySQL: TADOQuery;
qryMs: TADOQuery;
dsDtl: TDataSource;
qryDtl: TADOQuery;
Label4: TLabel;
txtJenisPO: TStaticText;
edtKeteranganPO: TEdit;
Label5: TLabel;
cbItemsGroupCode: TComboBox;
qryDtlIDSysUser: TWideStringField;
qryDtlNoUrut: TSmallintField;
qryDtlItemsCode: TWideStringField;
qryDtlMerek: TWideStringField;
qryDtlJumlah: TBCDField;
qryDtlHarga: TBCDField;
qryDtlSubTotal: TBCDField;
qryDtlKeterangan: TWideStringField;
qryDtlItemsName: TWideStringField;
txtStatus: TStaticText;
txtNoPO: TStaticText;
btnTutupPO: TSpeedButton;
Label6: TLabel;
txtStatusPO: TStaticText;
qryDtlItemsGroupCode: TWideStringField;
qryDtlItemsGroup: TWideStringField;
cbAlamat: TComboBox;
cbTelepon: TComboBox;
qryDtlJmlPR: TBCDField;
cbMerek: TComboBox;
InputKetTutup: TAdvInputTaskDialog;
frxReportPO: TfrxReport;
frxDBDataset1: TfrxDBDataset;
qryDtlPO: TADOQuery;
procedure btnCloseClick(Sender: TObject);
procedure IsiComboBox;
procedure IsiComboItems;
procedure OpenQrySQL;
function CekValid : Boolean;
function CekValidAll : Boolean;
procedure FormShow(Sender: TObject);
procedure cbItemsGroupChange(Sender: TObject);
procedure cbItemsCodeChange(Sender: TObject);
procedure cbItemsNameChange(Sender: TObject);
procedure BersihkanTabelTemp;
procedure BersihkanPO;

```

```

procedure BersihkanDtlPO;
procedure StatusReadOnly(Status:Boolean);
procedure btnSaveDtlClick(Sender: TObject);
procedure btnDeleteDtlClick(Sender: TObject);
procedure OpenQryTemp;
procedure btnEditDtlClick(Sender: TObject);
procedure btnSaveClick(Sender: TObject);
procedure SaveSeluruh;
procedure BuatNoPO;
procedure btnSaveConfirmClick(Sender: TObject);
procedure btnTutupPOClick(Sender: TObject);
procedure edtHargaChange(Sender: TObject);
procedure edtJmlChange(Sender: TObject);
procedure UpdateQtyOrderTambah;
procedure cbSupplierNameChange(Sender: TObject);
function ValidJmlPR:boolean;
procedure ViewPO;
procedure IsiComboMerek;
procedure TutupPO;
procedure edtJmlKeyPress(Sender: TObject; var Key: Char);
procedure edtHargaKeyPress(Sender: TObject; var Key: Char);
procedure PrintPO;
private
  { Private declarations }
public
  { Public declarations }
end;
var
  frmInputPO: TfrmInputPO;
implementation
uses FormComponent,GlobalUnit, FormDaftarPO;
var temp, NoPO, KeteranganTutup: string;
    NoUrut,i:integer;
    Total:Currency;
    TotSelisih,JmlPR:Single;
{$R *.dfm}
procedure TfrmInputPO.BersihkanDtlPO;
begin
  cbItemsGroup.ItemIndex := -1;
  cbItemsCode.Items.Clear;
  cbItemsName.Items.Clear;
  cbMerek.Items.Clear;
  edtMerek.Clear;
  edtJml.Clear;
  edtHarga.Clear;
  txtSubTotal.Caption := '0';
  edtKeteranganDtl.Clear;
end;

```

```

function TfrmInputPO.CekValidAll;
begin
    CekValidAll := True;
    if (cbSupplierName.Text = "") OR (cbSupplierName.ItemIndex = -1) then
    begin
        ShowMessage('Supplier harus diisi !');
        cbSupplierName.SetFocus;
        CekValidAll := False;
    end
    else if qryDtl.IsEmpty then
    begin
        ShowMessage('Barang tidak boleh kosong !');
        CekValidAll := False;
    end
    else if Total <= 0 then
    begin
        ShowMessage('Total tidak boleh kosong !');
        CekValidAll := False;
    end
    else if txtTotal.Caption = " " then
    begin
        ShowMessage('Total tidak boleh kosong !');
        CekValidAll := False;
    end
end;
function TfrmInputPO.CekValid : Boolean;
begin
    CekValid := True;
    if (cbItemsGroup.Text = "") or (cbItemsGroup.Code.Text = "") then
    begin
        ShowMessage('Items required!');
        cbItemsGroup.SetFocus;
        CekValid := False;
    end
    else if cbItemsCode.Text = " " then
    begin
        ShowMessage('Code Items required!');
        cbItemsCode.SetFocus;
        CekValid := False;
    end
    else if cbItemsName.Text = " " then
    begin
        ShowMessage('Items Name required!');
        cbItemsName.SetFocus;
        CekValid := False;
    end
    else if cbMerek.Text = " " then
    begin

```

```

    ShowMessage('Merk required!');
    cbMerek.SetFocus;
    CekValid := False;
end
else if edtJml.Text = '' then
begin
    ShowMessage('Jumlah required!');
    edtJml.SetFocus;
    CekValid := False;
end
else if edtHarga.Text = '' then
begin
    ShowMessage('Harga required!');
    edtHarga.SetFocus;
    CekValid := False;
end
else if StrToInt(edtJml.Text) <= 0 then
begin
    ShowMessage('Jumlah tidak boleh nol!');
    edtJml.SetFocus;
    CekValid := False;
end
else if StrToCurr(edtHarga.Text) <= 0 then
begin
    ShowMessage('Harga tidak boleh nol!');
    edtHarga.SetFocus;
    CekValid := False;
end;
ValidJmlPR;
end;
procedure TfrmInputPO.BersihkanPO;
begin
    txtNoPO.Caption := 'PO BARU';
    dtpTgl.Date := Date;
    cbSupplierName.ItemIndex := -1;
    cbSupplierCode.ItemIndex := -1;
    cbAlamat.ItemIndex := -1;
    cbTelepon.ItemIndex := -1;
    txtAlamat.Caption := '-';
    txtTelepon.Caption := '-';
    edtKeteranganPO.Clear;
    Total := 0;
    txtTotal.Caption := '0';
    BersihkanDtIPO;
end;
procedure TfrmInputPO.BersihkanTabelTemp;
begin
    temp := 'DELETE FROM Tbl_TempDtIPO WHERE IDSystemUser = ' +

```

