# LAMPIRAN A

## *LISTING PROGRAM*

### MAIN FORM

```
unit main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, ExtCtrls, ComCtrls, ToolWin, Menus, Buttons,
  crypt, sha1, misty1, FrAbout,
  FrPass, FrHowTo, ImgList, LbAsym, LbRSA, LbCipher, LbClass;

type
  TMainform = class(TForm)
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Reset1: TMenuItem;
    Exit1: TMenuItem;
    HowTo1: TMenuItem;
    About1: TMenuItem;
    ToolBar1: TToolBar;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    TabSheet3: TTabSheet;
    Panel1: TPanel;
    Label1: TLabel;
    Inputbox: TEdit;
    Outputbox: TEdit;
    GroupBox1: TGroupBox;
    btnGo: TBitBtn;
    OpenDialog1: TOpenDialog;
    mmoInput: TMemo;
    SaveDialog1: TSaveDialog;
    GroupBox3: TGroupBox;
    Label2: TLabel;
    Label3: TLabel;
    ImageList1: TImageList;
    btnopenfile: TToolButton;
    btnsave: TToolButton;
    btnclear: TToolButton;
    GroupBox4: TGroupBox;
    mmoTextDig: TMemo;
    GroupBox5: TGroupBox;
    edtPrivateE: TEdit;
    GroupBox7: TGroupBox;
    edtPublicE: TEdit;
    GroupBox8: TGroupBox;
    mmoTextVal: TMemo;
    GroupBox9: TGroupBox;
    Label4: TLabel;
    mmoValKey: TMemo;
    Label5: TLabel;
    mmoVerKey: TMemo;
    chkfiledel: TCheckBox;
    StatusBar1: TStatusBar;
    LbRSASSA1: TLbRSASSA;
```

```
SaveDlgPub: TSaveDialog;
edtPublicM: TEdit;
Label6: TLabel;
Label7: TLabel;
SaveDlgSign: TSaveDialog;
OpenDlgPub: TOpenDialog;
OpenDlgSign: TOpenDialog;
btnOpDig: TToolButton;
Label8: TLabel;
Label9: TLabel;
mmoSignature: TMemo;
btnGenerate: TToolButton;
btnSignature: TToolButton;
btnVerSign: TToolButton;
btnpublickey: TToolButton;
PopupMenu1: TPopupMenu;
SavePubKey: TMenuItem;
LoadPubKey: TMenuItem;
btnSaveSign: TToolButton;
PopupMenu2: TPopupMenu;
SaveSign: TMenuItem;
LoadSign: TMenuItem;
btnClearSign: TToolButton;
btnOpenVal: TToolButton;
btnClearVal: TToolButton;
btnValText: TToolButton;
btnVerText: TToolButton;
ToolButton1: TToolButton;
ToolButton2: TToolButton;
ToolButton3: TToolButton;
btnSavText: TToolButton;
GroupBox2: TGroupBox;
btnenkrip: TRadioButton;
btndekrip: TRadioButton;
Image1: TImage;
GroupBox6: TGroupBox;
btnteks: TRadioButton;
btngambarbmp: TRadioButton;
btngambaremf: TRadioButton;
procedure Exit1Click(Sender: TObject);
procedure btnenkripClick(Sender: TObject);
procedure btndekripClick(Sender: TObject);
procedure btnclearClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btnGoClick(Sender: TObject);
procedure Reset1Click(Sender: TObject);
procedure btnopenfileClick(Sender: TObject);
procedure btnsaveClick(Sender: TObject);
procedure HowTo1Click(Sender: TObject);
procedure About1Click(Sender: TObject);
procedure mmoTextDigChange(Sender: TObject);
procedure PageControl1Change(Sender: TObject);
procedure btnOpDigClick(Sender: TObject);
procedure btnGenerateClick(Sender: TObject);
procedure btnSignatureClick(Sender: TObject);
procedure btnVerSignClick(Sender: TObject);
procedure btnClearSignClick(Sender: TObject);
procedure LbRSASSA1GetSignature(Sender: TObject;
  var Sig: TRSASignatureBlock);
procedure SavePubKeyClick(Sender: TObject);
procedure LoadPubKeyClick(Sender: TObject);
procedure SaveSignClick(Sender: TObject);
procedure LoadSignClick(Sender: TObject);
procedure mmoTextValChange(Sender: TObject);
```

```
    procedure btnOpenValClick(Sender: TObject);
    procedure btnClearValClick(Sender: TObject);
    procedure btnValTextClick(Sender: TObject);
    procedure btnVerTextClick(Sender: TObject);
    procedure btnSavTextClick(Sender: TObject);
    procedure LbRSASSA1Progress(Sender: TObject; var Abort: Boolean);
    procedure btnteksClick(Sender: TObject);
    procedure btngambaremfClick(Sender: TObject);
    procedure btngambarbmpClick(Sender: TObject);

  private
    { Private declarations }
    ifname    : String;
    CRC32     : LongWord;
    CRC32Text : LongWord;
    CRC32File : LongWord;
    procedure updatedisplay;
  public
    { Public declarations }
  end;

var
  Mainform: TMainform;
  ref : longword;

implementation
//unit pendukung program utama
uses CRC32, LbUtils, Math, LbBigInt;
{$R *.dfm}

//kostanta /variabel dari program utama
const
  FC_EXT = '.misty';
  TXT_FLT = 'Text File (*.txt)|*.txt|PDF File (*.pdf)|*.pdf|Document File
(*.doc)|*.doc';
  TXT1_FLT = 'Text File (*.txt)|*.txt';
  MIS_FLT = 'Misty File (*'+FC_EXT+')|*'+FC_EXT+'';
  BIT_FLT = 'Bitmap File (*.bmp)|*.bmp';
  EMF_FLT = 'Metafile File (*.emf)|*.emf';
  VAL_FLT = 'Valid Key  (*.val)|*.val';
  sTunggu = 'Silahkan Tunggu Sebentar......';
  sPass = 'Verifikasi Sukses';
  sFail = 'Verifikasi Gagal';

procedure TMainform.Exit1Click(Sender: TObject);
begin
if MessageDlg('Keluar Program ?',mtConfirmation,
                     [mbYes,mbNo],0) = mrYes then
      Application.Terminate;
end;

procedure TMainform.btnenkripClick(Sender: TObject);
begin
  if btnenkrip.Checked= true then
  begin
  btnGo.Caption:= 'Encrypt';
  GroupBox1.Caption:= 'Input';
  chkfiledel.Visible:= False;
  btnclear.Click;
  Pagecontrol1.TabIndex :=0;
  end;
  if PageControl1.TabIndex = 0 then
  begin
    btnteks.Visible:= False;
```

```
    btngambaremf.Visible:=False;
    btngambarbmp.Visible:=False;
    GroupBox6.Visible:=False;
    btnopenfile.Visible:=False;
    btnsave.Visible:=False;
    btnclear.Visible:=False;
    btnOpDig.Visible:=True;
    btnSavText.Visible:=True;
    btnGenerate.Visible:=True;
    btnSignature.Visible:=True;
    btnVerSign.Visible:=False;
    btnpublickey.Visible:=True;
    btnSaveSign.Visible:=True;
    btnClearSign.Visible:=True;
    btnOpenVal.Visible:=False;
    btnClearVal.Visible:=False;
    btnValText.Visible:=False;
    btnVerText.Visible:=False;
    ToolButton1.Visible:=True;
    ToolButton2.Visible:=True;
    end;
  if PageControl1.TabIndex = 2 then
  begin
    btnteks.Visible:= False;
    btngambaremf.Visible:=False;
    btngambarbmp.Visible:=False;
    GroupBox6.Visible:=False;
    btnopenfile.Visible:=False;
    btnsave.Visible:=True;
    btnclear.Visible:=False;
    btnOpDig.Visible:=False;
    btnSavText.Visible:=False;
    btnGenerate.Visible:=False;
    btnSignature.Visible:=False;
    btnVerSign.Visible:=False;
    btnpublickey.Visible:=False;
    btnSaveSign.Visible:=False;
    btnClearSign.Visible:=False;
    btnOpenVal.Visible:=True;
    btnClearVal.Visible:=True;
    btnValText.Visible:=True;
    btnVerText.Visible:=False;
    ToolButton1.Visible:=False;
    ToolButton2.Visible:=False;
    ToolButton3.Visible:=True;
    end;
end;

procedure TMainform.btndekripClick(Sender: TObject);
begin
  if btndekrip.Checked= true then
  begin
  btnGo.Caption:= 'Decrypt';
  GroupBox1.Caption:= 'Output';
  chkfiledel.Visible:= True;
  chkfiledel.Checked:= False;
  btnclear.Click;
  Pagecontrol1.TabIndex :=2;
  end;
  if PageControl1.TabIndex = 0 then
  begin
    btnteks.Visible:= False;
    btngambarbmp.Visible:=False;
    btngambaremf.Visible:=False;
```

```
    GroupBox6.Visible:=False;
    btnopenfile.Visible:=False;
    btnsave.Visible:=False;
    btnclear.Visible:=False;
    btnOpDig.Visible:=True;
    btnSavText.Visible:=False;
    btnGenerate.Visible:=False;
    btnSignature.Visible:=False;
    btnVerSign.Visible:=True;
    btnpublickey.Visible:=True;
    btnSaveSign.Visible:=True;
    btnClearSign.Visible:=True;
    btnOpenVal.Visible:=False;
    btnClearVal.Visible:=False;
    btnValText.Visible:=False;
    btnVerText.Visible:=False;
    ToolButton1.Visible:=True;
    ToolButton2.Visible:=True;
  end;
  if PageControl1.TabIndex = 2 then
  begin
    btnteks.Visible:= False;
    btngambarbmp.Visible:=False;
    btngambaremf.Visible:=False;
    GroupBox6.Visible:=False;
    btnopenfile.Visible:=False;
    btnsave.Visible:=False;
    btnclear.Visible:=False;
    btnOpDig.Visible:=False;
    btnSavText.Visible:=False;
    btnGenerate.Visible:=False;
    btnSignature.Visible:=False;
    btnVerSign.Visible:=False;
    btnpublickey.Visible:=False;
    btnSaveSign.Visible:=False;
    btnClearSign.Visible:=False;
    btnOpenVal.Visible:=True;
    btnClearVal.Visible:=True;
    btnValText.Visible:=False;
    btnVerText.Visible:=True;
    ToolButton1.Visible:=False;
    ToolButton2.Visible:=False;
    ToolButton3.Visible:=True;
  end;
end;

//untuk mengambil dan memasukkan data ke mmoInput
procedure TMainform.btnopenfileClick(Sender: TObject);
const
  BufferSize = 16384;
var
  IOBuffer  : Array of Byte;  //buffer for CRC16 computation
  Error32   : Word;
  blk       : string;
  FileBytes : TInteger8;
  bagi      : real;
begin
  mmoInput.ScrollBars:=ssVertical;
  OpenDialog1.Options :=
    [ofHideReadOnly,ofAllowMultiSelect,ofEnableSizing];
  OpenDialog1.FileName:='';
  if btnenkrip.Checked = true then
  begin
    if btnteks.Checked =true then
```

```
    OpenDialog1.Filter := TXT_FLT;
    if btngambarbmp.Checked =true then
    OpenDialog1.Filter := BIT_FLT;
    if btngambaremf.Checked =true then
    OpenDialog1.Filter := EMF_FLT;
  end
  else
    if btndekrip.Checked = true then
    begin
    btnteks.Checked:=True;
    btnteks.Visible:=True;
    btngambarbmp.Visible:=True;
    btngambaremf.Visible:=True;
    GroupBox6.Visible:=True;
    OpenDialog1.Filter := MIS_FLT;
    end;
  if OpenDialog1.Execute then
  begin
      with OpenDialog1 do
      begin
        if btnteks.Checked =true then
        begin
        ifname:=OpenDialog1.FileName;
        StatusBar1.Panels[1].Text := OpenDialog1.FileName;
        mmoInput.Lines.LoadFromFile(opendialog1.filename)
        end
        else
        begin
        ifname:=OpenDialog1.FileName;
