Lampiran Listing Program PCDes

```
Program PCDes;

uses
  Forms,
  DesForm in 'DesForm.pas' {Form1},
  acknowledge in 'Acknowledge.pas' {Form6};

{$R *.RES}

begin
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TForm6, Form6);
  Application.Run;
end.
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit DesForm

```
Unit DesForm;

interface

uses
  WinTypes, WinProcs, Classes, Graphics, Forms, Controls, Buttons,
  StdCtrls, Cryptcon2, ExtCtrls, SysUtils, Dialogs,Messages,
  mmsystem, DESUnit2, Menus,PCCrAbot, PCCrHelp,Acknowledge;

type
  TForm1 = class(TForm)
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Open1: TMenuItem;
    Encryp1: TMenuItem;
    Decryp1: TMenuItem;
    Exit1: TMenuItem;
    Help1: TMenuItem;
    About1: TMenuItem;
    Help2: TMenuItem;
    Penghormatan1: TMenuItem;
    Label1: TLabel;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    Edit1: TEdit;
    Edit2: TEdit;
    IVectorEdit: TEdit;
    Label2: TLabel;
    Bevel1: TBevel;
    Edit3: TEdit;
    CipherModeGroup: TRadioGroup;
    OpenDialog1: TOpenDialog;
    TimePanel: TPanel;
    Source: TLabel;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure Penghormatan1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  boutarray: array[0..31] of char;
  gIV: String;
implementation

{$R *.DFM}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit DesForm

```
procedure TForm1.BitBtn1Click(Sender: TObject);
{begin}
var
 DES: TDES;
 startTime: LongInt;
 encytObject : TCrypto;
 outarray: array[0..15] of char;
begin
DES := TDES.Create(Self);
 try
  If OpenDialog1.Execute then
  Begin
  Edit2.text := OpenDialog1.FileName;
  Screen.Cursor := crHourglass;
    encytObject := DES;
    Try
     encytObject.Key :=
      Edit1.text{'syarif_tauhidpas'{'1234567890123456'};
     encytObject.InputType := SourceFile;
     {CipherMode is an enumerated type}
      encytObject.CipherMode :=
      TCipherMode(CipherModeGroup.ItemIndex);
     if (encytObject.CipherMode > ECBMode) then begin
      if (Length(IVectorEdit.Text) > 7) then{most ciphers have a
           lock size of 8, IVectors must be the same
           size as the BLOCK}
        encytObject.IVector := IVectorEdit.Text
      else
        encytObject.IVector := '';{Let Object Generate IVector}
     end;
     encytObject.InputFilePath := OpenDialog1.FileName;
     encytObject.OutputFilePath :=
ChangeFileExt(OpenDialog1.FileName, '.ccc');
     startTime := timeGetTime;
     encytObject.EncipherData(False);
     IVectorEdit.Text:= encytObject.IVector;
      Except
      On EAccessViolation do
      Application.MassageBox ('Apakah Anda sudah memasukkan
password ?','Peringatan ! Terjadi ekspesi EaccessViolation ',
      MB_OK or MB_ICONSTOP);
      END;
     TimePanel.Caption := IntToStr(timeGetTime - startTime);
  end;{if}
 finally
   DES.free;
   Screen.Cursor := crDefault;
 end;
end;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit DesForm

```
procedure TForm1.BitBtn2Click(Sender: TObject);
{begin}
var
 DES: TDES;
 startTime: LongInt;
 encytObject : TCrypto;
begin
DES := TDES.Create(Self);
 try
  If OpenDialog1.Execute then
  begin
  edit2.text := OpenDialog1.FileName;
  Screen.Cursor := crHourglass;
    encytObject := DES;
     encytObject.Key := Edit1.Text{'1234567890123456'};
     encytObject.InputType := SourceFile;
     encytObject.CipherMode :=
TCipherMode(CipherModeGroup.ItemIndex);
     if encytObject.CipherMode > ECBMode then
        encytObject.IVector := IVectorEdit.Text;
     encytObject.InputFilePath := OpenDialog1.FileName;
     encytObject.OutputFilePath :=
ChangeFileExt(OpenDialog1.FileName, '.ddd');
     startTime := timeGetTime;
     encytObject.DecipherData(False);
     {gIV:= encytObject.IVector;}
     TimePanel.Caption := IntToStr(timeGetTime - startTime);
  end;{if}
 finally
   DES.free;
   Screen.Cursor := crDefault;
 end;
end;

procedure TForm1.Exit1Click(Sender: TObject);
begin
if messagedlg('Apakah Yakin ingin keluar dari Aplikasi EncrypDes?'
              ,mtConfirmation,[mbYes,mbNo],0)=mrYes then
              Close;
end;

procedure TForm1.Penghormatan1Click(Sender: TObject);
begin
     form6.show;
end;

end.
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Acknowledge

```
Unit Acknowledge;

interface

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

type
  TForm6 = class(TForm)
    Button1: TButton;
    Timer1: TTimer;
    Panel1: TPanel;
    PaintBox1: TPaintBox;
    procedure Timer1Timer(Sender: TObject);
    procedure PaintBox1Paint(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form6: TForm6;

Var
   B:Tbitmap;
   St:string;
   Xst,Yst,Dyst,DTime:integer;
   Ts,Td:TRect;

implementation

uses DesForm{PCCrmain};

{$R *.DFM}

procedure TForm6.Timer1Timer(Sender: TObject);
begin
     With Ts do begin
        Top:=Dyst;
        Bottom:=Dyst+Form6.PaintBox1.Height;
        Left:=0;
        Right:=Form6.PaintBox1.Width;
     end;
     With Td do begin
        Top:=0;
        Bottom:=Form6.PaintBox1.Height;
        Left:=0;
        Right:=Form6.PaintBox1.Width;
     end;
     Dyst:=Dyst+1;
     Form6.PaintBox1.Canvas.CopyRect(Td,B.canvas,Ts);
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Acknowledge

```
     If Dyst=640 then begin
        Dyst:=Dyst-1;
        DTime:=DTime+1;
     end;
     If DTime=10 then begin
        DTime:=0;
        Dyst:=0;
     end;
end;

procedure TForm6.PaintBox1Paint(Sender: TObject);
begin
     Form6.PaintBox1.Canvas.CopyRect(Td,B.canvas,Ts);
end;