```

        QuotedStr(frmComponent.lblUserName.Caption);
    with qrySQL do
    begin
        Close;
        SQL.Clear;
        SQL.Add(temp);
        ExecSQL;
    end;
end;
procedure TfrmInputPO.btnCloseClick(Sender: TObject);
begin
    Close;
end;
procedure TfrmInputPO.btnDeleteDtlClick(Sender: TObject);
var Jawab : integer;
begin
    if not(qryDtl.IsEmpty) then
    begin
        Jawab := MessageDlgPos('Items ini mau
dihapus?',mtCustom,mbOKCancel,0,Left+300,Top+430);
        if Jawab = mrOK then
        begin
            Total := Total - qryDtl.FieldByName('SubTotal').AsFloat;
            txtTotal.Caption := FloatToStrF(Total,ffNumber,10,2);
            qryDtl.Delete;
        end;
        OpenQryTemp;
    end;
end;
procedure TfrmInputPO.btnEditDtlClick(Sender: TObject);
begin
    if not(qryDtl.IsEmpty) then
    with qryDtl do
    begin
        NoUrut := FieldByName('NoUrut').AsInteger;
        edtJml.Text := FieldByName('Jumlah').AsString;
        edtKeteranganDtl.Text := FieldByName('Keterangan').AsString;
        edtHarga.Text := FieldByName('Harga').AsString;
        JmlPR := FieldByName('JmlPR').AsFloat;
        Total := Total - FieldByName('SubTotal').AsFloat;
        txtTotal.Caption := FloatToStrF(Total,ffNumber,10,2);
        i := 0;
        repeat
            cbItemsGroupCode.ItemIndex := i;
            i := i + 1;
        until (cbItemsGroupCode.Text = FieldByName('ItemsGroupCode').AsString) or (i >
cbItemsGroupCode.Items.Count);
        if (i > cbItemsGroupCode.Items.Count) then

```

```

    cbItemsGroupCode.ItemIndex := -1;
    cbItemsGroup.ItemIndex := cbItemsGroupCode.ItemIndex;
    IsiComboItems;
    i := 0;
    repeat
        cbItemsCode.ItemIndex := i;
        i := i + 1;
    until (cbItemsCode.Text = FieldByName('ItemsCode').AsString) or (i >
cbItemsCode.Items.Count);
    if (i > cbItemsCode.Items.Count) then
        cbItemsCode.ItemIndex := -1;
    cbItemsName.ItemIndex := cbItemsCode.ItemIndex;
    IsiComboMerek;
    i := 0;
    repeat
        cbMerek.ItemIndex := i;
        i := i + 1;
    until (cbMerek.Text = FieldByName('Merek').AsString) or (i > cbMerek.Items.Count);
    if (i > cbMerek.Items.Count) then
        cbMerek.ItemIndex := -1;
    qryDtl.Delete;
    btnEditDtl.Enabled := False;
    btnDeleteDtl.Enabled := False;
    grdDtl.Enabled := False;
    btnSaveDtl.Enabled := True;
    end;
end;
procedure TfrmInputPO.btnSaveClick(Sender: TObject);
begin
    if CekValidAll then
        begin
            SaveSeluruh;
        end
    end;
end;
procedure TfrmInputPO.PrintPO;
var
    Memo: TfrxMemoView;
    Component: TfrxComponent;
begin
    Component := frxReportPO.FindObject('nop0');
    Memo := Component as TfrxMemoView;
    Memo.Text := txtNoPO.Caption;
    Component := frxReportPO.FindObject('tanggal');
    Memo := Component as TfrxMemoView;
    Memo.Text := DateToStr(dtpTgl.Date);
    Component := frxReportPO.FindObject('namasupplier');
    Memo := Component as TfrxMemoView;
    Memo.Text := cbSupplierName.Text;
end;

```

```

Component := frxReportPO.FindObject('suppliername');
Memo := Component as TfrxMemoView;
Memo.Text := cbSupplierName.Text;
Component := frxReportPO.FindObject('telepon');
Memo := Component as TfrxMemoView;
Memo.Text := cbTelepon.Text;

temp:='SELECT * FROM Tbl_MstSupplier WHERE SupplierCode = ' +
QuotedStr(cbSupplierCode.Text);

with qrySQL do
begin
  OpenQrySQL;
end;
Component := frxReportPO.FindObject('contactperson');
Memo := Component as TfrxMemoView;
Memo.Text := qrySQL.FieldByName('ContactPerson').AsString;
Component := frxReportPO.FindObject('personcontact');
Memo := Component as TfrxMemoView;
Memo.Text := qrySQL.FieldByName('ContactPerson').AsString;
Component := frxReportPO.FindObject('total');
Memo := Component as TfrxMemoView;
Memo.Text := txtTotal.Caption;
Component := frxReportPO.FindObject('username');
Memo := Component as TfrxMemoView;
Memo.Text := frmComponent.lblUserName.Caption;
frxReportPO.ShowReport;
end;
procedure TfrmInputPO.btnSaveConfirmClick(Sender: TObject);
begin
  if not frmComponent.adoTerra.InTransaction then
  with qryMs do
  begin
    frmComponent.adoTerra.BeginTrans;
    try
      temp := 'UPDATE Tbl_MsPO SET StatusPO = ' + QuotedStr('CONFIRM') +
        ', IDSystemConfirm = ' + QuotedStr(frmComponent.lblUserName.Caption ) +
        ', ConfirmTime = ' + QuotedStr(FormatDateTime('yyyy-mm-dd',Date) + '' +
FormatDateTime('hh:mm',Time)) +
        ' WHERE NoPO = ' + QuotedStr(NoPO);
      Close;
      SQL.Clear;
      SQL.Add(temp);
      ExecSQL;
      qryDtl.First;
      while not(qryDtl.Eof) do
      begin
        UpdateQtyOrderTambah;
        qryDtl.Next;
      end;
    end;
  end;
end;

```