        StatusBar1.Panels[1].Text := OpenDialog1.FileName;
        Image1.Picture.LoadFromFile(opendialog1.filename);
        end;
        Inputbox.Text:=ifname;
        SetLength(IOBuffer, BufferSize);
        CalcFileCRC32(OpenDialog1.FileName, CRC32File, FileBytes,
Error32);
        bagi:=(FileBytes/1024);
        if bagi>1024 then
          blk:=Format('%.2f MB', [bagi/1024])
        else
          blk:=Format('%2f KB', [bagi]);
          StatusBar1.Panels[0].Text:='Total Size ='+blk;
          Screen.Cursor:=crHourGlass;
          Screen.Cursor:=crDefault;
      end;
    end;
end;

//untuk membersihkan teks
procedure TMainform.btnclearClick(Sender: TObject);
begin
  mmoInput.Clear;
  mmoInput.ScrollBars:=ssNone;
  Inputbox.Text:='';
  Outputbox.Text:='';
  StatusBar1.Panels[0].Text:= 'No Message';
  StatusBar1.Panels[1].Text:= '';
end;

//untuk memanggil isi dari How To
procedure TMainform.HowTo1Click(Sender: TObject);
var
  manual: string;
begin
```

```
  manual := ExtractFileDir(ParamStr(0))+'\'+'HowTo.txt';
  if FileExists(manual) then
  begin
    FormHowTo.RichEdit1.ScrollBars:=ssVertical;
    FormHowTo.RichEdit1.Lines.LoadFromFile(manual);
    FormHowTo.ShowModal;
  end;
end;

//untuk menampilkan form about
procedure TMainform.About1Click(Sender: TObject);
begin
  FormAbout.Show;
end;

//inisialisasi program
procedure TMainform.FormCreate(Sender: TObject);
begin
  mmoInput.Clear;
  mmoTextDig.Clear;
  mmoSignature.Clear;
  mmoTextVal.Clear;
  mmoValKey.Clear;
  mmoVerKey.Clear;
  edtPrivateE.Clear;
  edtPublicE.Clear;
  edtPublicM.Clear;
  Inputbox.Text:='';
  Outputbox.Text:='';
  chkfiledel.Checked:=False;
  chkfiledel.Visible:=False;
  btnenkrip.Checked:=True;
  btndekrip.Checked:=False;
  btnteks.Visible:=False;
  btngambarbmp.Visible:=False;
  btngambaremf.Visible:=False;
  GroupBox6.Visible:=False;
  btnopenfile.Visible:=False;
  btnsave.Visible:=False;
  btnclear.Visible:=False;
  btnOpDig.Visible:=True;
  btnGenerate.Visible:=True;
  btnSignature.Visible:=True;
  btnVerSign.Visible:=False;
  btnpublickey.Visible:=True;
  btnSaveSign.Visible:=True;
  btnClearSign.Visible:=True;
  btnOpenVal.Visible:=False;
  btnClearVal.Visible:=False;
  btnValText.Visible:=False;
  btnVerText.Visible:=False;
  btnSavText.Visible:=True;
  PageControl1.TabIndex :=0;
  StatusBar1.Panels[0].Text:= 'No message';
  StatusBar1.Panels[1].Text:= '';
end;

//utk mereset program agar menjadi default
procedure TMainform.Reset1Click(Sender: TObject);
begin
  OnCreate(Sender);
end;

//perintah tombol save
```

```
procedure TMainform.btnsaveClick(Sender: TObject);
begin
  SaveDialog1.FileName := '';
  if PageControl1.TabIndex = 1 then
  begin
  if Inputbox.Text ='' then
    begin
      MessageDlg('Tidak ada yang bisa disimpan,'+#13+
                 'Silahkan mengambil file ',
                mtWarning,[mbok],0);
      exit;
    end;
    if btnenkrip.Checked = true then
      begin
      SaveDialog1.Filter := MIS_FLT;
      if SaveDialog1.Execute then
         begin
         Outputbox.Text:= SaveDialog1.FileName+SaveDialog1.DefaultExt;
         end;
      end
    else
    begin
      if btndekrip.Checked = true then
      begin
      if btnteks.Checked = true then
        SaveDialog1.Filter := TXT_FLT;
      if btngambarbmp.Checked = true then
        SaveDialog1.Filter := BIT_FLT;
      if btngambaremf.Checked = true then
        SaveDialog1.Filter := EMF_FLT;
      end;
      if SaveDialog1.Execute then
      begin
         Outputbox.Text:= SaveDialog1.FileName+SaveDialog1.DefaultExt;
      end;
    end;
  end;
  if PageControl1.TabIndex = 2 then
    begin
    if mmoTextVal.Text ='' then
    begin
      MessageDlg('Tidak ada yang dapat disimpan,'+#13+
                 'Silahkan masukkan teks',
                mtWarning,[mbok],0);
      exit;
    end;
    SaveDialog1.Filter := MIS_FLT;
    if SaveDialog1.Execute then
      begin

mmoTextVal.Lines.SaveToFile(SaveDialog1.FileName+SaveDialog1.DefaultExt);
      end;
    end;
end;

procedure TMainform.mmoTextDigChange(Sender: TObject);
var
  s : String;
begin
  s :=  mmoSignature.Lines.Text;
  CRC32Text :=  $FFFFFFFF;  //To match PKZIP
  If Length(s) > 0  //avoid access violation in D4
  then CalcCRC32(Addr(s[1]), length(s), CRC32text);
  CRC32Text :=  NOT CRC32Text;  //To match PKZIP
```

```
  CRC32 :=  CRC32Text;
end;

//inisialisasi submenu
procedure TMainform.PageControl1Change(Sender: TObject);
begin
  if PageControl1.TabIndex  =0 then
  begin
    btnClearSign.Click;
    btnteks.Visible:=False;
    btngambaremf.Visible:=False;
    btngambarbmp.Visible:=False;
    GroupBox6.Visible:=False;
    if btnenkrip.Checked=True then
    begin
    btnopenfile.Visible:=False;
    btnsave.Visible:=False;
    btnclear.Visible:=False;
    btnOpDig.Visible:=True;
    btnSavText.Visible:=True;
    btnGenerate.Visible:=True;
    btnSignature.Visible:=True;
    btnVerSign.Visible:=False;
    btnpublickey.Visible:=True;
    btnSaveSign.Visible:=True;
    btnClearSign.Visible:=True;
    btnOpenVal.Visible:=False;
    btnClearVal.Visible:=False;
    btnValText.Visible:=False;
    btnVerText.Visible:=False;
    ToolButton1.Visible:=True;
    ToolButton2.Visible:=True;
    end;
    if btndekrip.Checked=True then
    begin
    btnopenfile.Visible:=False;
    btnsave.Visible:=False;
    btnclear.Visible:=False;
    btnOpDig.Visible:=True;
    btnSavText.Visible:=False;
    btnGenerate.Visible:=False;
    btnSignature.Visible:=False;
    btnVerSign.Visible:=True;
    btnpublickey.Visible:=True;
    btnSaveSign.Visible:=True;
    btnClearSign.Visible:=True;
    btnOpenVal.Visible:=False;
    btnClearVal.Visible:=False;
    btnValText.Visible:=False;
    btnVerText.Visible:=False;
    ToolButton1.Visible:=True;
    ToolButton2.Visible:=True;
    end;
  end;

  if PageControl1.TabIndex  =1 then
  begin
    if btnenkrip.Checked=True then
    begin
    btnteks.Checked:=True;
    btnteks.Visible:=True;
    btngambarbmp.Visible:=True;
    btngambaremf.Visible:=True;
    GroupBox6.Visible:=True;
```

```
  btnclear.Click;
  btnopenfile.Visible:=True;
  btnsave.Visible:=True;
  btnclear.Visible:=True;
  btnOpDig.Visible:=False;
  btnSavText.Visible:=False;
  btnGenerate.Visible:=False;
  btnSignature.Visible:=False;
  btnVerSign.Visible:=False;
  btnpublickey.Visible:=False;
  btnSaveSign.Visible:=False;
  btnClearSign.Visible:=False;
  btnOpenVal.Visible:=False;
  btnClearVal.Visible:=False;
  btnValText.Visible:=False;
  btnVerText.Visible:=False;
  end;
  if btndekrip.Checked=True then
  begin
  btnteks.Checked:=True;
  btnteks.Visible:=False;
  btngambarbmp.Visible:=False;
  btngambaremf.Visible:=False;
  GroupBox6.Visible:=False;
  btnclear.Click;
  btnopenfile.Visible:=True;
  btnsave.Visible:=True;
  btnclear.Visible:=True;
  btnOpDig.Visible:=False;
  btnSavText.Visible:=False;
  btnGenerate.Visible:=False;
  btnSignature.Visible:=False;
  btnVerSign.Visible:=False;
  btnpublickey.Visible:=False;
  btnSaveSign.Visible:=False;
  btnClearSign.Visible:=False;
  btnOpenVal.Visible:=False;
  btnClearVal.Visible:=False;
  btnValText.Visible:=False;
  btnVerText.Visible:=False;
  end;
end;

if PageControl1.TabIndex  =2 then
begin
  btnClearVal.Click;
  btnteks.Visible:=False;
  btngambarbmp.Visible:=False;
  btngambaremf.Visible:=False;
  GroupBox6.Visible:=False;
  if btnenkrip.Checked =True then
  begin
  btnopenfile.Visible:=False;
  btnsave.Visible:=True;
  btnclear.Visible:=False;
  btnOpDig.Visible:=False;
  btnSavText.Visible:=False;
  btnGenerate.Visible:=False;
  btnSignature.Visible:=False;
  btnVerSign.Visible:=False;
  btnpublickey.Visible:=False;
  btnSaveSign.Visible:=False;
  btnClearSign.Visible:=False;
  btnOpenVal.Visible:=True;
```

```
    btnClearVal.Visible:=True;
    btnValText.Visible:=True;
    btnVerText.Visible:=False;
    ToolButton1.Visible:=False;
    ToolButton2.Visible:=False;
    ToolButton3.Visible:=True;
    end;
    if btndekrip.Checked=True then
    begin
    btnopenfile.Visible:=False;
    btnsave.Visible:=False;
    btnclear.Visible:=False;
    btnOpDig.Visible:=False;
    btnSavText.Visible:=False;
    btnGenerate.Visible:=False;
    btnSignature.Visible:=False;
    btnVerSign.Visible:=False;
    btnpublickey.Visible:=False;
    btnSaveSign.Visible:=False;
    btnClearSign.Visible:=False;
    btnOpenVal.Visible:=True;
    btnClearVal.Visible:=True;
    btnValText.Visible:=False;
    btnVerText.Visible:=True;
    ToolButton1.Visible:=False;
    ToolButton2.Visible:=False;
    ToolButton3.Visible:=True;
    end;
    if btnteks.Checked=True then
    begin
    mmoInput.Visible:=True;
    end;
    if btngambaremf.Checked=True then
    begin
    mmoInput.Visible:=False;
    end;
    if btngambarbmp.Checked=True then
    begin
    mmoInput.Visible:=False;
    end;
  end;
end;

procedure TMainform.btnteksClick(Sender: TObject);
begin
if PageControl1.TabIndex = 1 then
  begin
  mmoInput.Visible:=True;
  Image1.Visible:=False;
  end;
end;

procedure TMainform.btngambaremfClick(Sender: TObject);
begin
if PageControl1.TabIndex = 1 then
  begin
  Image1.Visible:=True;
  Image1.Picture:=nil;
  mmoInput.Visible:=False;
  if btndekrip.Checked = True then
  begin
  mmoInput.Visible:=True;
  Image1.Visible:=False;
  end;
```

```
  end;
end;

procedure TMainform.btngambarbmpClick(Sender: TObject);
begin
if PageControl1.TabIndex = 1 then
  begin
  Image1.Visible:=True;
  Image1.Picture:=nil;
  mmoInput.Visible:=False;
  if btndekrip.Checked = True then
  begin
  mmoInput.Visible:=True;
  Image1.Visible:=False;
  end;
  end;
end;
```