procedure TForm6.FormCreate(Sender: TObject);
begin
     Dtime:=0;
     B:=Tbitmap.create;
     B.Width:=338;
     B.Height:=800;
     Xst:=338 div 2;
     Yst:=180;
     With B.Canvas do begin
          Brush.color:=clBlack;
          Pen.color:=clwhite;
          FillRect(Rect(0,0,338,800));
          Font.Name:='Arial';
          Font.Size:=10;
          Font.color:=clWhite;
          St:='Digital Data Encryption With Data Encryption
Standard Method';TextOut(Xst-TextWidth(ST)div 2,Yst,St);
          Moveto(Xst-150,Yst+20);Lineto(Xst+150,Yst+20);
          St:=' Oleh : Syarif Tauchid';TextOut(Xst-
TextWidth(ST)div 2,Yst+30,St);
          St:='Mahasiswa FMIPA UNDIP Semarang';TextOut(Xst-
TextWidth(ST)div 2,Yst+30+20,St);
          {}
          St:='CREDITS';TextOut(Xst-TextWidth(ST)div
2,Yst+30+20*4,St);
          Moveto(Xst-40,Yst+130);Lineto(Xst+40,Yst+130);
          St:='Supervisor & Counsellor 1';TextOut(Xst-
TextWidth(ST)-10,Yst+40+20*5,St);
          St:='Drs. Kushartantya, MI
Komp.';TextOut(Xst+10,Yst+40+20*5,St);
          St:='Supervisor & Counsellor 2';TextOut(Xst-
textWidth(ST)-10,yst+40+20*6,St);
          St:='Drs. Putut Sri
Wasito';TextOut(Xst+10,Yst+40+20*6,St);
          {}
          St:='TCrypto Designer';TextOut(Xst-TextWidth(ST)-
10,Yst+40+20*8,St);
          St:='Greg Carter';TextOut(Xst+10,Yst+40+20*8,St);
          St:='Visual Designer';TextOut(Xst-TextWidth(ST)-
10,Yst+40+20*10,St);
          St:='Syarif Tauchid';TextOut(Xst+10,Yst+40+20*10,St);
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Acknowledge

```
          {}
          St:='Special Thanks:';TextOut(Xst-TextWidth(ST)div
2,Yst+30+20*13,St);
          St:='Allah, my parent (Tarsim and Ipa L), my family
(Ade, Diah, Opik, Sudi, and Reni)';TextOut(Xst-TextWidth(ST)div
2,Yst+40+20*14,St);
          St:='and all my best friends (Adi, Ari, Igun, Ismail)
';TextOut(Xst-TextWidth(ST)div 2,Yst+40+20*15,St);
          St:='';TextOut(Xst-TextWidth(ST)div 2,Yst+40+20*16,St);
          St:='';TextOut(Xst-TextWidth(ST)div 2,Yst+40+20*17,St);
          {}
          St:='(c) 2002 - Syarif Tauchid';TextOut(Xst-
TextWidth(ST)div 2,Yst+40+20*25,St);
          St:='Diponegoro University Central Java';TextOut(Xst-
TextWidth(ST)div 2,Yst+40+20*26,St);
     end;
     Dyst:=0;
     Timer1.Interval:=0;
end;

procedure TForm6.Button1Click(Sender: TObject);
begin
     Timer1.interval:=0;
     Dyst:=0;
     Close;
     Form6.PaintBox1.Canvas.Brush.color:=clBlack;
     Form6.PaintBox1.Canvas.FillRect(Rect(0,0,338,800));
end;

procedure TForm6.FormShow(Sender: TObject);
begin
     Timer1.Enabled:=true;
     Timer1.interval:=1;
end;


end.
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Desunit2

```
unit Desunit2;

interface
{Declare the compiler defines}
{$I CRYPTDEF.INC}
{------Changeable compiler switches-----------------------------}
{$A+   Word align variables }
{$F+   Force Far calls }
{$K+   Use smart callbacks
{$N+   Allow coprocessor instructions }
{$P+   Open parameters enabled }
{$S+   Stack checking }
{$T-   @ operator is NOT typed }
{$U-   Non Pentium safe FDIV }
{$Z-   No automatic word-sized enumerations}
{-----------------------------------------------------------------}
uses SysUtils, Cryptcon2{$IFDEF DELPHI}, Classes, Controls{$ENDIF}

type
{$IFDEF DELPHI}
 TDES = class(TCrypto)
{$ENDIF}
Private
  {DES Key Elements}
  FRoundKeys : Array [1..16,1..48] Of Byte;
  FC: Array [1..28] Of Byte;
  FD: Array [1..28] Of Byte;
  FInputValue  : Array [1..64] Of Byte;
  FOutputValue : Array [1..64] Of Byte;
  FL, FR, FFunctionResult : Array [1..32] Of Byte;
  Procedure F (var FK);

  Procedure SetBit (Var Data; Index, Value : Byte);
  Function  GetBit (Var Data; Index : Byte) : Byte;
  Procedure Shift (Var SubKeyPart);
  Procedure SubKey (Round : Byte; Var SubKey);
 {$IFDEF DELPHI}
  Procedure SetKeys;      override; {Sets up En\DecipherKey
SubKeys}
  Procedure EncipherBLOCK;override;
  Procedure DecipherBLOCK;override;
{$ENDIF}
public
    { Public declarations }
{$IFDEF DELPHI}
  constructor Create(Owner: TComponent);override;
  destructor  Destroy;override;
{$ENDIF}
end;{TRC4}
{$IFDEF DELPHI}
 procedure Register;{register the component to the Delphi toolbar}
{$ENDIF}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Desunit2

```
Const
  IP : Array [1..64] Of Byte = (58,50,42,34,26,18,10,2,
                                60,52,44,36,28,20,12,4,
                                62,54,46,38,30,22,14,6,
                                64,56,48,40,32,24,16,8,
                                57,49,41,33,25,17, 9,1,
                                59,51,43,35,27,19,11,3,
                                61,53,45,37,29,21,13,5,
                                63,55,47,39,31,23,15,7);
  InvIP : Array [1..64] Of Byte = (40, 8,48,16,56,24,64,32,
                                   39, 7,47,15,55,23,63,31,
                                   38, 6,46,14,54,22,62,30,
                                   37, 5,45,13,53,21,61,29,
                                   36, 4,44,12,52,20,60,28,
                                   35, 3,43,11,51,19,59,27,
                                   34, 2,42,10,50,18,58,26,
                                   33, 1,41, 9,49,17,57,25);
  E : Array [1..48] Of Byte = (32, 1, 2, 3, 4, 5,
                               4, 5, 6, 7, 8, 9,
                               8, 9,10,11,12,13,
                              12,13,14,15,16,17,
                              16,17,18,19,20,21,
                              20,21,22,23,24,25,
                              24,25,26,27,28,29,
                              28,29,30,31,32, 1);
  P : Array [1..32] Of Byte = (16, 7,20,21,
                              29,12,28,17,
                               1,15,23,26,
                               5,18,31,10,
                               2, 8,24,14,
                              32,27, 3, 9,
                              19,13,30, 6,
                              22,11, 4,25);
  SBoxes : Array [1..8,0..3,0..15] Of Byte =
           (((14, 4,13, 1, 2,15,11, 8, 3,10, 6,12, 5, 9, 0, 7),
             ( 0,15, 7, 4,14, 2,13, 1,10, 6,12,11, 9, 5, 3, 8),
             ( 4, 1,14, 8,13, 6, 2,11,15,12, 9, 7, 3,10, 5, 0),
             (15,12, 8, 2, 4, 9, 1, 7, 5,11, 3,14,10, 0, 6,13)),

            ((15, 1, 8,14, 6,11, 3, 4, 9, 7, 2,13,12, 0, 5,10),
             ( 3,13, 4, 7,15, 2, 8,14,12, 0, 1,10, 6, 9,11, 5),
             ( 0,14, 7,11,10, 4,13, 1, 5, 8,12, 6, 9, 3, 2,15),
             (13, 8,10, 1, 3,15, 4, 2,11, 6, 7,12, 0, 5,14, 9)),

            ((10, 0, 9,14, 6, 3,15, 5, 1,13,12, 7,11, 4, 2, 8),
             (13, 7, 0, 9, 3, 4, 6,10, 2, 8, 5,14,12,11,15, 1),
             (13, 6, 4, 9, 8,15, 3, 0,11, 1, 2,12, 5,10,14, 7),
             ( 1,10,13, 0, 6, 9, 8, 7, 4,15,14, 3,11, 5, 2,12)),

            (( 7,13,14, 3, 0, 6, 9,10, 1, 2, 8, 5,11,12, 4,15),
             (13, 8,11, 5, 6,15, 0, 3, 4, 7, 2,12, 1,10,14, 9),
             (10, 6, 9, 0,12,11, 7,13,15, 1, 3,14, 5, 2, 8, 4),
             ( 3,15, 0, 6,10, 1,13, 8, 9, 4, 5,11,12, 7, 2,14)),

            (( 2,12, 4, 1, 7,10,11, 6, 8, 5, 3,15,13, 0,14, 9),
             (14,11, 2,12, 4, 7,13, 1, 5, 0,15,10, 3, 9, 8, 6),
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Desunit2