```

frmComponent.adoTerra.CommitTrans;
btnSaveConfirm.Enabled := False;
ShowMessage('PO CONFIRM');
with qryDtlPO do
begin
  Close;
  Parameters[0].Value := txtNoPO.Caption;
  Open;
end;
PrintPO;
Except
  frmComponent.adoTerra.RollbackTrans;
end; // end try
end; //end if ado.InTrans / with qry
end;
procedure TfrmInputPO.btnSaveDtlClick(Sender: TObject);
begin
  if CekValid then
  with qryDtl do
  begin
    Close;
    Open;
    if {(lblStatusDtl.Caption = 'NEW') and } (not(IsEmpty)) then
    begin
      Last;
      NoUrut := FieldByName('NoUrut').AsInteger + 1;
    end;
    if {(lblStatusDtl.Caption = 'NEW') and } (IsEmpty) then
      NoUrut := 1;
    Insert;
    FieldByName('IDSystemUser').AsString := frmComponent.lblUserName.Caption;
    FieldByName('NoUrut').AsInteger := NoUrut;
    FieldByName('ItemsGroupCode').AsString := cbItemsGroupCode.Text;
    FieldByName('ItemsGroup').AsString := cbItemsGroup.Text;
    FieldByName('ItemsCode').AsString := cbItemsCode.Text;
    FieldByName('ItemsName').AsString := cbItemsName.Text;
    //FieldByName('Merek').AsString := edtMerek.Text;
    FieldByName('Merek').AsString := cbMerek.Text;
    FieldByName('Jumlah').AsFloat := StrToFloat(edtJml.Text);
    FieldByName('JmlPR').AsFloat := JmlPR;
    FieldByName('Harga').AsCurrency := StrToCurr(edtHarga.Text);
    FieldByName('Keterangan').AsString := edtKeteranganDtl.Text;
    Post;
    Total := Total + StrToFloat(edtJml.Text) * StrToCurr(edtHarga.Text);
    txtTotal.Caption := FloatToStrF(Total,ffNumber,10,2);
    Close;
    Open;
    BersihkanDtlPO;
  end;
end;

```



```

    btnEditDtl.Enabled := True;
    btnDeleteDtl.Enabled := True;
    grdDtl.Enabled := True;
    OpenQryTemp;
end
end;
procedure TfrmInputPO.btnTutupPOClick(Sender: TObject);
var Jawab : integer;
begin
    if TotSelisih <> 0 then
    begin
        Jawab := MessageDlgPos('PO ini benar mau ditutup? Ada Barang yang belum diterima
!',mtCustom,mbOKCancel,0,Left+300,Top+430);
        if Jawab = mrOK then
            TutupPO
        else
            ShowMessage('Masih ada Items yang belum diterima !');
        end
    else
        TutupPO;
    end;
    procedure TfrmInputPO.BuatNoPO;
    var TempNoPO : integer;
    begin
        temp := 'SELECT NoPO FROM Tbl_MsPO ORDER BY NoPO';
        OpenQrySQL;
        if not(qrySQL.IsEmpty) then
        begin
            qrySQL.Last;
            TempNoPO := StrToInt(RightStr(qrySQL.FieldByName('NoPO').AsString,4)) + 1;
        end
        else
            TempNoPO := 1;
        NoPO := 'PO-' + FormatDateTime('yy',Date) + TambahNolRibuan(TempNoPO);
    end;
    procedure TfrmInputPO.cbItemsCodeChange(Sender: TObject);
    begin
        cbItemsName.ItemIndex := cbItemsCode.ItemIndex;
        IsiComboMerek;
    end;
    procedure TfrmInputPO.cbItemsGroupChange(Sender: TObject);
    begin
        cbItemsGroupCode.ItemIndex := cbItemsGroup.ItemIndex;
        IsiComboItems;
    end;
    procedure TfrmInputPO.cbItemsNameChange(Sender: TObject);
    begin
        cbItemsCode.ItemIndex := cbItemsName.ItemIndex;

```

```

    IsiComboMerek;end;
procedure TfrmInputPO.cbSupplierNameChange(Sender: TObject);
begin
    cbSupplierCode.ItemIndex := cbSupplierName.ItemIndex;
    cbTelepon.ItemIndex := cbSupplierName.ItemIndex;
    cbAlamat.ItemIndex := cbSupplierName.ItemIndex;
    txtTelepon.Caption := cbTelepon.Text;
    txtAlamat.Caption := cbAlamat.Text;
end;
procedure TfrmInputPO.edtHargaChange(Sender: TObject);
begin
    if (edtJml.Text <> "") and (edtHarga.Text <> "") then
        txtSubTotal.Caption :=
FloatToStrF(StrToFloat(edtJml.Text)*StrToCurr(edtHarga.Text),ffNumber,10,2);
end;
procedure TfrmInputPO.edtHargaKeyPress(Sender: TObject; var Key: Char);
begin
    if not (Key in ['0'..'9','!','#27,#13,#8,#9]) then
        begin
            edtHarga.Clear;
            edtHarga.SetFocus;
            Key := #0;
        end
    end;
procedure TfrmInputPO.edtJmlChange(Sender: TObject);
begin
    if (edtJml.Text <> "") and (edtHarga.Text <> "") then
        txtSubTotal.Caption :=
FloatToStrF(StrToFloat(edtJml.Text)*StrToCurr(edtHarga.Text),ffNumber,10,2);
end;
procedure TfrmInputPO.edtJmlKeyPress(Sender: TObject; var Key: Char);
begin
    if not (Key in ['0'..'9','!','#27,#13,#8,#9]) then
        begin
            edtJml.Clear;
            edtJml.SetFocus;
            Key := #0;
        end
    end;
procedure TfrmInputPO.FormShow(Sender: TObject);
begin
    if txtJenisPO.Caption <> 'PR' then
        BersihkanTabelTemp;
    IsiComboBox;
    if txtStatus.Caption = 'NEW' then
        begin
            NoPO := 'NEW';
            BersihkanPO;
        end
    end;

```