## MODULE DIGITAL SIGNATURE

```
//perintah tombol open text message
procedure TMainform.btnOpDigClick(Sender: TObject);
const
  BufferSize = 16384;
var
  IOBuffer  : Array of Byte;  //buffer for CRC16 computation
  Error32   : Word;
  blk       : string;
  FileBytes : TInteger8;
  bagi      : real;
begin
  mmoTextDig.ScrollBars :=  ssVertical;
  OpenDialog1.Filter := TXT1_FLT;
  OpenDialog1.FileName:='';
  if OpenDialog1.Execute then
  begin
    with OpenDialog1 do
    begin
      mmoTextDig.Lines.LoadFromFile(opendialog1.filename);
      SetLength(IOBuffer, BufferSize);
      CalcFileCRC32(OpenDialog1.FileName, CRC32File, FileBytes, Error32);
      bagi:=(FileBytes/1024);
      if bagi>1024 then
        blk:=Format('%.2f MB', [bagi/1024])
      else
        blk:=Format('%2f KB', [bagi]);
        StatusBar1.Panels[0].Text:='Total Size ='+blk;
        StatusBar1.Panels[1].Text:='';
        Screen.Cursor:=crHourGlass;
        Screen.Cursor:=crDefault;
      end;
  end;
end;

//perintah untuk menyimpan teks
procedure TMainform.btnSavTextClick(Sender: TObject);
begin
    if mmoTextDig.Text ='' then
    begin
      MessageDlg('Tidak ada yang dapat disimpan'+#13+
                 'Silahkan masukkan teks',
                mtWarning,[mbok],0);
      exit;
    end;
    SaveDialog1.Filter:=TXT1_FLT;
```

```
    if SaveDialog1.Execute then
    begin

mmoTextDig.Lines.SaveToFile(savedialog1.filename+SaveDialog1.DefaultExt)
    end;
end;

//generate pasangan kunci RSA digital signature
procedure TMainform.btnGenerateClick(Sender: TObject);
begin
  StatusBar1.Panels[1].Text := sTunggu;
  Screen.Cursor := crAppStart;
  try
    LbRSASSA1.GenerateKeyPair;
    edtPublicE.Text :=  LbRSASSA1.PublicKey.ExponentAsString;
    edtPublicM.Text :=  LbRSASSA1.PublicKey.ModulusAsString;
    edtPrivateE.Text:=  LbRSASSA1.PrivateKey.ExponentAsString;
    finally
    Screen.Cursor :=  crDefault;
    StatusBar1.Panels[1].Text :=  '';
  end;
end;

//sign message string, display signature as hex string
procedure TMainform.btnSignatureClick(Sender: TObject);
begin
  if mmoTextDig.Text='' then
  begin
      MessageDlg('No Message?',
                 mtWarning,[mbok],0);
      exit;
  end;
  if edtPublicE.Text='' then
  begin
      MessageDlg('Generate your key',
                 mtWarning,[mbok],0);
      exit;
  end;

  Screen.Cursor :=  crHourGlass;
  try
    LbRSASSA1.SignString(mmoTextDig.Lines.Text);
    mmoSignature.Text :=  LbRSASSA1.Signature.IntStr;
    MessageDlg('Message Has Been Sign'+#13+
                '',mtInformation,[mbOK],0);
    exit;
  finally
    Screen.Cursor :=  crDefault;
  end;
end;

//verify message
procedure TMainform.btnVerSignClick(Sender: TObject);
begin
  if mmoTextDig.Text ='' then
    begin
      MessageDlg('Tidak ada pesan untuk diverifikasi?',
                  mtWarning,[mbok],0);
      exit;
    end;
  if edtPublicM.Text ='' then
    begin
      MessageDlg('Please Load Public Key First!',
                 mtWarning,[mbok],0);
```

```
      exit;
    end;
    if mmoSignature.Text ='' then
    begin
      MessageDlg('Please Load Signature!',
                 mtWarning,[mbok],0);
      exit;
    end;
  if LbRSASSA1.VerifyString(mmoTextDig.Lines.Text) then
    begin
    MessageDlg('Verifikasi Tanda Tangan: SUKSES'+#13+
                'Pesan Orisinil',mtInformation,[mbOK],0);
    exit;
    StatusBar1.Panels[1].Text := sPass
    end
  else
    begin
    MessageDlg('Verifikasi Tanda Tangan: GAGAL'+#13+
                'Pesan Tidak Orisinil',mtWarning,[mbOK],0);
    exit;
    StatusBar1.Panels[1].Text := sFail;
    end
end;

//perintah untuk save public key
procedure TMainform.SavePubKeyClick(Sender: TObject);
var
  FS  : TFileStream;
begin
  if SaveDlgPub.Execute then begin
    FS := TFileStream.Create((SaveDlgPub.FileName+SaveDlgPub.DefaultExt),
fmCreate);
    Screen.Cursor := crHourGlass;
    try
      LbRSASSA1.PublicKey.StoreToStream(FS);
    finally
      FS.Free;
      Screen.Cursor := crDefault;
    end;
  end;
end;

//perintah untuk load public key
procedure TMainform.LoadPubKeyClick(Sender: TObject);
var
  FS : TFileStream;
begin
  if OpenDlgPub.Execute then begin
    FS := TFileStream.Create(OpenDlgPub.FileName, fmOpenRead);
    Screen.Cursor := crHourGlass;
    try
      LbRSASSA1.PublicKey.LoadFromStream(FS);
      edtPublicE.Text := LbRSASSA1.PublicKey.ExponentAsString;
      edtPublicM.Text := LbRSASSA1.PublicKey.ModulusAsString;
    finally
      FS.Free;
      Screen.Cursor := crDefault;
    end;
  end;
end;

//perintah untuk save signature
procedure TMainform.SaveSignClick(Sender: TObject);
var
```

```
  FS : TFileStream;
begin
  if SaveDlgSign.Execute then begin
    FS := TFileStream.Create(SaveDlgSign.FileName +
SaveDlgSign.DefaultExt, fmCreate);
    Screen.Cursor := crHourGlass;
    try
     mmoSignature.Lines.SaveToStream(fs);
   finally
      FS.Free;
      Screen.Cursor := crDefault;
    end;
  end;
end;

//perintah untuk load signature
procedure TMainform.LoadSignClick(Sender: TObject);
var
  FS : TFileStream;
begin
  if OpenDlgSign.Execute then begin
    FS := TFileStream.Create(OpenDlgSign.FileName, fmOpenRead);
    Screen.Cursor := crHourGlass;
    try
    mmoSignature.Lines.LoadFromStream(FS);
    finally
      FS.Free;
      Screen.Cursor := crDefault;
    end;
  end;
end;

//convert signature string to binary and return it
procedure TMainform.LbRSASSA1GetSignature(Sender: TObject;
  var Sig: TRSASignatureBlock);
begin
  HexToBuffer(mmoSignature.Text, Sig, SIZEOf(Sig));
end;

//clear all fields
procedure TMainform.btnClearSignClick(Sender: TObject);
begin
  LbRSASSA1.PrivateKey.Clear;
  LbRSASSA1.PublicKey.Clear;
  edtPrivateE.Text := '';
  edtPublicE.Text := '';
  edtPublicM.Text := '';
  mmoSignature.Clear;
  mmoTextDig.Clear;
  mmoTextDig.SetFocus;
  StatusBar1.Panels[0].Text:= 'No Message';
  StatusBar1.Panels[1].Text:='';
end;

procedure TMainform.LbRSASSA1Progress(Sender: TObject; var Abort:
Boolean);
begin
  Application.ProcessMessages;
end;
```