```
              ( 4, 2, 1,11,10,13, 7, 8,15, 9,12, 5, 6, 3, 0,14),
              (11, 8,12, 7, 1,14, 2,13, 6,15, 0, 9,10, 4, 5, 3)),

             ((12, 1,10,15, 9, 2, 6, 8, 0,13, 3, 4,14, 7, 5,11),
              (10,15, 4, 2, 7,12, 9, 5, 6, 1,13,14, 0,11, 3, 8),
              ( 9,14,15, 5, 2, 8,12, 3, 7, 0, 4,10, 1,13,11, 6),
              ( 4, 3, 2,12, 9, 5,15,10,11,14, 1, 7, 6, 0, 8,13)),

             (( 4,11, 2,14,15, 0, 8,13, 3,12, 9, 7, 5,10, 6, 1),
              (13, 0,11, 7, 4, 9, 1,10,14, 3, 5,12, 2,15, 8, 6),
              ( 1, 4,11,13,12, 3, 7,14,10,15, 6, 8, 0, 5, 9, 2),
              ( 6,11,13, 8, 1, 4,10, 7, 9, 5, 0,15,14, 2, 3,12)),

             ((13, 2, 8, 4, 6,15,11, 1,10, 9, 3,14, 5, 0,12, 7),
              ( 1,15,13, 8,10, 3, 7, 4,12, 5, 6,11, 0,14, 9, 2),
              ( 7,11, 4, 1, 9,12,14, 2, 0, 6,10,13,15, 3, 5, 8),
              ( 2, 1,14, 7, 4,10, 8,13,15,12, 9, 0, 3, 5, 6,11)));

  PC_1 : Array [1..56] Of Byte = (57,49,41,33,25,17, 9,
                                   1,58,50,42,34,26,18,
                                  10, 2,59,51,43,35,27,
                                  19,11, 3,60,52,44,36,
                                  63,55,47,39,31,23,15,
                                   7,62,54,46,38,30,22,
                                  14, 6,61,53,45,37,29,
                                  21,13, 5,28,20,12, 4);

  PC_2 : Array [1..48] Of Byte = (14,17,11,24, 1, 5,
                                   3,28,15, 6,21,10,
                                  23,19,12, 4,26, 8,
                                  16, 7,27,20,13, 2,
                                  41,52,31,37,47,55,
                                  30,40,51,45,33,48,
                                  44,49,39,56,34,53,
                                  46,42,50,36,29,32);

  ShiftTable : Array [1..16] Of Byte =
(1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1);

  DESKEYLENGTH = 8;

implementation

{$IFDEF DELPHI}
procedure Register;
begin
  RegisterComponents('Crypto', [TDES]);
end;
{$ENDIF}

{==============================TDES==========================}
{$IFDEF DELPHI}
constructor TDES.Create(Owner: TComponent);
{$ENDIF}
begin
  FBLOCKSIZE := 8;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

```
  FIVTemp := nil;
{  GetMem(FIVTemp, FBLOCKSIZE);}
 {$IFDEF DELPHI}
  inherited Create(Owner);
 {$ENDIF}
end;{TDES.Create}

{$IFDEF DELPHI}
destructor TDES.Destroy;
{$ENDIF}
begin
  {$IFDEF DELPHI}
  inherited Destroy;
  {$ENDIF}
end;{TDES.Destroy;}

Function TDES.GetBit (Var Data; Index : Byte) : Byte;
Var
  Bits : Array [0..7] Of Byte ABSOLUTE Data;

Begin
  Dec (Index);
  If Bits[Index DIV 8] And (128 SHR (Index MOD 8))>0 then
GetBit:=1 Else GetBit:=0;
End;{GetBit}

Procedure TDES.SetBit (Var Data; Index, Value : Byte);
Var
  Bits : Array [0..7] Of Byte ABSOLUTE Data;
  Bit : Byte;

Begin
  Dec (Index);
  Bit:=128 SHR (Index MOD 8);
  Case Value Of
    0 : Bits[Index DIV 8]:=Bits[Index DIV 8] And (Not Bit);
    1 : Bits[Index DIV 8]:=Bits[Index DIV 8] Or Bit;
  End;
End;{SetBit}

Procedure TDES.F (Var FK);
Var
  {R : Array [1..48] Of Byte ABSOLUTE FR;}
  K : Array [1..48] Of Byte ABSOLUTE FK;
  Temp1 : Array [1..48] Of Byte;
  Temp2 : Array [1..32] Of Byte;
  n, h, i, j, Row, Column : Integer;
  {TotalOut : Array [1..32] Of Byte ABSOLUTE Output;}

Begin
  For n:=1 to 48 Do Temp1[n]:=FR[E[n]] Xor K[n];
  For n:=1 to 8 Do Begin
    i:=(n-1)*6;
    j:=(n-1)*4;
    Row:=Temp1[i+1]*2+Temp1[i+6];
Column:=Temp1[i+2]*8 + Temp1[i+3]*4 + Temp1[i+4]*2 + Temp1[i+5];
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

```
    For h:=1 to 4 Do Begin
      Case h Of
        1 : Temp2[j+h]:=(SBoxes[n,Row,Column] And 8) DIV 8;
        2 : Temp2[j+h]:=(SBoxes[n,Row,Column] And 4) DIV 4;
        3 : Temp2[j+h]:=(SBoxes[n,Row,Column] And 2) DIV 2;
        4 : Temp2[j+h]:=(SBoxes[n,Row,Column] And 1);
      End;
    End;
  End;
  For n:=1 to 32 Do FFunctionResult[n]:=Temp2[P[n]];
End;{F}

Procedure TDES.Shift (Var SubKeyPart);
Var
  SKP : Array [1..28] Of Byte ABSOLUTE SubKeyPart;
  n, b : Byte;

Begin
  b:=SKP[1];
  For n:=1 to 27 Do SKP[n]:=SKP[n+1];
  SKP[28]:=b;
End;{Shift}

Procedure TDES.SubKey (Round : Byte; Var SubKey);
Var
  SK : Array [1..48] Of Byte ABSOLUTE SubKey;
  n, b : Byte;