```

StatusReadOnly(FALSE);
btnSave.Enabled := True;
btnSaveConfirm.Enabled := False;
btnTutupPO.Enabled := False;
txtStatusPO.Caption := 'OPEN';
end;
if txtStatus.Caption = 'CONFIRM' then
begin
    NoPO := frmDaftarPO.qryMs.FieldByName('NoPO').AsString;
    ViewPO;
    StatusReadOnly(TRUE);
    btnSaveConfirm.Enabled := True;
    btnSave.Enabled := False;
    btnTutupPO.Enabled := False;
    txtStatusPO.Caption := 'TO CONFIRM';
end;
if txtStatus.Caption = 'CLOSE' then
begin
    NoPO := frmDaftarPO.qryMs.FieldByName('NoPO').AsString;
    ViewPO;
    StatusReadOnly(TRUE);
    btnSaveConfirm.Enabled := False;
    btnSave.Enabled := False;
    btnTutupPO.Enabled := True;
    txtStatusPO.Caption := 'TO CLOSE';
end;
if txtStatus.Caption = 'VIEW' then
begin
    NoPO := frmDaftarPO.qryMs.FieldByName('NoPO').AsString;
    ViewPO;
    StatusReadOnly(TRUE);
    btnSaveConfirm.Enabled := False;
    btnSave.Enabled := False;
    btnTutupPO.Enabled := False;
    txtStatusPO.Caption := frmDaftarPO.qryMs.FieldByName('StatusPO').AsString;
end;
if txtStatus.Caption = 'EDIT' then
begin
    NoPO := frmDaftarPO.qryMs.FieldByName('NoPO').AsString;
    txtStatusPO.Caption := frmDaftarPO.qryMs.FieldByName('StatusPO').AsString;
    ViewPO;
    StatusReadOnly(FALSE);
    btnSave.Enabled := True;
    btnSaveConfirm.Enabled := False;
    btnTutupPO.Enabled := False;
end;
txtNoPO.Caption := NoPO;
lblStatusDtl.Caption := 'NEW';

```

```

    OpenQryTemp;
end;
procedure TfrmInputPO.IsiComboBox;
begin
    temp := 'SELECT * FROM Tbl_MstSupplier ORDER BY SupplierName';
    OpenQrySQL;
    cbSupplierCode.Items.Clear;
    cbSupplierName.Items.Clear;
    cbAlamat.Items.Clear;
    cbTelepon.Items.Clear;
    while not(qrySQL.Eof) do
    begin
        cbSupplierCode.Items.Add(qrySQL.FieldByName('SupplierCode').AsString);
        cbSupplierName.Items.Add(qrySQL.FieldByName('SupplierName').AsString);
        cbAlamat.Items.Add(qrySQL.FieldByName('SupplierAddress').AsString);
        cbTelepon.Items.Add(qrySQL.FieldByName('Phone').AsString);
        qrySQL.Next;
    end;
    temp := 'SELECT ItemsGroupCode, ItemsGroup FROM Tbl_MstItemsGroup ORDER BY
ItemsGroup';
    OpenQrySQL;
    cbItemsGroupCode.Items.Clear;
    cbItemsGroup.Items.Clear;
    while not(qrySQL.Eof) do
    begin
        cbItemsGroupCode.Items.Add(qrySQL.FieldByName('ItemsGroupCode').AsString);
        cbItemsGroup.Items.Add(qrySQL.FieldByName('ItemsGroup').AsString);
        qrySQL.Next;
    end;end;
procedure TfrmInputPO.IsiComboItems;
begin
    temp := 'SELECT DISTINCT ItemsCode, ItemsName FROM Tbl_MstItems '+
        'WHERE ItemsGroupCode = ' + QuotedStr(cbItemsGroupCode.Text) +
        ' ORDER BY ItemsCode';
    OpenQrySQL;
    cbItemsCode.Items.Clear;
    cbItemsName.Items.Clear;
    while not(qrySQL.Eof) do
    begin
        cbItemsCode.Items.Add(qrySQL.FieldByName('ItemsCode').AsString);
        cbItemsName.Items.Add(qrySQL.FieldByName('ItemsName').AsString);
        qrySQL.Next;
    end;end;
procedure TfrmInputPO.IsiComboMerek;
begin
    temp := 'SELECT Merek FROM Tbl_MstItems '+
        'WHERE ItemsCode = ' + QuotedStr(cbItemsCode.Text) +
        ' ORDER BY Merek';

```

```

OpenQrySQL;
cbMerek.Items.Clear;
while not(qrySQL.Eof) do
begin
    cbMerek.Items.Add(qrySQL.FieldByName('Merek').AsString);
    qrySQL.Next;
end;end;
procedure TfrmInputPO.OpenQrySQL;
begin
    with qrySQL do
    begin
        Close;
        SQL.Clear;
        SQL.Add(temp);
        Open;
    end;end;
procedure TfrmInputPO.OpenQryTemp;
begin
    temp := 'SELECT * FROM Tbl_TempDtIPO WHERE IDSystemUser = ' +
        QuotedStr(frmComponent.lblUserName.Caption);
    with qryDtl do
    begin
        Close;
        SQL.Clear;
        SQL.Add(temp);
        Open;
        if not(IsEmpty) then
        begin
            btnEditDtl.Enabled := True;
            btnDeleteDtl.Enabled := True;
        end
        else
        begin
            btnEditDtl.Enabled := False;
            btnDeleteDtl.Enabled := False;
        end;
        if txtJenisPO.Caption = 'PR' then
            btnSaveDtl.Enabled := False
        else
            btnSaveDtl.Enabled := True;
    end;end;
procedure TfrmInputPO.SaveSeluruh;
begin
    temp := 'SELECT * FROM Tbl_MsPO WHERE NoPO = ' + QuotedStr(NoPO);
    with qryMs do
    begin
        Close;
        SQL.Clear;

```

```

SQL.Add(temp);
Open;
if not frmComponent.adoTerra.InTransaction then
begin
    frmComponent.adoTerra.BeginTrans;
    try
        If txtStatus.Caption = 'NEW' then
        begin
            BuatNoPO;
            Insert;
        end
        else
            Edit;