## MODULE VALIDASI

```
//perintah untuk melihat setiap perubahan pesan
procedure TMainform.mmoTextValChange(Sender: TObject);
  VAR
```

```
    s    :  STRING;
begin
  s := mmoTextVal.Lines.Text;
  CRC32Text := $FFFFFFFF;   // To match PKZIP
  IF   LENGTH(s) > 0    // Avoid access violation in D4
  THEN CalcCRC32 (Addr(s[1]), LENGTH(s), CRC32text);
  CRC32Text := NOT CRC32Text;   // TO match PKZIP

  CRC32 := CRC32Text;
  UpdateDisplay
end;

procedure TMainform.updatedisplay;
begin
  mmoValKey.Lines.Text:=IntToHex(CRC32,8)+IntToStr(CRC32);
end;

//perintah untuk membuka pesan cipher
procedure TMainform.btnOpenValClick(Sender: TObject);
  CONST BufferSize = 16384;
  VAR
    IOBuffer :  ARRAY OF BYTE;   // buffer for CRC16 computation
    Error32  :  WORD;
    FileBytes:  TInteger8;
begin
  OpenDialog1.Filter := MIS_FLT;
  OpenDialog1.FileName:='';
 IF   OpenDialog1.Execute
  THEN BEGIN
    StatusBar1.Panels[0].Text := '';
    mmoTextVal.Lines.LoadFromFile(opendialog1.filename);
    SetLength(IOBuffer, BufferSize);
    CalcFileCRC32 (OpenDialog1.FileName, CRC32File, FileBytes, Error32);
    StatusBar1.panels[1].Text := 'Size :'+ IntToStr(FileBytes) + ' bytes'
+ '    Path:   ' + OpenDialog1.FileName;
    IF   Error32 <> 0
    THEN StatusBar1.panels[0].Text := StatusBar1.panels[0].Text + ',
CRC32 error'
  END;

  CRC32 := CRC32File;
end;

procedure TMainform.btnClearValClick(Sender: TObject);
begin
  mmoTextVal.Clear;
  mmoValKey.Clear;
  mmoVerKey.Clear;
  mmoTextVal.SetFocus;
  StatusBar1.Panels[1].Text :='';
  StatusBar1.Panels[0].Text:= 'No Message';
end;

//perintah untuk memvalidasi ciphertext asal
procedure TMainform.btnValTextClick(Sender: TObject);
begin
    if mmoTextVal.Lines.Text ='' then
    begin
      MessageDlg('Tidak ada yang dapat diproses !'+#13+ 'Coba lagi
?',mtWarning,[mbRetry],0);
      exit;
    end
    else
    begin
```

```
       if  MessageDlg('Now Save your Validation Key ?
',mtConfirmation,[mbYes],0) = mrYes then
       begin
              SaveDialog1.Filter := VAL_FLT;
              SaveDialog1.FileName:='';
              if SaveDialog1.Execute then
              mmoValKey.Lines.SaveToFile(savedialog1.filename +
SaveDialog1.DefaultExt)
              else exit;
      end;
    end;
  MessageDlg('Kunci validasi telah disimpan! ',mtConfirmation,[mbYes],0)
;
end;

//perintah untuk memverifikasi apakah ciphertext sama
procedure TMainform.btnVerTextClick(Sender: TObject);
begin
  if mmoValKey.Lines.Text='' then
  begin
      MessageDlg('Silahkan masukkan pesan',mtWarning,[mbOK],0);
      exit
  end
  else
  begin
    if mmoVerKey.Lines.Text = '' then
    begin
      OpenDialog1.Filter := VAL_FLT;
      OpenDialog1.FileName:='';
      if OpenDialog1.Execute then
      begin
      mmoVerKey.Lines.LoadFromFile(OpenDialog1.FileName);
      end;
    end;
    if mmoValKey.Lines.Text = mmoVerKey.Lines.Text then
    begin
      MessageDlg('Verifikasi SUKSES'+#13+
             'Pesan Yang Anda Terima Orisinil',mtInformation,[mbYes],0)
    end
    else
        MessageDlg('Verifikasi GAGAL'+#13+
             'Pesan Yang Anda Terima Tidak
Orisinil!!!',mtWarning,[mbOK],0 )
  end;
end;
```