Begin
  For n:=1 to ShiftTable[Round] Do Begin
    Shift (FC);
    Shift (FD);
  End;
  For n:=1 to 48 Do Begin
    b:=PC_2[n];
    If b<=28 then SK[n]:=FC[b] Else SK[n]:= FD[b-28];
  End;
End;{SubKey}

Procedure TDES.SetKeys;
var
 n: BYTE;
 Key: Array [0..7] of BYTE;
begin
  move(FKey[1], Key, DESKEYLENGTH);
  For n:=1 to 28 Do Begin
    FC[n]:=GetBit(Key,PC_1[n]);
    FD[n]:=GetBit(Key,PC_1[n+28]);
  End;
  For n:=1 to 16 Do SubKey (n,FRoundKeys[n]);
end;

Procedure TDES.EncipherBLOCK;
var
  n, i, b, Round : Byte;
begin
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Desunit2

```
  For n:=1 to 64 Do FInputValue[n]:=GetBit (FSmallBuffer,n);
  For n:=1 to 64 Do If n<=32 then FL[n]:=FInputValue[IP[n]] Else
FR[n-32]:=FInputValue[IP[n]];
  For Round:=1 to 16 Do Begin
    F (FRoundKeys[Round]);
    For n:=1 to 32 Do FFunctionResult[n]:=FFunctionResult[n] Xor
FL[n];
    FL := FR;
    FR := FFunctionResult;
  End;
  For n:=1 to 64 Do Begin
    b:=InvIP[n];
    If b<=32 then FOutputValue[n]:=FR[b] Else
FOutputValue[n]:=FL[b-32];
  End;
  For n:=1 to 64 Do SetBit (FSmallBuffer,n,FOutputValue[n]);
end;{TDES.EncipherBLOCK}

Procedure TDES.DecipherBLOCK;
var
  n, i, b, Round : Byte;
begin
  For n:=1 to 64 Do FInputValue[n]:=GetBit (FSmallBuffer,n);
  For n:=1 to 64 Do If n<=32 then FL[n]:=FInputValue[IP[n]] Else
FR[n-32]:=FInputValue[IP[n]];
  For Round:=1 to 16 Do Begin
    F (FRoundKeys[17-Round]);
    For n:=1 to 32 Do FFunctionResult[n]:=FFunctionResult[n] Xor
FL[n];
    FL := FR;
    FR := FFunctionResult;
  End;
  For n:=1 to 64 Do Begin
    b:=InvIP[n];
    If b<=32 then FOutputValue[n]:=FR[b] Else
FOutputValue[n]:=FL[b-32];
  End;
  For n:=1 to 64 Do SetBit (FSmallBuffer,n,FOutputValue[n]);
end;{TDES.DecipherBLOCK}

end.
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Cryptcon2

```
unit Cryptcon2;
{*****************************************************************
 UNIT: Cryptcon
Description: This unit contains an Object Pascal Object which can
             be used as a base class for constructing Encryption
             Objects for BLOCK Ciphers.  This contains all the
             necessary fields/methods for doing encryption.  In
             most cases all that a user of this base class will
             need to do is override the
             EncipherBLOCK/DecipherBLOCK and SetKeys(generate
             SubKeys) methods. It handles things such as File
             encryption, Byte Array encryption, different modes of
             operation(CBC, ECB, CFB).
 ----------------------------------------------------------------
 Code Author:  Greg Carter, gregc@cryptocard.com
 Organization: CRYPTOCard Corporation, info@cryptocard.com
               R&D Division, Carleton Place, ON, CANADA, K7C 3T2
               1-613-253-3152 Voice, 1-613-253-4685 Fax.
 Date of V.1:  Jan. 30 1996.

 Compatibility & Testing with BP7.0: Marcel Roorda,
garfield@xs4all.nl
 ----------------------------------------------------------------}

{Declare the compiler defines}
{$I CRYPTDEF.INC}
{------Changeable compiler switches----------------------------}
{$A+   Word align variables }
{$F+   Force Far calls }
{$K+   Use smart callbacks
{$N+   Allow coprocessor instructions }
{$P+   Open parameters enabled }
{$S+   Stack checking }
{$T-   @ operator is NOT typed }
{$IFDEF DELPHI}
{$U-   Non Pentium safe FDIV }
{$Z-   No automatic word-sized enumerations}
{$ENDIF}
{----------------------------------------------------------------}
interface
uses {$IFDEF DELPIH32} Windows,{$ENDIF}SysUtils{$IFDEF DELPHI},
      Classes, Controls{$ENDIF},{$IFDEF BP7}Object{$ENDIF};

type
 {An enumerated typt which tells the object what type the input to
the cipher is}
 TSourceType = (SourceFile, SourceByteArray,SourceString);
 {Different modes of cipher operation}
 TCipherMode = (ECBMode, CBCMode, CFBMode);

 UWORD_32bits = LongInt;
 UWORD_16bits = WORD;
 UBYTE_08bits = BYTE;

 BArray  = array[0..7] of BYTE;
 PByte   = ^Byte;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Cryptcon2

```
 PArray  = ^BArray;
 LArray  = array[0..1] of UWORD_32bits;
 PLong   = ^UWORD_32bits;
 PLArray = ^LArray;

 {Motorola}
 {$IFDEF ORDER_ABCD}
   singleBytes = Record
          byte0: BYTE; {MSB}
          byte1: BYTE;
          byte2: BYTE;
          byte3: BYTE; {LSB}
   end;{SingleBytes}
 {$DEFINE MAC}
 {$ENDIF}

{Intelx86}
{$IFDEF ORDER_DCBA}
   singleBytes = Record
          byte3: BYTE; {LSB}
          byte2: BYTE;
          byte1: BYTE;
          byte0: BYTE; {MSB}
   end;{SingleBytes}
{$DEFINE INTEL}
{$ENDIF}

{VAX, anyone using Pascal on the Vax???}
{$IFDEF ORDER_BADC}
   singleBytes = Record
          byte1: BYTE;
          byte0: BYTE;
          byte3: BYTE;
          byte2: BYTE;
   end;{SingleBytes}
 {$ENDIF}

 aword = record
  case Integer Of
   0: (LWord: UWORD_32bits);
   1: (fByte: Array[0..3] of UBYTE_08bits);
   2: (w: singleBytes);
 end;{aword, 32bits!}

 Paword = ^aword;
{-----------------------------------------------------------------}
{TCrypto Object: TCrypto Object descends from the base class(of
you respestive compiler(TComponent for Delphi, TObject for BP7).
It is a basic 'container' to hold all the input/output information
to an encryption routine. When using a cipher in CBC or CFB mode
the cipher needs an initialization vector(IV).  The user can
either supply an IV vector, by assigning a string value(which MUST
be the same length as block of the cipher) to the FIVector field,
or have an IVector generated for them.  If the user chooses to
have the IVector generated(Do not assign it anything), then upon
completion of the current encryption, the FIVector field will hold
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