        FieldByName('NoPO').AsString := NoPO;
        FieldByName('StatusPO').AsString := 'OPEN';
        FieldByName('TglIPO').AsDateTime := dtpTgl.Date;
        FieldByName('SupplierCode').AsString := cbSupplierCode.Text;
        FieldByName('JenisPO').AsString := txtJenisPO.Caption;
        FieldByName('Total').AsCurrency := Total;
        FieldByName('TotalBayar').AsCurrency := 0;
        FieldByName('Keterangan').AsString := edtKeteranganPO.Text;
        FieldByName('IDSystemUser').AsString := frmComponent.lblUserName.Caption;
        FieldByName('UpdateTime').AsDateTime := Date;
        Post;
        Refresh;
        if txtStatus.Caption = 'EDIT' then
        with qrySQL do
        begin
            temp := 'DELETE FROM Tbl_DtlPO WHERE NoPO = ' + QuotedStr(NoPO);
            Close;
            SQL.Clear;
            SQL.Add(temp);
            ExecSQL;
        end;
        temp := 'SELECT * FROM Tbl_DtlPO WHERE NoPO = ' + QuotedStr(NoPO);
        OpenQrySQL;
        qryDtl.First;
        while not(qryDtl.Eof) do
        begin
            qrySQL.Insert;
            qrySQL.FieldByName('NoPO').AsString := NoPO;
            qrySQL.FieldByName('NoUrut').AsInteger :=
qryDtl.FieldByName('NoUrut').AsInteger;
            qrySQL.FieldByName('ItemsCode').AsString
:=qryDtl.FieldByName('ItemsCode').AsString;
            qrySQL.FieldByName('Merek').AsString
:=qryDtl.FieldByName('Merek').AsString;

```

```

        qrySQL.FieldByName('Jumlah').AsFloat :=
qryDtl.FieldByName('Jumlah').AsFloat;
        qrySQL.FieldByName('Harga').AsCurrency :=
qryDtl.FieldByName('Harga').AsCurrency;
        qrySQL.FieldByName('Keterangan').AsString
:=qryDtl.FieldByName('Keterangan').AsString;
        qrySQL.Post;
        //UpdateQtyOrder;
        qryDtl.Next;
    end;
    frmComponent.adoTerra.CommitTrans;
    ShowMessage('SAVE FINISH');
    btnSave.Enabled := False;
Except
    frmComponent.adoTerra.RollbackTrans;
end end end; end;

procedure TfrmInputPO.StatusReadOnly(Status: Boolean);
begin
    cbSupplierName.Enabled := Not(Status);
    cbSupplierCode.Enabled := Not(Status);
    dtpTgl.Enabled := Not(Status);
    edtKeteranganPO.Enabled := Not(Status);
    cbItemsGroup.Enabled := Not(Status);
    cbItemsCode.Enabled := Not(Status);
    cbItemsName.Enabled := Not(Status);
    edtMerek.ReadOnly := Status;
    edtJml.ReadOnly := Status;
    edtHarga.ReadOnly := Status;
    edtKeteranganDtl.ReadOnly := Status;
    grpDetail.Enabled := Not(Status);
end;

procedure TfrmInputPO.TutupPO;
var TotTerima : String;
begin
    if not frmComponent.adoTerra.InTransaction then
    with qryMs do
    begin
        frmComponent.adoTerra.BeginTrans;
        try
            InputKetTutup.Execute;
            temp := 'SELECT SUM(SubTotalTerima) AS TotTerima FROM Tbl_DtlPO' +
                ' WHERE NoPO = ' + QuotedStr(NoPO);
            OpenQrySQL;
            TotTerima := qrySQL.FieldByName('TotTerima').AsString;
            temp := 'UPDATE Tbl_MsPO SET StatusPO = ' + QuotedStr('CLOSE') +
                ', TotalTerima = ' + TotTerima +
                ', KeteranganTutup = ' + QuotedStr(InputKetTutup.InputText) +
                ', IDSystemClose = ' + QuotedStr(frmComponent.IblUserName.Caption) +

```

```

        ', CloseTime = ' + QuotedStr(FormatDateTime('yyyy-mm-dd',Date) + ' ' +
FormatDateTime('hh:mm',Time)) + ' WHERE NoPO = ' + QuotedStr(NoPO);
    Close;
    SQL.Clear;
    SQL.Add(temp);
    ExecSQL;
    frmComponent.adoTerra.CommitTrans;
    btnTutupPO.Enabled := False;
    ShowMessage('PO CLOSE');
Except
    frmComponent.adoTerra.RollbackTrans;
end; // end try
end;end;
procedure TfrmInputPO.UpdateQtyOrderTambah;
begin
    temp := 'UPDATE Tbl_MstItems SET QtyOnOrder = QtyOnOrder + ' +
        qryDtl.FieldByName('Jumlah').AsString + ' WHERE ItemsCode = ' +
        QuotedStr(qryDtl.FieldByName('ItemsCode').AsString) +
        ' AND Merek = ' + QuotedStr(qryDtl.FieldByName('Merek').AsString);
    with qrySQL do
    begin
        Close;
        SQL.Clear;
        SQL.Add(temp);
        ExecSQL;
    end;
end;
function TfrmInputPO.ValidJmlPR: boolean;
begin
    if (txtJenisPO.Caption = 'PR') and (StrToFloat(edtJml.Text) <= JmlPR) then
        ValidJmlPR := True
    else
        ValidJmlPR := False;
    if (txtJenisPO.Caption <> 'PR') then
        ValidJmlPR := True;
end;
procedure TfrmInputPO.ViewPO;
begin
    with frmDaftarPO.qryMs do
    begin
        txtNoPO.Caption := NoPO;
        dtpTgl.Date := FieldByName('TglIPO').AsDateTime;
        i := 0;
        repeat
            cbSupplierCode.ItemIndex := i;
            i := i + 1;
        until (cbSupplierCode.Text = FieldByName('SupplierCode').AsString) or (i >
cbSupplierCode.Items.Count);
    end;
end;

```