## MODUL ENKRIP DAN DEKRIP

```
//perintah  untuk enkripsi dan dekripsi
procedure TMainform.btnGoClick(Sender: TObject);
var
  FileIn, FileOut: file;
  Buffer: array[0..7] of byte;
  Hash: Tsha1;
  HashDigest: array[0..31] of byte;
  cipherenkrip: Cmisty1;
  cipherdekrip: Cmisty1;
  Read: integer;
  keystr: string;
  HashRead: array[0..31] of byte;

begin
  if Inputbox.Text ='' then
    begin
    MessageDlg('Masukkan file yang akan dienkripsi/didekripsi'+#13+
```

```
              'Tekan tombol open untuk membuka file',mtWarning,[mbok],0);
    exit;
    end;
  if Outputbox.Text='' then
    begin
    MessageDlg('File output yang diinginkan ?'+#13+
              'Tekan tombol save',mtWarning,[mbok],0);
    exit;
    end;

  //perintah utama pemanggil enkripsi ke unit MISTY1
  if btnenkrip.Checked =True then
  begin
    keystr:= '';
    with FormPass do
    begin
      edtpass.Clear;
      ShowModal;
      keystr:=edtpass.Text;
      if keystr='' then exit;
    end;

  AssignFile(FileIn,Inputbox.Text);
  begin
    if FileExists(Outputbox.Text) then
    if (MessageDlg('Output file already exists.
Overwrite?',mtConfirmation,mbYesNoCancel,0) <> mrYes) then
      Exit;
    try
    Reset(FileIn,1);
  except
    MessageDlg('Unable to open the in file',mtInformation,[mbOK],0);
    Exit;
  end;
  AssignFile(FileOut,Outputbox.Text);
  try
    Rewrite(FileOut,1);
  except
    CloseFile(FileIn);
    MessageDlg('Unable to open then out file',mtInformation,[mbOK],0);
    Exit;
  end;
  Screen.Cursor := crAppStart;
  ref :=GetTickCount;
  // baca file input
  FillChar(HashDigest,Sizeof(HashDigest),$FF);
  Hash:= Tsha1.Create(Self);
  Hash.Init;
  //ambil isi password untuk dijadikan kunci
  Hash.UpdateStr(keystr);
  Hash.Final(HashDigest);
  Hash.Free;
  cipherenkrip:= Cmisty1.Create(Self);
  if (Sizeof(HashDigest)*8)> cipherenkrip.MaxKeySize then
     cipherenkrip.Init(HashDigest,cipherenkrip.MaxKeySize,nil)
  else
  cipherenkrip.Init(keystr,Sizeof(keystr)*8,nil);
  cipherenkrip.EncryptData(HashDigest,HashDigest,Sizeof(HashDigest));
  cipherenkrip.Reset;
  BlockWrite(FileOut,HashDigest,Sizeof(HashDigest));
  repeat
    BlockRead(FileIn,Buffer,Sizeof(Buffer),Read);
    //baca file input, enkrip lalu tulis ke file output
    cipherenkrip.EncryptData(Buffer,Buffer,Read);
```

```
    cipherenkrip.Reset;
    BlockWrite(FileOut,Buffer,Read);
  until Read<> Sizeof(Buffer);
  cipherenkrip.Burn; //lalukan prosedur enkrip
  CloseFile(FileIn);
  CloseFile(FileOut);
      Image1.Picture:=nil;
      mmoInput.Visible:=True;
      mmoInput.Clear;
      mmoInput.Lines.LoadFromFile(Outputbox.Text);
      ref :=GetTickCount -ref;
      Screen.Cursor := crDefault;
      StatusBar1.Panels[1].Text:=Format('Encrypting in %2.3f s.', [ref /
1000.0]);
      MessageDlg('File encrypted done',mtConfirmation,[mbOK],0);
      Inputbox.Text:='';
      end;
  end;

  if btndekrip.Checked = True  then
  begin
   keystr:= '';
    with FormPass do
    begin
      edtpass.Clear;
      ShowModal;
      keystr:=edtpass.Text;
      Close;
    end;
 begin
   if FileExists(Outputbox.Text) then
    if (MessageDlg('Output file already exists.
Overwrite?',mtConfirmation,mbYesNoCancel,0) <> mrYes) then
      Exit;
    AssignFile(FileIn,Inputbox.Text);
   try
    Reset(FileIn,1);
    except
    MessageDlg('Unable to open the in file',mtInformation,[mbOK],0);
    Exit;
  end;
  AssignFile(FileOut,Outputbox.Text);
  try
    Rewrite(FileOut,1);
    except
    CloseFile(FileIn);
    MessageDlg('Unable to open the out file',mtInformation,[mbOK],0);
    Exit;
  end;
  Screen.Cursor := crAppStart;
  ref :=GetTickCount;
  // baca file input
  FillChar(HashDigest,Sizeof(HashDigest),$FF);
  Hash:= Tsha1.Create(Self);
  Hash.Init;
  //ambil isi password untuk dijadikan kunci
  Hash.UpdateStr(keystr);
  Hash.Final(HashDigest);
  Hash.Free;
  cipherdekrip:=Cmisty1.Create(Self);
  if (Sizeof(HashDigest)*8)> cipherdekrip.MaxKeySize then
    cipherdekrip.Init(HashDigest,cipherdekrip.MaxKeySize,nil)
  else
  cipherdekrip.Init(keystr,Sizeof(keystr)*8,nil);
```

```
  cipherdekrip.EncryptData(HashDigest,HashDigest,Sizeof(HashDigest));
  cipherdekrip.Reset;
  BlockRead(FileIn,HashRead,Sizeof(HashRead));
  //pengecekan password yang digunakan
  if not CompareMem(@HashRead,@HashDigest,Sizeof(HashRead)) then
  begin
    CloseFile(FileIn);
    CloseFile(FileOut);
    cipherdekrip.Burn;  //lalukan prosedur dekrip
    MessageDlg('Proses Gagal...,'+#13+
               'Password Salah !',mtWarning,[mbOK],0);
    Screen.Cursor := crDefault;
    Exit;
  end;
  repeat
    BlockRead(FileIn,Buffer,Sizeof(Buffer),Read);
    //baca file input, dekrip lalu tulis ke file output
    cipherdekrip.DecryptData(Buffer,Buffer,Read);
    cipherdekrip.Reset;
    BlockWrite(FileOut,Buffer,Read);
  until Read<> Sizeof(Buffer);
  cipherdekrip.Burn;
  CloseFile(FileIn);
  CloseFile(FileOut);
      if btnteks.Checked=True then
        begin
        mmoInput.Clear;
        mmoInput.Lines.LoadFromFile(Outputbox.Text);
        end;
      if btngambaremf.Checked=True then
        begin
        mmoInput.Visible:=False;
        Image1.Visible:=True;
        Image1.Picture.Metafile.LoadFromFile(Outputbox.Text);
        end;
      if btngambarbmp.Checked=True then
        begin
        mmoInput.Visible:=False;
        Image1.Visible:=True;
        Image1.Picture.Bitmap.LoadFromFile(Outputbox.Text);
        end;
      ref :=GetTickCount -ref;
      Screen.Cursor := crDefault;
      StatusBar1.Panels[1].Text:=Format('Encrypting in %2.3f s.', [ref /
1000.0]);
      MessageDlg('File decrypted done',mtConfirmation,[mbOK],0);
      Inputbox.Text:='';
    end;
    end;
//perintah untuk menghapus ciphertext setelah proses dekripsi
  if chkfiledel.Checked then
  DeleteFile(Inputbox.Text);
end;
```