```
the IV vector used.  You need to keep a copy of the IV vector used
to encrypt, in order to decrypt anything.  It is not necessary to
keep this value secret. A destructor fills the Objects copy of the
User Key with zeros on exit for security reasons.  'Protected'
Fields under Delphi are accessible to desendents of the inherited
object, but not to users of the desendent objects.
-----------------------------------------------------------------}
{$IFDEF DELPHI}
TCrypto = class(TComponent)
 Protected
{ Protected declarations }
{$ENDIF}
{$IFDEF BP7}
 PTCrypto = ^TCrypto; {For BP7 Objects}
 TCrypto = object(TObject)
 Public              {Since BP7 doesn't support Properties, we make
these Public}
{$ENDIF}
  FIVector: String;  {Initial IVector MUST be as long as
FBLOCKSIZE}
  FIVTemp:  PArray;  {IVector during cipher}
  FKey: String;      {Local Copy of User Key}
  FInputType:  TSourceType;{SourceString, SourceByteArray,
SourceFile}
  FCipherMode: TCipherMode;{ECBMode, CBCMode, CFBMode}
  FInputFilePath:  String; {Path to input file}
  FOutputFilePath: String; {Path to output file}
  FInputArray:  PArray;    {Pointer to input array}
  FOutputArray: PArray;    {Output Array}
  FInputString: String;    {Pascal String to Encipher}
  FInputLength: WORD;      {16bit Unsigned Length of ByteArray}
  FBuffer:      array[0..4096] of BYTE; {Local Copy of Data}
  FSmallBuffer: array[0..63]   of BYTE;
  FDoneFile:  Boolean;     {Signal reading of file or array is
done}
  FBLOCKSIZE: BYTE;        {MUST be initialized in Constructor}
  Procedure ShiftLeft(pIV, pNewData: PByte; Pos: WORD);
  function  MIN(Aparam, Bparam: integer): integer;
  Procedure GenIVector;    {generates a psedo random IVector}
  Procedure InitIV;
  Procedure StartCipher(Continue: Boolean);
  {$IFDEF DELPHI}
  Procedure EncipherBLOCK; virtual;abstract;
  Procedure DecipherBLOCK; virtual;abstract;
 {do any SubKey generation or initialization here}
  Procedure SetKeys;virtual;abstract;
  {$ENDIF}
  {$IFDEF BP7}
  Procedure EncipherBLOCK; virtual;
  Procedure DecipherBLOCK; virtual;
  Procedure SetKeys;virtual;
  {$ENDIF}
  Procedure Encipher_File;
  Procedure Decipher_File;
  Procedure Encipher_Bytes;virtual;
  Procedure Decipher_Bytes;virtual;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Cryptcon2

```
{Different modes for Block Ciphers}
  Procedure EncipherECB;
  Procedure DecipherECB;
  Procedure EncipherCFB;
  Procedure DecipherCFB;
  Procedure EncipherCBC;
  Procedure DecipherCBC;

 public
{ Public declarations }
  Procedure DecipherData(Continue: Boolean);  {Users call these to
perform}
  Procedure EncipherData(Continue: Boolean);
{Encryption/Decryption}
  {Continue is used for CBC and CFB Modes, where the encryption
procedure
   needs to know whether to generate a new Initialization Vector,
or use the
   one generated in the last round of the previous encryption}
{$IFDEF DELPHI}
  {constructor Create(Owner: TComponent);override;}
  destructor  Destroy;override;
  Property pInputArray:  PArray read FInputArray  write
FInputArray;
  Property pOutputArray: PArray read FOutputArray write
FOutputArray;{!!See FOutputArray}
 published
{Published properties show up in the object inspector}
  Property Key: String write FKey Stored False;
  Property InputType:  TSourceType read FInputType      write
FInputType;
  Property InputFilePath:   String read FInputFilePath  write
FInputFilePath;
  Property OutputFilePath:  String read FOutputFilePath write
FOutputFilePath;
  Property InputString:     String read FInputString    write
FInputString;
  Property InputLength:       WORD read FInputLength    write
FInputLength;
  Property CipherMode: TCipherMode read FCipherMode     write
FCipherMode;
  Property IVector:         String read FIVector        write
FIVector Stored False;
{$ENDIF}
{$IFDEF BP7}
  {constructor Init;}
  destructor  Done;virtual;
{$ENDIF}
end;{TCrypto}

implementation

{$IFDEF DELPHI}
destructor TCrypto.Destroy;
{$ENDIF}
{$IFDEF BP7}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

```
destructor TCrypto.Done;
{$ENDIF}
  var i: integer;
  begin
   If FIVTemp <> nil then begin
      FreeMem(FIVTemp, FBLOCKSIZE);
   end;
   for i := 1 to Length(FKey) do begin
    FKey[i] := #0;
   end;
{$IFDEF DELPHI}
  inherited Destroy;
{$ENDIF}
  end;{destructor}

{$IFDEF BP7}
{============Dummy Methods, You must Override these!===========}
Procedure TCrypto.EncipherBLOCK;
begin
end;
Procedure TCrypto.DecipherBLOCK;
begin
end;
Procedure TCrypto.SetKeys;
begin
end;
{$ENDIF}

{=======================Misc. Methods============================}
function TCrypto.MIN(Aparam, Bparam: integer): integer;
begin
 if Aparam > Bparam then
  MIN := Bparam
 else
  MIN := Aparam;
end;

Procedure TCrypto.ShiftLeft(pIV, pNewData: PByte; Pos: WORD);
{Used in CFB Mode}
var
 TempPtr: PByte;
 i: BYTE;
begin
 TempPtr := pIV; Inc(TempPtr, Pos);
 For i:= 1 To (FBLOCKSIZE - Pos) do
  pIV^ := TempPtr^; Inc(pIV); Inc(TempPtr);
 repeat
  pIV^ := pNewData^; Inc(pIV); Inc(pNewData);
  Dec(Pos);
 Until Pos = 0;
end;{TCrypto.ShiftLeft}

Procedure TCrypto.GenIVector;
var
 i: WORD;
begin
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Cryptcon2

```
 Randomize;
 FIVector := '';
 For i:= 1 to FBLOCKSIZE do begin
   FIVector := FIVector + Chr(BYTE(Random(93) + 33));
   {add 33 so all in ascii printable range}
 end;
end;

Procedure TCrypto.InitIV;
begin
 {If the user wishes to supply an IV vector then let them,
therwise we generate one, and put it in FIVector, so that they can
see it, also put IV in FIVTemp, which is used during the cipher
routines}
 If FIVector = '' then
    GenIVector;
 if FIVTemp = nil then
    GetMem(FIVTemp, FBLOCKSIZE);
 Move(FIVector[1], FIVTemp^, FBLOCKSIZE);
end;

Procedure TCrypto.StartCipher(Continue: Boolean);
begin
if Not Continue then begin
 SetKeys;
 Case FCipherMode of
  ECBMode:
   begin
   end;
  CBCMode:
   begin
    InitIV;
   end;
  CFBMode:
   begin
    InitIV;
   end;
 end;{Case}
end;{if}
end;{TCrypto.StartCipher}

{===============Main Entry Public Methods=========================}
Procedure TCrypto.EncipherData(Continue: Boolean);
{Public/Protected Procedure used to encipher data}
var
pStr: PChar;
 begin
   StartCipher(Continue);
   case FInputType of
    SourceFile:
    begin
     Encipher_File;
    end;
    SourceByteArray:
    begin
     {Check Length!!!!}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Cryptcon2