```

if (i > cbSupplierCode.Items.Count) then
  cbSupplierCode.ItemIndex := -1;
cbSupplierName.ItemIndex := cbSupplierCode.ItemIndex;
cbAlamat.ItemIndex := cbSupplierCode.ItemIndex;
cbTelepon.ItemIndex := cbSupplierCode.ItemIndex;
txtTelepon.Caption := cbTelepon.Text;
txtAlamat.Caption := cbAlamat.Text;
txtJenisPO.Caption := FieldByName('JenisPO').AsString;
Total := FieldByName('Total').AsCurrency;
txtTotal.Caption := FloatToStrF(Total,ffNumber,10,2);
edtKeteranganPO.Text := FieldByName('Keterangan').AsString; end;
BersihkanTabelTemp; OpenQryTemp;
TotSelisih := 0;
frmDaftarPO.qryDtl.First;
while not(frmDaftarPO.qryDtl.Eof) do
begin
  qryDtl.Insert;
  qryDtl.FieldByName('IDSystemUser').AsString :=
frmComponent.lblUserName.Caption;
  qryDtl.FieldByName('NoUrut').AsInteger :=
frmDaftarPO.qryDtl.FieldByName('NoUrut').AsInteger;
  qryDtl.FieldByName('ItemsGroupCode').AsString
:=frmDaftarPO.qryDtl.FieldByName('ItemsGroupCode').AsString;
  qryDtl.FieldByName('ItemsGroup').AsString
:=frmDaftarPO.qryDtl.FieldByName('ItemsGroup').AsString;
  qryDtl.FieldByName('ItemsCode').AsString
:=frmDaftarPO.qryDtl.FieldByName('ItemsCode').AsString;
  qryDtl.FieldByName('ItemsName').AsString
:=frmDaftarPO.qryDtl.FieldByName('ItemsName').AsString;
  qryDtl.FieldByName('Merek').AsString
:=frmDaftarPO.qryDtl.FieldByName('Merek').AsString;
  qryDtl.FieldByName('Jumlah').AsFloat :=
frmDaftarPO.qryDtl.FieldByName('Jumlah').AsFloat;
  qryDtl.FieldByName('Harga').AsCurrency :=
frmDaftarPO.qryDtl.FieldByName('Harga').AsCurrency;
  qryDtl.FieldByName('Keterangan').AsString
:=frmDaftarPO.qryDtl.FieldByName('Keterangan').AsString;
  qryDtl.Post;
  TotSelisih := TotSelisih + frmDaftarPO.qryDtl.FieldByName('JumlahSelisih').AsFloat;
  frmDaftarPO.qryDtl.Next;
end;end;end.

```

## **FormDaftarPO.pas**

```
unit FormDaftarPO;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
  Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ComCtrls, Vcl.Buttons,
  Data.DB, Data.Win.ADODB, Vcl.Grids, AdvObj, BaseGrid, AdvGrid, DBAdvGrid,
  Vcl.ExtCtrls;
type
  TfrmDaftarPO = class(TForm)
    GroupBox1: TGroupBox;
    btnAdd: TSpeedButton;
    btnEditDraft: TSpeedButton;
    btnConfirm: TSpeedButton;
    btnRefresh: TSpeedButton;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    btnCancel: TSpeedButton;
    btnTutupSPK: TSpeedButton;
    dtpAwal: TDateTimePicker;
    cbStatus: TComboBox;
    dtpAkhir: TDateTimePicker;
    GroupBox2: TGroupBox;
    Splitter1: TSplitter;
    GroupBox3: TGroupBox;
    grdMs: TDBAdvGrid;
    grdDtI: TDBAdvGrid;
    dsMs: TDataSource;
    dsDtI: TDataSource;
    qryMs: TADOQuery;
    qryDtI: TADOQuery;
    chkPR: TCheckBox;
    qryMsNoPO: TWideStringField;
    qryMsJenisPO: TWideStringField;
    qryMsTglPO: TDateTimeField;
    qryMsSupplierCode: TWideStringField;
    qryMsTotal: TBCDField;
    qryMsStatusPO: TWideStringField;
    qryMsIDSystemUser: TWideStringField;
    qryMsUpdateTime: TDateTimeField;
    qryMsIDSystemConfirm: TWideStringField;
    qryMsConfirmTime: TDateTimeField;
    qryMsIDSystemClose: TWideStringField;
    qryMsCloseTime: TDateTimeField;
    qryMsKeterangan: TWideStringField;
    qryMsSupplierName: TWideStringField;
```

```

qryDtlNoPO: TWideStringField;
qryDtlNoUrut: TSmallintField;
qryDtlItemsCode: TWideStringField;
qryDtlMerek: TWideStringField;
qryDtlJumlah: TBCDField;
qryDtlJumlahDiterima: TBCDField;
qryDtlJumlahSelisih: TBCDField;
qryDtlHarga: TBCDField;
qryDtlSubTotal: TFMTBCDField;
qryDtlKeterangan: TWideStringField;
qryDtlItemsName: TWideStringField;
qryDtlUOM: TWideStringField;
qryDtlItemsGroupCode: TWideStringField;
qryDtlItemsGroup: TWideStringField;
btnTerima: TSpeedButton;
qryMsPhone: TWideStringField;
qryMsSupplierAddress: TWideStringField;
procedure btnRefreshClick(Sender: TObject);
procedure RefreshDtl;
procedure dtpAwalChange(Sender: TObject);
procedure dtpAkhirChange(Sender: TObject);
procedure cbStatusChange(Sender: TObject);
procedure chkPRClick(Sender: TObject);
procedure grdMsClick(Sender: TObject);
procedure btnCancelClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure btnAddClick(Sender: TObject);
procedure btnConfirmClick(Sender: TObject);
procedure btnTutupSPKClick(Sender: TObject);
procedure btnEditDraftClick(Sender: TObject);
procedure grdMsDbIclick(Sender: TObject);
procedure btnTerimaClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  frmDaftarPO: TfrmDaftarPO;
implementation
{$R *.dfm}
uses FormComponent, FormInputPO, FormReceiveItems, FormReceiveItemsTires;
procedure TfrmDaftarPO.btnAddClick(Sender: TObject);
begin
  frmInputPO.txtJenisPO.Caption := 'No PR';
  frmInputPO.txtStatus.Caption := 'NEW';
  frmInputPO.ShowModal;
end;

```