## UNIT MISTY1

```
//isi dari algoritma MISTY1

unit Misty1;

interface
{$I crypt.Inc}
uses
  Classes, Sysutils, crypt;
```

```
const
  NUMROUNDS= 8;


type
  Cmisty1= class(Tblockcipher)
  protected
    IV, LB: array[0..7] of byte;
    KeyData: array[0..31] of DWord;
    function FI(const FI_IN, FI_KEY: DWord): DWord;
    function FO(const FO_IN: DWord; const k: longint): DWord;
    function FL(const FL_IN: DWord; const k: longint): DWord;
    function FLINV(const FL_IN: DWord; const k: longint): DWord;
    procedure Encrypt(const InBlock; var OutBlock);
    procedure Decrypt(const InBlock; var OutBlock);
  public
    procedure Init(var Key; Size: longint; IVector: pointer); override;
    procedure Burn; override;
    procedure Reset; override;
    procedure EncryptData(const InData; var OutData; Size: longint);
override;
    procedure DecryptData(const InData; var OutData; Size: longint);
override;
    constructor Create(AOwner: TComponent); override;
  end;


implementation

{$I Misty1.Inc}


constructor Cmisty1.Create(AOwner: TComponent);
begin
  inherited Create(AOwner);
  fAlgorithm:= 'Misty1';
  fBlockSize:= 64;
  fMaxKeySize:= 128;
  fID:= 11;
  Burn;
end;



function Cmisty1.FI(const FI_IN, FI_KEY: DWord): DWord;
var
  d7, d9: DWord;
begin
  d9:= (FI_IN shr 7) and $1ff;
  d7:= FI_IN and $7f;
  d9:= S9Table[d9] xor d7;
  d7:= (S7Table[d7] xor d9) and $7f;
  d7:= d7 xor ((FI_KEY shr 9) and $7f);
  d9:= d9 xor (FI_KEY and $1ff);
  d9:= S9Table[d9] xor d7;
  Result:= (d7 shl 9) or d9;
end;


function Cmisty1.FO(const FO_IN: DWord; const k: longint): DWord;
var
  t0, t1: DWord;
begin
  t0:= FO_IN shr 16;
```

```
  t1:= FO_IN and $FFFF;
  t0:= t0 xor KeyData[k];
  t0:= FI(t0,KeyData[((k+5) mod 8) + 8]);
  t0:= t0 xor t1;
  t1:= t1 xor KeyData[(k+2) mod 8];
  t1:= FI(t1,KeyData[((k+1) mod 8) + 8]);
  t1:= t1 xor t0;
  t0:= t0 xor KeyData[(k+7) mod 8];
  t0:= FI(t0,KeyData[((k+3) mod 8) + 8]);
  t0:= t0 xor t1;
  t1:= t1 xor KeyData[(k+4) mod 8];
  Result:= (t1 shl 16) or t0;
end;


function Cmisty1.FL(const FL_IN: DWord; const k: longint): DWord;
var
  d0, d1: DWord;
  t: byte;
begin
  d0:= FL_IN shr 16;
  d1:= FL_IN and $FFFF;
  if (k mod 2)<> 0 then
  begin
    t:= (k-1) div 2;
    d1:= d1 xor (d0 and KeyData[((t + 2) mod 8) + 8]);
    d0:= d0 xor (d1 or KeyData[(t + 4) mod 8]);
  end
  else
  begin
    t:= k div 2;
    d1:= d1 xor (d0 and KeyData[t]);
    d0:= d0 xor (d1 or KeyData[((t+6) mod 8) + 8]);
  end;
  Result:= (d0 shl 16) or d1;
end;


function Cmisty1.FLINV(const FL_IN: DWord; const k: longint): DWord;
var
  d0, d1: DWord;
  t: byte;
begin
  d0:= FL_IN shr 16;
  d1:= FL_IN and $FFFF;
  if (k mod 2)<> 0 then
  begin
    t:= (k-1) div 2;
    d0:= d0 xor (d1 or KeyData[(t+4) mod 8]);
    d1:= d1 xor (d0 and KeyData[((t+2) mod 8) + 8]);
  end
  else
  begin
    t:= k div 2;
    d0:= d0 xor (d1 or KeyData[((t+6) mod 8) + 8]);
    d1:= d1 xor (d0 and KeyData[t]);
  end;
  Result:= (d0 shl 16) or d1;
end;

procedure Cmisty1.Encrypt(const InBlock; var OutBlock);
var
  d0, d1: DWord;
  i: longint;
```

```
begin
  Move(InBlock,d0,4);
  Move(pointer(longint(@InBlock)+4)^,d1,4);
  for i:= 0 to NUMROUNDS-1 do
  begin
    if (i mod 2)= 0 then
    begin
      d0:= FL(D0,i);
      d1:= FL(D1,i+1);
      d1:= d1 xor FO(d0,i);
    end
    else
      d0:= d0 xor FO(d1,i);
  end;
  d0:= FL(d0,NUMROUNDS);
  d1:= FL(d1,NUMROUNDS+1);
  Move(d1,OutBlock,4);
  Move(d0,pointer(longint(@OutBlock)+4)^,4);
end;


procedure Cmisty1.Decrypt(const InBlock; var OutBlock);
var
  d0, d1: DWord;
  i: longint;
begin
  Move(InBlock,d1,4);
  Move(pointer(longint(@InBlock)+4)^,d0,4);
  d1:= FLINV(d1,NUMROUNDS+1);
  d0:= FLINV(d0,NUMROUNDS);
  for i:= NUMROUNDS-1 downto 0 do
  begin
    if (i mod 2)= 0 then
    begin
      d1:= d1 xor FO(d0,i);
      d0:= FLINV(D0,i);
      d1:= FLINV(D1,i+1);
    end
    else
      d0:= d0 xor FO(d1,i);
  end;
  Move(d0,OutBlock,4);
  Move(d1,pointer(longint(@OutBlock)+4)^,4);
end;


procedure Cmisty1.Init(var Key; Size: longint; IVector: pointer);
var
  KeyB: array[0..15] of byte;
  i: longint;
begin
  if fInitialized then
    Burn;
  if (Size> fMaxKeySize) or (Size<= 0) or ((Size mod 8)<> 0) then
    raise Exception.Create(Format('Misty1: Invalid key size -
%d',[Size]));

  if (Size> 128) or (Size<= 0) or ((Size mod 8)<> 0) then
    Exit;
  FillChar(KeyB,Sizeof(KeyB),0);
  Move(Key,KeyB,Size div 8);
  for i:= 0 to 7 do
    KeyData[i]:= (KeyB[i*2] * 256) + KeyB[i*2+1];
  for i:= 0 to 7 do
```

```
  begin
    KeyData[i+8]:= FI(KeyData[i],KeyData[(i+1) mod 8]);
    KeyData[i+16]:= KeyData[i+8] and $1FF;
    KeyData[i+24]:= KeyData[i+8] shr 9;
  end;

  if IVector= nil then
  begin
    FillChar(IV,Sizeof(IV),$FF);
    Encrypt(IV,IV);
    Move(IV,LB,Sizeof(LB));
  end
  else
  begin
    Move(IVector^,IV,Sizeof(IV));
    Move(IV,LB,Sizeof(IV));
  end;
  fInitialized:= true;
end;

procedure Cmisty1.Burn;
begin
  FillChar(KeyData,Sizeof(KeyData),$FF);
  FillChar(IV,Sizeof(IV),$FF);
  FillChar(LB,Sizeof(LB),$FF);
  fInitialized:= false;
end;

procedure Cmisty1.Reset;
begin
  Move(IV,LB,Sizeof(LB));
end;

procedure Cmisty1.EncryptData(const InData; var OutData; Size: longint);
var
  TB: array[0..7] of byte;
  i: longint;
begin
  if not fInitialized then
    raise Exception.Create('Misty1: Not initialized');
  for i:= 1 to (Size div 8) do
  begin
    XorBlock(pointer(longint(@InData)+((i-1)*8)),@LB,@TB,Sizeof(TB));
    Encrypt(TB,TB);
    Move(TB,pointer(longint(@OutData)+((i-1)*8))^,Sizeof(TB));
    Move(TB,LB,Sizeof(TB));
  end;
  if (Size mod 8)<> 0 then
  begin
    Encrypt(LB,TB);
    XorBlock(@TB,@pointer(longint(@InData)+Size-(Size mod
8))^,@pointer(longint(@OutData)+Size-(Size mod 8))^,Size mod 8);
  end;
  FillChar(TB,Sizeof(TB),$FF);
end;

procedure Cmisty1.DecryptData(const InData; var OutData; Size: longint);
var
  TB: array[0..7] of byte;
  i: longint;
begin
  if not fInitialized then
    raise Exception.Create('Misty1: Not initialized');
  for i:= 1 to (Size div 8) do
```

```
  begin
    Move(pointer(longint(@InData)+((i-1)*8))^,TB,Sizeof(TB));
    Decrypt(pointer(longint(@InData)+((i-
1)*8))^,pointer(longint(@OutData)+((i-1)*8))^);
    XorBlock(@LB,pointer(longint(@OutData)+((i-
1)*8)),pointer(longint(@OutData)+((i-1)*8)),Sizeof(TB));
    Move(TB,LB,Sizeof(TB));
  end;
  if (Size mod 8)<> 0 then
  begin
    Encrypt(LB,TB);
    XorBlock(@TB,@pointer(longint(@InData)+Size-(Size mod
8))^,@pointer(longint(@OutData)+Size-(Size mod 8))^,Size mod 8);
  end;
  FillChar(TB,Sizeof(TB),$FF);
end;

end.
```