```
      Move(FInputArray^, FBuffer, FInputLength);
      Encipher_Bytes;
     end;
     SourceString:
    begin
     {Convert Pascal String to Byte Array}
 {$IFDEF DELPHI}
     pStr := StrAlloc(Length(FInputString) + 1);
     try {protect dyanmic memory allocation}
     StrPCopy(pStr, FInputString);
 {$ENDIF}
 {$IFDEF BP7}
     GetMem(pStr, Length(FInputString));
     Move(FInputString[1],pStr^, Length(FInputString));
 {$ENDIF}
     FInputLength := Length(FInputString);
     FInputArray := Pointer(pStr);
     {Check Length!!!!}
     Move(FInputArray^, FBuffer, FInputLength);
     Encipher_Bytes;
 {$IFDEF DELPHI}
     finally
      StrDispose(pStr);
     end;
 {$ENDIF}
 {$IFDEF BP7}
     FreeMem(pStr,Length(FInputString));
 {$ENDIF}
    end;{SourceString}
    end;{case}
end;{TCrypto.EncipherData}

Procedure TCrypto.DecipherData(Continue: Boolean);
{Public/Proctected Procedure used to Decipher data}
 begin
   StartCipher(Continue);
   case FInputType of
    SourceFile:
    begin
     Decipher_File;
    end;
    SourceByteArray:
    begin
    {Check Length!!!!}
     Move(FInputArray^, FBuffer, FInputLength);
     Decipher_Bytes;
    end;
    SourceString:
    begin
    {FIXME: Error, can't decipher input as Pascal string}
    end;{SourceString}
   end;{case}
end;{TCrypto.DecipherData}

{==================Data handling Methods=========================}
Procedure TCrypto.Encipher_Bytes;
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Cryptcon2

```
begin
    Case FCipherMode of
     ECBMode:
      EncipherECB;
     CBCMode:
      EncipherCBC;
     CFBMode:
      EncipherCFB;
    end;{Case}
end;

Procedure TCrypto.Encipher_File;
var
  InputFile, OutputFile: File;
  NumWrite, NumRead: {{$IFDEF DELPHI32}integer
{{$ELSE}{WORD{$ENDIF};
  DoneFile: Boolean;
begin
 DoneFile := False;
 {$IFDEF DELPHI}
 AssignFile(InputFile, FInputFilePath);
 Reset(InputFile, 1);
 NumWrite := FileCreate(FOutputFilePath);
 FileClose(NumWrite);
 AssignFile(OutputFile, FOutputFilePath);
{$ENDIF}
{$IFDEF BP7}
 Assign(InputFile,FInputFilePath);
 Reset(InputFile, 1);
 Assign(OutputFile, FOutputFilePath);
{$ENDIF}
 Reset(OutputFile, 1);
 repeat
    BlockRead(InputFile,FBuffer,4096, NumRead{FInputLength});
    FInputLength := NumRead;
    if FInputLength<>4096 then DoneFile := True;
    {Call Encipher_Bytes to handle the actual encryption}
    FInputArray := @FBuffer;
    FOutputArray := @FBuffer;
    Encipher_Bytes;
    {Case FCipherMode of
     ECBMode:
      EncipherECB;
     CBCMode:
      EncipherCBC;
     CFBMode:
      EncipherCFB;
    end;{Case}
    {Put in OutputFile}
    BlockWrite(OutputFile,FBuffer, FInputLength,NumWrite);
    {Should signal a disk full error when numwrite<>FInputLength}
 until DoneFile or (NumWrite <> FInputLength);
{$IFDEF DELPHI}
  CloseFile(InputFile);
  CloseFile(OutputFile);
{$ENDIF}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Cryptcon2

```
{$IFDEF BP7}
  Close(InputFile);
  Close(OutputFile);
{$ENDIF}
{   FInputLength := TotalRead;}
end;{TCrypto.Encipher_File}

Procedure TCrypto.Decipher_Bytes;
begin
  Case FCipherMode of {keep in this order for compiler
                        optimization}
     ECBMode:
      DecipherECB;
     CBCMode:
      DecipherCBC;
     CFBMode:
      DecipherCFB;
    end;{Case}
end;{TCrypto.Decipher_Bytes}

Procedure TCrypto.Decipher_File;
var
  InputFile, OutputFile: File;
  NumWrite, NumRead: {{$IFDEF DELPHI32}integer
{{$ELSE}{WORD{$ENDIF};
  DoneFile: Boolean;
begin
 DoneFile := False;
{$IFDEF DELPHI}
 AssignFile(InputFile, FInputFilePath);
 Reset(InputFile, 1);
 NumWrite := FileCreate(FOutputFilePath);
 FileClose(NumWrite);
 AssignFile(OutputFile, FOutputFilePath);
{$ENDIF}
{$IFDEF BP7}
 Assign(InputFile, FInputFilePath);
 Reset(InputFile, 1);
 Assign(OutputFile, FOutputFilePath);
{$ENDIF}
 Reset(OutputFile, 1);

 repeat
    BlockRead(InputFile,FBuffer,4096, NumRead{FInputLength});
    FInputLength := NumRead;
    if FInputLength<>4096 then DoneFile := True;
    FInputArray := @FBuffer;
    FOutputArray := @FBuffer;
    Decipher_Bytes;
    {Put in OutputFile}
    BlockWrite(OutputFile,FBuffer, FInputLength,NumWrite);
    {Should signal a disk full error when numwrite<>FInputLength}
 until DoneFile or (NumWrite <> FInputLength);
{$IFDEF DELPHI}
  CloseFile(InputFile);
  CloseFile(OutputFile);
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Cryptcon2

```
{$ENDIF}
{$IFDEF BP7}
  Close(InputFile);
  Close(OutputFile);
{$ENDIF}
{  FInputLength := TotalRead;}
end;{TCrypto.Decipher_File}

{=======================Cipher Mode Methods=======================}
Procedure TCrypto.EncipherCFB;
var
 i: WORD;
 WhatsLeft, Index: Longint;
 pOut: PByte;
 curSize : BYTE;
begin
  WhatsLeft := FInputLength;
  curSize   := MIN(FBLOCKSIZE, WhatsLeft);
  pOut      := PByte(FOutputArray); {get pointer to users
outputarray}
  Index := 0;
  while (curSize > 0) do begin
    Move(FIVTemp^, FSmallBuffer, FBLOCKSIZE);
    EncipherBLOCK;
    For i:= 0 to (curSize - 1) do begin
     PArray(pOut)^[i] := FBuffer[Index + i] Xor FSmallBuffer[i];
    end;
    If curSize = FBLOCKSIZE then
       Move(pOut^, FIVTemp^, FBLOCKSIZE)
    else
       ShiftLeft(Pointer(FIVTemp), Pointer(pOut), curSize);
    Dec(WhatsLeft, curSize);
    Inc(pOut,  curSize);
    Inc(Index, curSize);
    curSize:= MIN(FBLOCKSIZE, WhatsLeft);
 end;{while}
end;{TCrypto.EncipherCFB}

Procedure TCrypto.DecipherCFB;
var
 i: WORD;
 WhatsLeft, Index: Longint;
 pOut: PByte;
 curSize : BYTE;
begin
  WhatsLeft := FInputLength;
  curSize   := MIN(FBLOCKSIZE, WhatsLeft);
  pOut   := PByte(FOutputArray);   {save pointer to users
outputarray}
  Index := 0;
  while (curSize > 0) do begin
    Move(FIVTemp^, FSmallBuffer, FBLOCKSIZE);
    EncipherBLOCK;
    {Put Cipher Text in Feeback Register, IVTemp}
    If curSize = FBLOCKSIZE then
       Move(FBuffer[Index], FIVTemp^, FBLOCKSIZE)
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