```

procedure TfrmDaftarPO.btnCancelClick(Sender: TObject);
begin
  qryMs.Close;
  qryDtl.Close;
  Close;end;
procedure TfrmDaftarPO.btnConfirmClick(Sender: TObject);
begin
  if not(qryMs.IsEmpty) then
  begin
    frmInputPO.txtStatus.Caption := 'CONFIRM';
    frmInputPO.ShowModal;
    btnRefresh.Click;
  end;end;
procedure TfrmDaftarPO.btnEditDraftClick(Sender: TObject);
begin
  if not(qryMs.IsEmpty) then
  begin
    frmInputPO.txtStatus.Caption := 'EDIT';
    frmInputPO.ShowModal;
    btnRefresh.Click;
  end;end;
procedure TfrmDaftarPO.btnRefreshClick(Sender: TObject);
begin
  if cbStatus.Text = 'OPEN' then
  begin
    btnEditDraft.Enabled := True;
    btnTutupSPK.Enabled := False;
    btnConfirm.Enabled := True;
    btnTerima.Enabled := False;
  end;
  if cbStatus.Text = 'CONFIRM' then
  begin
    btnEditDraft.Enabled := False;
    btnTutupSPK.Enabled := True;
    btnConfirm.Enabled := False;
    btnTerima.Enabled := True;
  end;
  if cbStatus.Text = 'CLOSE' then
  begin
    btnEditDraft.Enabled := False;
    btnTutupSPK.Enabled := False;
    btnConfirm.Enabled := False;
    btnTerima.Enabled := False;
  end;
  with qryMs do
  begin
    Close;
    Parameters[0].Value := cbStatus.Text;
  end;
end;

```

```

Parameters[1].Value := dtpAwal.Date;
Parameters[2].Value := dtpAkhir.Date;
if chkPR.Checked then
begin
  Parameters[3].Value := 'PR';
  btnEditDraft.Enabled := False;
end
else
begin
  Parameters[3].Value := 'No PR';
  if cbStatus.Text = 'OPEN' then
    btnEditDraft.Enabled := True
  else
    btnEditDraft.Enabled := False;
end; Open; end; RefreshDtl;end;
procedure TfrmDaftarPO.btnTerimaClick(Sender: TObject);
begin
  if (qryDtl.FieldByName('ItemsName').AsString <> 'Ban') then
  begin
    frmReceiveItems.txtNoPO.Caption := qryMs.FieldByName('NoPO').AsString;
    frmReceiveItems.lblNoUrut.Caption := qryDtl.FieldByName('NoUrut').AsString;
    frmReceiveItems.lblHarga.Caption := qryDtl.FieldByName('Harga').AsString;
    frmReceiveItems.txtKodeBarang.Caption := qryDtl.FieldByName('ItemsCode').AsString;
    frmReceiveItems.txtNamaBarang.Caption := qryDtl.FieldByName('ItemsName').AsString;
    frmReceiveItems.txtMerek.Caption := qryDtl.FieldByName('Merek').AsString;
    frmReceiveItems.txtJmlPO.Caption := qryDtl.FieldByName('Jumlah').AsString;
    frmReceiveItems.txtJmlSdhTerima.Caption :=
    qryDtl.FieldByName('JumlahDiterima').AsString;
    frmReceiveItems.edtJmlTerima.Text := qryDtl.FieldByName('JumlahSelisih').AsString;
    frmReceiveItems.ShowModal;
    RefreshDtl;
  end
  else
  begin
    frmReceiveItemsTires.txtNoPO.Caption := qryMs.FieldByName('NoPO').AsString;
    frmReceiveItemsTires.lblNoUrut.Caption := qryDtl.FieldByName('NoUrut').AsString;
    frmReceiveItemsTires.lblHarga.Caption := qryDtl.FieldByName('Harga').AsString;
    frmReceiveItemsTires.txtKodeBarang.Caption :=
    qryDtl.FieldByName('ItemsCode').AsString;
    frmReceiveItemsTires.txtNamaBarang.Caption :=
    qryDtl.FieldByName('ItemsName').AsString;
    frmReceiveItemsTires.txtMerek.Caption := qryDtl.FieldByName('Merek').AsString;
    frmReceiveItemsTires.txtJmlPO.Caption := qryDtl.FieldByName('Jumlah').AsString;
    frmReceiveItemsTires.txtJmlSdhTerima.Caption :=
    qryDtl.FieldByName('JumlahDiterima').AsString;
    frmReceiveItemsTires.ShowModal;
  end;end;
procedure TfrmDaftarPO.btnTutupSPKClick(Sender: TObject);

```

```

begin
  if not(qryMs.IsEmpty) then
    begin
      frmInputPO.txtStatus.Caption := 'CLOSE';
      frmInputPO.ShowModal;
      btnRefresh.Click;
    end;end;
procedure TfrmDaftarPO.cbStatusChange(Sender: TObject);
begin
  btnRefresh.Click;end;
procedure TfrmDaftarPO.chkPRClick(Sender: TObject);
begin
  btnRefresh.Click;end;
procedure TfrmDaftarPO.dtpAkhirChange(Sender: TObject);
begin
  if dtpAwal.Date > dtpAkhir.Date then
    dtpAwal.Date := dtpAkhir.Date;
  btnRefresh.Click;end;
procedure TfrmDaftarPO.dtpAwalChange(Sender: TObject);
begin
  if dtpAwal.Date > dtpAkhir.Date then
    dtpAwal.Date := dtpAkhir.Date;
  btnRefresh.Click;end;
procedure TfrmDaftarPO.FormShow(Sender: TObject);
begin
  dtpAwal.Date := Date - StrToInt(FormatDateTime('dd',Date)) + 1;
  dtpAkhir.Date := Date;
  cbStatus.ItemIndex := 0;
  chkPR.Checked := False;
  btnRefresh.Click;end;
procedure TfrmDaftarPO.grdMsClick(Sender: TObject);
begin
  RefreshDtl;end;
procedure TfrmDaftarPO.grdMsDbfClick(Sender: TObject);
begin
  if not(qryMs.IsEmpty) then
    begin
      frmInputPO.txtStatus.Caption := 'VIEW';  frmInputPO.ShowModal; end;end;
procedure TfrmDaftarPO.RefreshDtl;
begin
  with qryDtl do
    begin
      Close;  Parameters[0].Value := qryMs.FieldName('NoPO').AsString;
      Open;  end;end;end.

```

## **FormReceiveItems.pas**