## FORM ABOUT

```
unit FrAbout;

interface

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

type
  TFormAbout = class(TForm)
    Label1: TLabel;
    Button1: TButton;
    Panel1: TPanel;
    Label2: TLabel;
    Label3: TLabel;
    Label5: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormAbout: TFormAbout;

implementation

{$R *.dfm}

procedure TFormAbout.Button1Click(Sender: TObject);
begin
close;
end;

end.
```

## FORM HOW TO

```
unit FrHowTo;

interface
```

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

type
  TFormHowTo = class(TForm)
    Panel1: TPanel;
    RichEdit1: TRichEdit;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormHowTo: TFormHowTo;

implementation

{$R *.dfm}


procedure TFormHowTo.Button1Click(Sender: TObject);
begin
  Close;
end;

end.
```

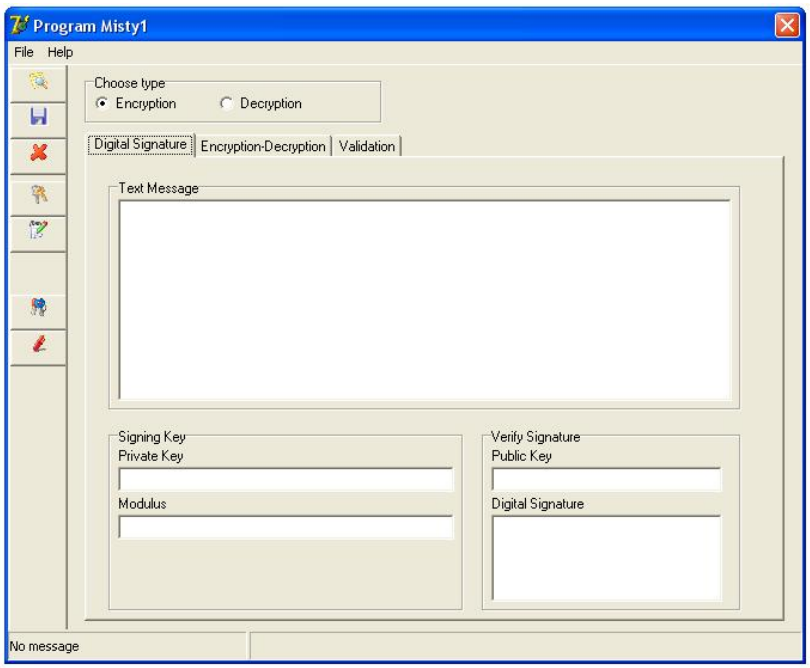
## FORM PASSWORD

```
unit FrPass;

interface

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

type
  TFormPass = class(TForm)
    GroupBox1: TGroupBox;
    edtpass: TEdit;
    chkmask: TCheckBox;
    BitBtn1: TBitBtn;
    procedure chkmaskClick(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FormPass: TFormPass;

implementation

{$R *.dfm}

procedure TFormPass.chkmaskClick(Sender: TObject);
```

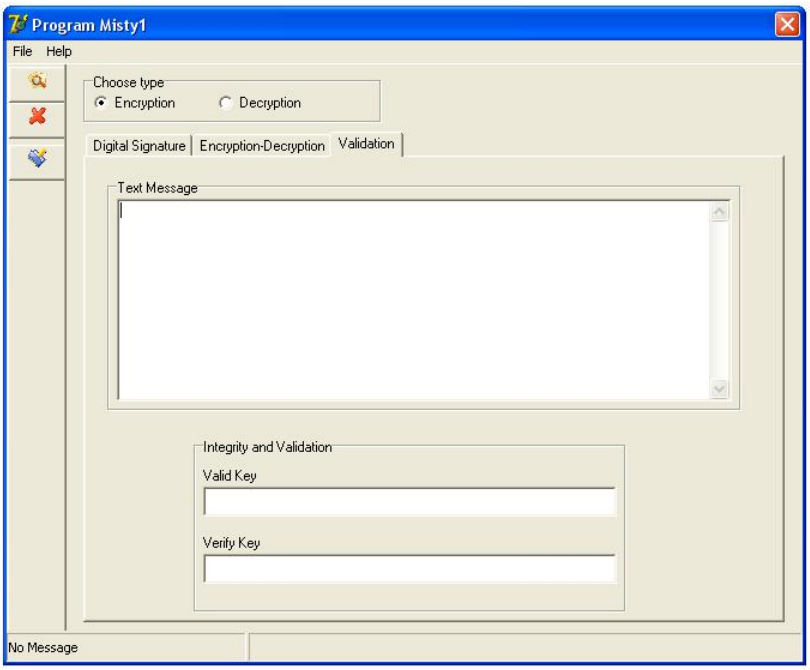
```
begin
    if chkmask.Checked =true then
    edtpass.PasswordChar:='*'
  else
    edtpass.PasswordChar:=#0;
    edtpass.SetFocus;
end;

procedure TFormPass.BitBtn1Click(Sender: TObject);
begin
 if edtpass.Text ='' then
  begin
  MessageDlg('Masukkan password !!' ,mtWarning,[mbOk],0);
  edtpass.SetFocus;
  end
 else close;
end;