```
    else
       ShiftLeft(Pointer(FIVTemp), @FBuffer[Index], curSize);
    For i:= 0 to (curSize - 1) do begin
     PArray(pOut)^[i] := FBuffer[Index + i] Xor FSmallBuffer[i];
    end;
    Dec(WhatsLeft, curSize);
    Inc(Index, curSize);
    Inc(pOut,  curSize);
    curSize   := MIN(FBLOCKSIZE, WhatsLeft);
 end;{while}
end;{TCrypto.DecipherCFB}

Procedure TCrypto.EncipherECB;
var Index: WORD;
begin
 {Pad the input to a multiple of 64bits(8BYTES) with Nulls}
  while (FInputLength mod FBLOCKSIZE)<>0 do begin
    FBuffer[FInputLength] := 0;
    Inc(FInputLength);
  end;
  Index := 0;
  repeat {Do one BLOCK at a time}
    Move(FBuffer[Index], FSmallBuffer, FBLOCKSIZE);
    EncipherBLOCK;
    Move(FSmallBuffer, FOutputArray^[Index], FBLOCKSIZE);
    Inc(Index,FBLOCKSIZE);
  until Index = FInputLength;
end;{TCrypto.EncipherECB}

Procedure TCrypto.DecipherECB;
var Index: WORD;
begin
 {Pad the input to a multiple of FBLOCKSIZE with Nulls}
  while (FInputLength mod FBLOCKSIZE)<>0 do begin
    FBuffer[FInputLength] := 0;
    Inc(FInputLength);
  end;
  Index := 0;
  repeat {Do one BLOCK at a time}
    Move(FBuffer[Index], FSmallBuffer, FBLOCKSIZE);
    DecipherBLOCK;
    Move(FSmallBuffer, FOutputArray^[Index], FBLOCKSIZE);
    Inc(Index,FBLOCKSIZE);
  until Index = FInputLength;
end;{TCrypto.DecipherECB}

Procedure TCrypto.EncipherCBC;
{Purpose:  Performs Cipher Block Chaining(CBC) mode encrytion.
         C_i := E(P_i Xor C_i-1 )}
var
  pOut: PArray;
  Index: LongInt;
  i: WORD;
begin
 pOut := PArray(FOutputArray);
 Index := 0;
```

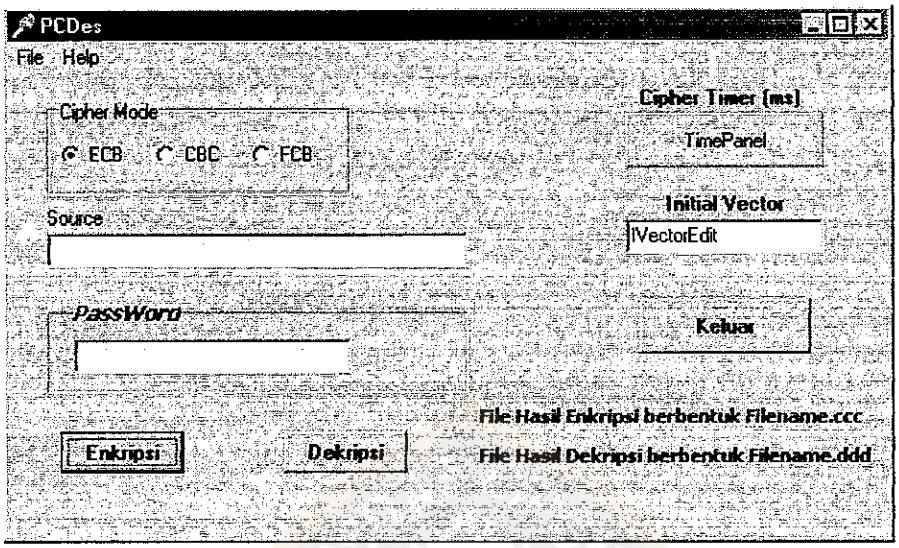
This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Listing Program PCDes Unit Cryptcon2

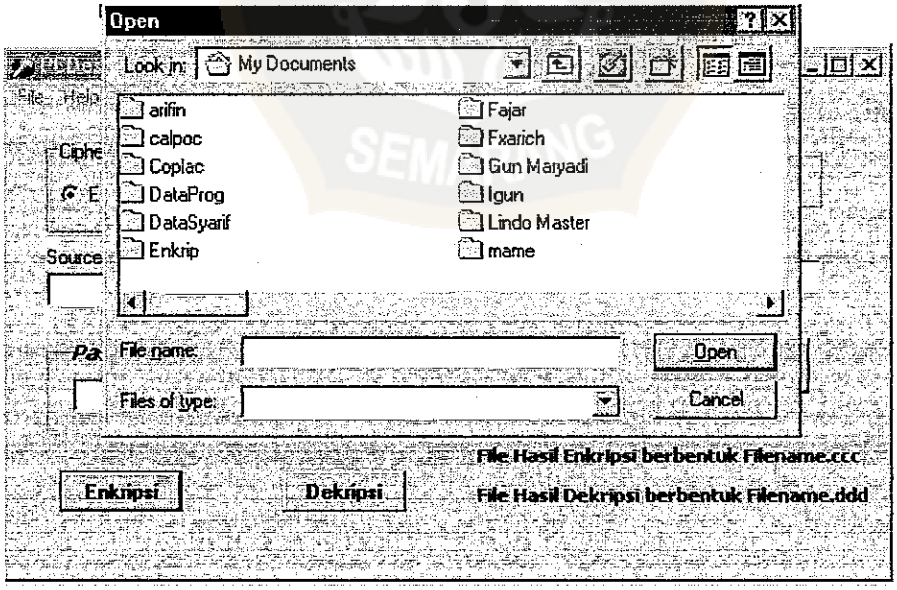
```
{Pad the input to a multiple of FBLOCKSIZE with Nulls}
  while (FInputLength mod FBLOCKSIZE)<>0 do begin
    FBuffer[FInputLength] := 0;
    Inc(FInputLength);
  end;
  repeat {Do one BLOCK at a time}
    Move(FBuffer[Index], FSmallBuffer, FBLOCKSIZE);
    For i:= 0 to (FBLOCKSIZE - 1) do begin
     FSmallBuffer[i] := FSmallBuffer[i] Xor PArray(FIVTemp)^[i];
    end;
    EncipherBLOCK;
    Move(FSmallBuffer, FIVTemp^, FBLOCKSIZE);    {Put Cipher text
in FB register}
    Move(FSmallBuffer, pOut^[Index], FBLOCKSIZE);{Put Cipher in
Output Buffer}
    Inc(Index,FBLOCKSIZE);
  until Index = FInputLength;
end;{TCrypto.EncipherCBC}