```
unit FormReceiveItems;
interface
uses
  Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes,
  Vcl.Graphics,
  Vcl.Controls, Vcl.Forms, Vcl.Dialogs, Vcl.StdCtrls, Vcl.ComCtrls, Vcl.Buttons,
  Vcl.Grids, AdvObj, BaseGrid, AdvGrid, DBAdvGrid, Data.DB, Data.Win.ADODB;
type
  TfrmReceiveItems = class(TForm)
    GroupBox1: TGroupBox;
    DBAdvGrid1: TDBAdvGrid;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    btnTerima: TSpeedButton;
    txtNoPO: TStaticText;
    txtKodeBarang: TStaticText;
    txtJmlPO: TStaticText;
    txtJmlSdhTerima: TStaticText;
    dtpTerima: TDateTimePicker;
    edtJmlTerima: TEdit;
    btnClose: TSpeedButton;
    lblNoUrut: TLabel;
    qryGrid: TADOQuery;
    qrySQL: TADOQuery;
    dsGrid: TDataSource;
    qryGridNoPO: TWideStringField;
    qryGridNoUrut: TSmallintField;
    qryGridNoUrutTerima: TSmallintField;
    qryGridTglTerima: TDateTimeField;
    qryGridJumlahDiterima: TBCDField;
    qryGridIDWarehouse: TWideStringField;
    qryGridIDSystemUser: TWideStringField;
    qryGridUpdateTime: TDateTimeField;
    qryGridItemsCode: TWideStringField;
    qryGridMerek: TWideStringField;
    qryGridJumlah: TBCDField;
    qryGridJmlSudahTerima: TBCDField;
    qryGridItemsName: TWideStringField;
    qryGridUOM: TWideStringField;
    lblHarga: TLabel;
    txtNamaBarang: TStaticText;
    txtMerek: TStaticText;
    Label7: TLabel;
```

```

procedure btnCloseClick(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure btnTerimaClick(Sender: TObject);
procedure edtJmlTerimaKeyPress(Sender: TObject; var Key: Char);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  frmReceiveItems: TfrmReceiveItems;
implementation
  {$R *.dfm}
  Uses FormComponent;
  var temp : string;
  procedure TfrmReceiveItems.btnCloseClick(Sender: TObject);
  begin
    Close;end;
  procedure TfrmReceiveItems.btnTerimaClick(Sender: TObject);
  var NoUrutTerima : integer;
  begin
    temp := 'SELECT * FROM Tbl_ReceiveItems WHERE NoPO = ' +
    QuotedStr(txtNoPO.Caption) +
      ' AND NoUrut = ' + lblNoUrut.Caption + ' ORDER BY NoUrutTerima';
    if (StrToFloat(edtJmlTerima.Text) <= StrToFloat(txtJmlPO.Caption) -
    StrToFloat(txtJmlSdhTerima.Caption)) then
      with qrySQL do
        begin
          Close;
          SQL.Clear;
          SQL.Add(temp);
          Open;
          if not(IsEmpty) then
            begin
              Last;   NoUrutTerima := FieldByName('NoUrutTerima').AsInteger + 1;
            end
          else
            NoUrutTerima := 1;
          if not frmComponent.adoTerra.InTransaction then
            begin
              frmComponent.adoTerra.BeginTrans;
              try
                Insert;
                FieldByName('NoPO').AsString := txtNoPO.Caption;
                FieldByName('NoUrut').AsString := lblNoUrut.Caption;
                FieldByName('NoUrutTerima').AsInteger := NoUrutTerima;
                FieldByName('TglTerima').AsDateTime := dtpTerima.Date;
                FieldByName('JumlahDiterima').AsFloat := StrToFloat(edtJmlTerima.Text);
              finally
                frmComponent.adoTerra.CommitTrans;
              end;
            end;
          else
            NoUrutTerima := NoUrutTerima + 1;
          end;
        end;
      end;
  end;
end;

```



```

FieldByName('IDWarehouse').AsString := 'BARU';
FieldByName('IDSystemUser').AsString := frmComponent.lblUserName.Caption;
FieldByName('UpdateTime').AsDateTime := Date;
Post;
temp := 'UPDATE Tbl_DtlPO SET JumlahDiterima = JumlahDiterima + ' +
      edtJmlTerima.Text +
      ' WHERE NoPO = ' + QuotedStr(txtNoPO.Caption) +
      ' AND NoUrut = ' + lblNoUrut.Caption;
Close;
SQL.Clear;
SQL.Add(temp);
ExecSQL;
temp := 'UPDATE Tbl_MstItems SET CostPrice = ((CostPrice*QtyStock + ' +
      edtJmlTerima.Text + '*' + lblHarga.Caption + ')/(QtyStock + ' +
      edtJmlTerima.Text + '), QtyStock = QtyStock + ' + edtJmlTerima.Text +
      ', QtyOnOrder = QtyOnOrder - ' + edtJmlTerima.Text +
      ' WHERE ItemsCode = ' + QuotedStr(txtKodeBarang.Caption) +
      ' AND Merek = ' + QuotedStr(txtMerek.Caption);
Close;
SQL.Clear; SQL.Add(temp);
ExecSQL;
frmComponent.adoTerra.CommitTrans;
ShowMessage('PENERIMAAN BARANG SELESAI');
btnTerima.Enabled := False;
qryGrid.Close;
qryGrid.Open;
Except
  frmComponent.adoTerra.RollbackTrans;
end; end; end; end;

procedure TfrmReceiveItems.edtJmlTerimaKeyPress(Sender: TObject; var Key: Char);
begin
if not (Key in ['0'..'9','.',',','#27,#13,#8,#9]) then
begin
edtJmlTerima.Clear;
edtJmlTerima.SetFocus;
Key := #0; end end;
procedure TfrmReceiveItems.FormShow(Sender: TObject);
begin
with qryGrid do
begin
Close;
Parameters[0].Value := txtNoPO.Caption;
Parameters[1].Value := lblNoUrut.Caption;
Open; end;
dtpTerima.Date := Date; edtJmlTerima.SetFocus; btnTerima.Enabled := True; end;v end.

```