Procedure TCrypto.DecipherCBC;
{Purpose:  Performs Cipher Block Chaining(CBC) mode Decrytion.
           P_i := C_i-1 Xor D(Ci)}
var
  pOut: PArray;
  Index : LongInt;
  i: WORD;
begin
 pOut := PArray(FOutputArray);
 Index := 0;
{Pad the input to a multiple of FBLOCKSIZE with Nulls}
  while (FInputLength mod FBLOCKSIZE)<>0 do begin
    FBuffer[FInputLength] := 0;
    Inc(FInputLength);
  end;
  repeat {Do one BLOCK at a time}
    Move(FBuffer[Index], FSmallBuffer, FBLOCKSIZE);
    DecipherBLOCK; {Decipher C_i}
    For i:= 0 to (FBLOCKSIZE - 1) do begin {xor with C_i-1}
     FSmallBuffer[i] := FSmallBuffer[i] Xor PArray(FIVTemp)^[i];
    end;
    Move(FBuffer[Index], FIVTemp^, FBLOCKSIZE); {Save next IV}
    Move(FSmallBuffer, PArray(pOut)^[Index], FBLOCKSIZE); {Save
Plain Text}
    Inc(Index,FBLOCKSIZE);
  until Index = FInputLength;
end;{TCrypto.EncipherCBC}

end.
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Tampilan Program



Tampilan 1. Menu utama program enkripsi dengan metode DES

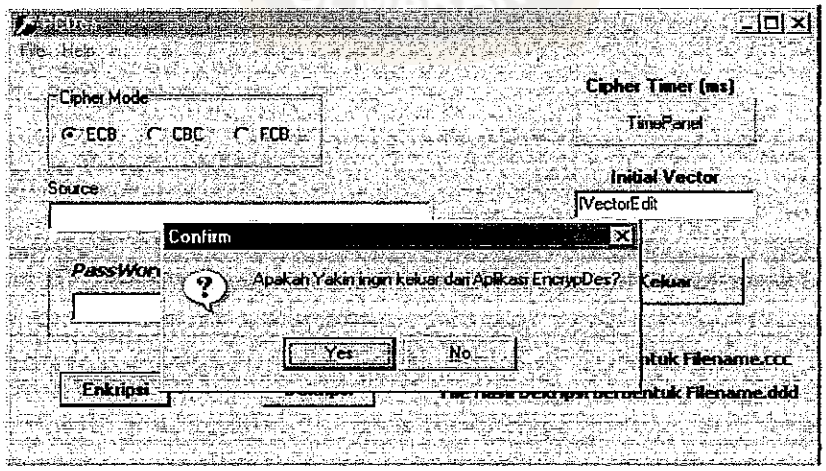


Tampilan 2. Tampilan pada saat button atau menu enkripsi/dekripsi dipilih

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Tampilan Program

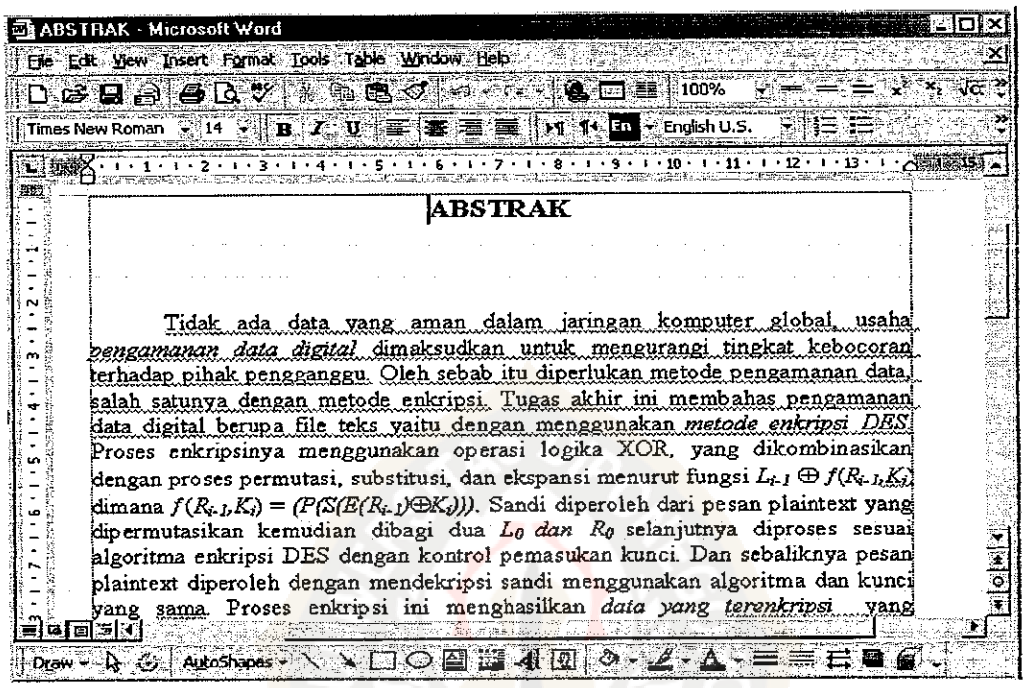
Tampilan 3. Tampilan pada saat menu acknowledge dipilih dari menu help



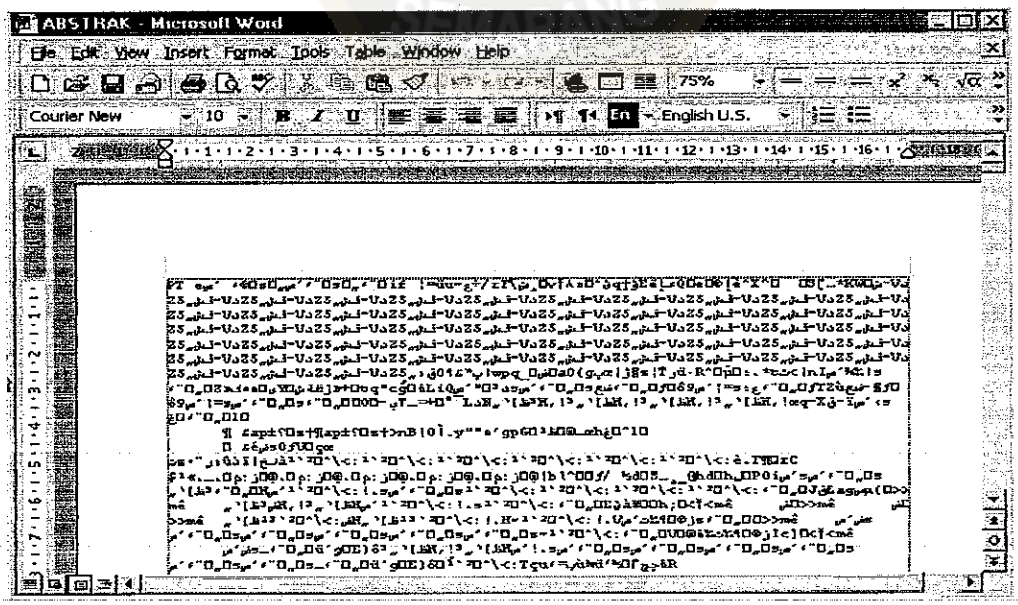
Tampilan 4. Tampilan pada saat menu atau button keluar dipilih dari menu file

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Lampiran Tampilan Program



Tampilan 5. Tampilan data yang belum dienkripsi



Tampilan 6. Tampilan data yang telah terenkripsi

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)