end.
```

# LAMPIRAN B

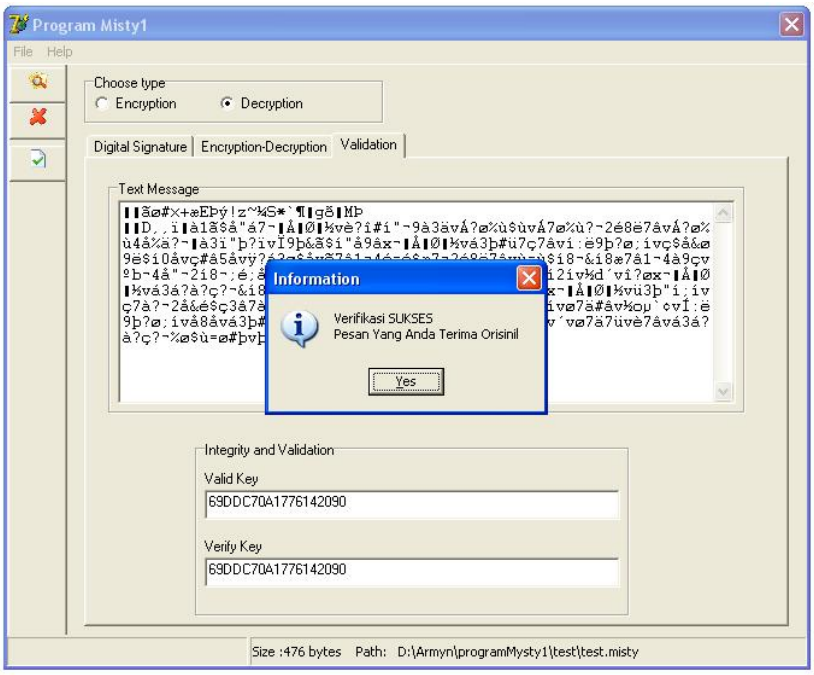
# TAMPILAN PROGRAM



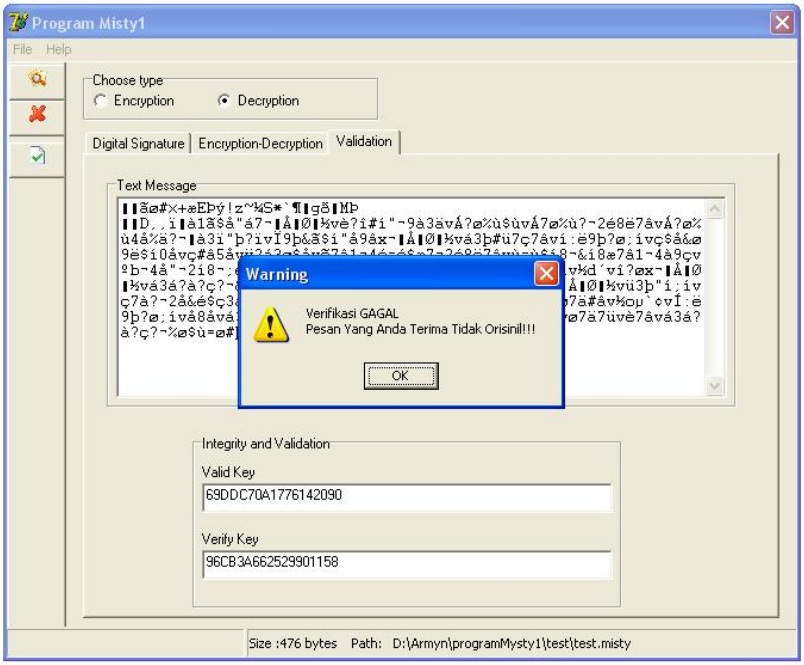

Tampilan awal program pengamanan data



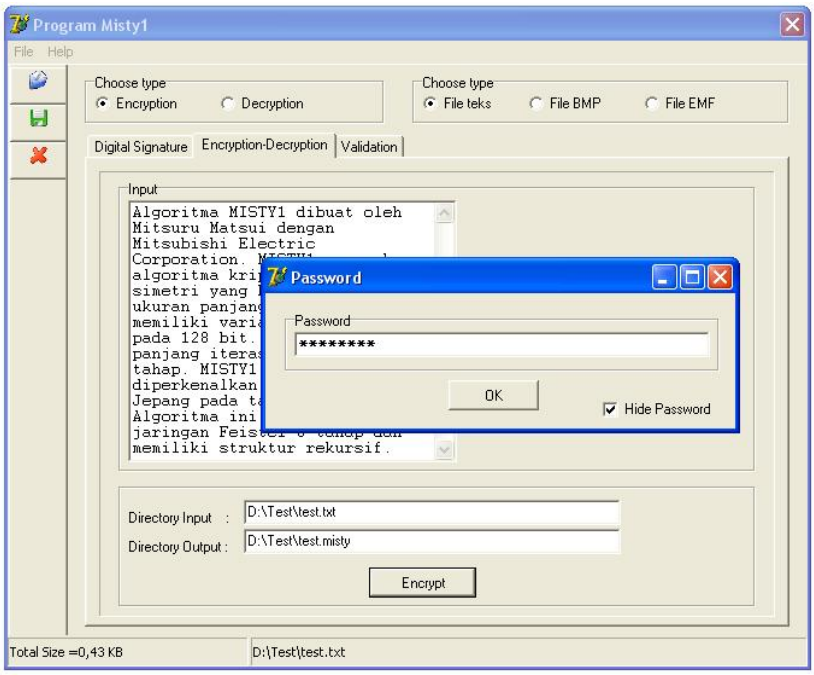

Tampilan proses validasi



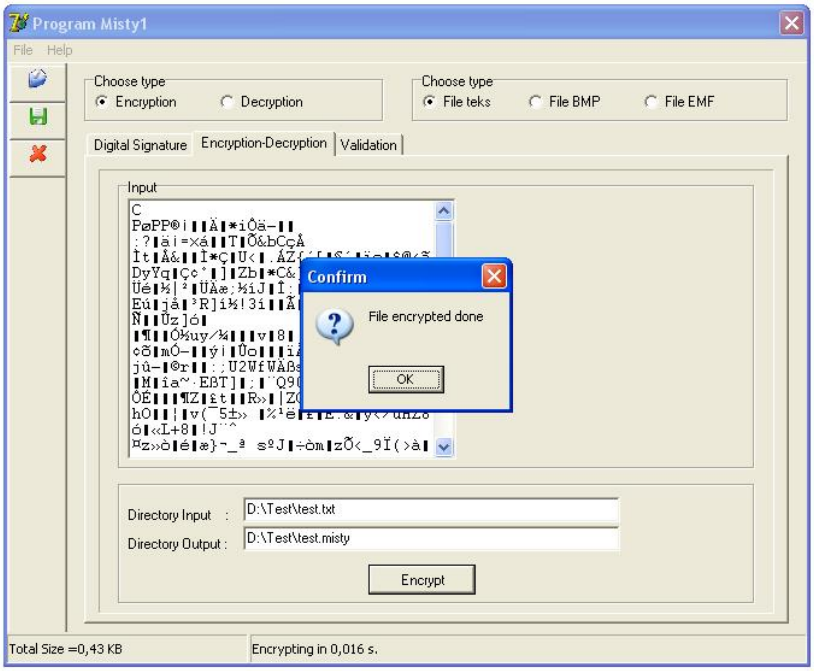

Tampilan akhir proses validasi



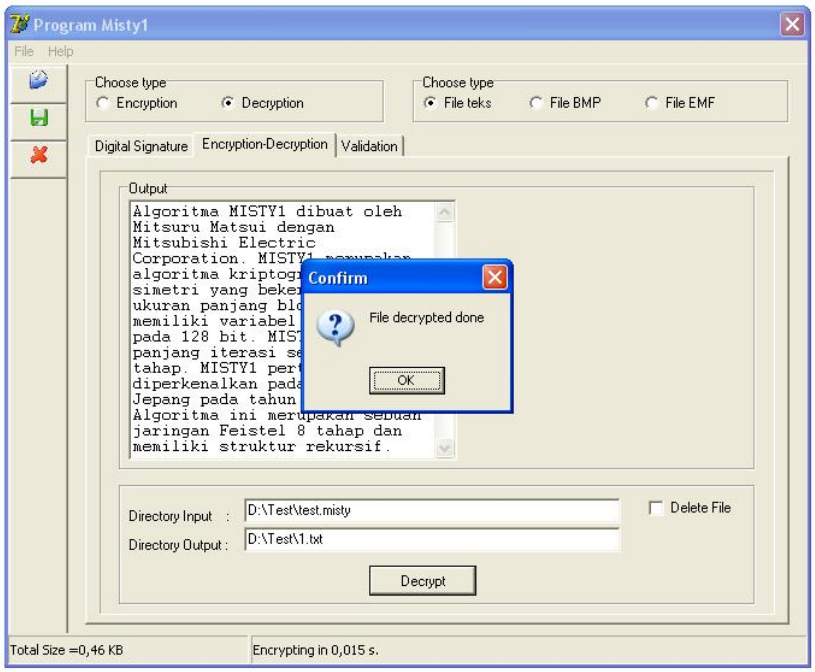

Tampilan akhir proses validasi



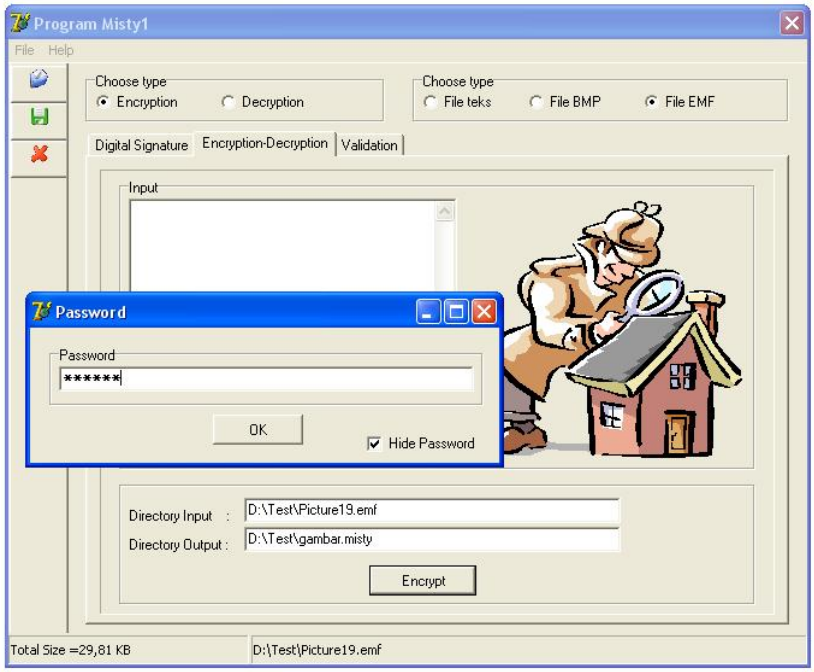

Tampilan proses enkripsi file teks



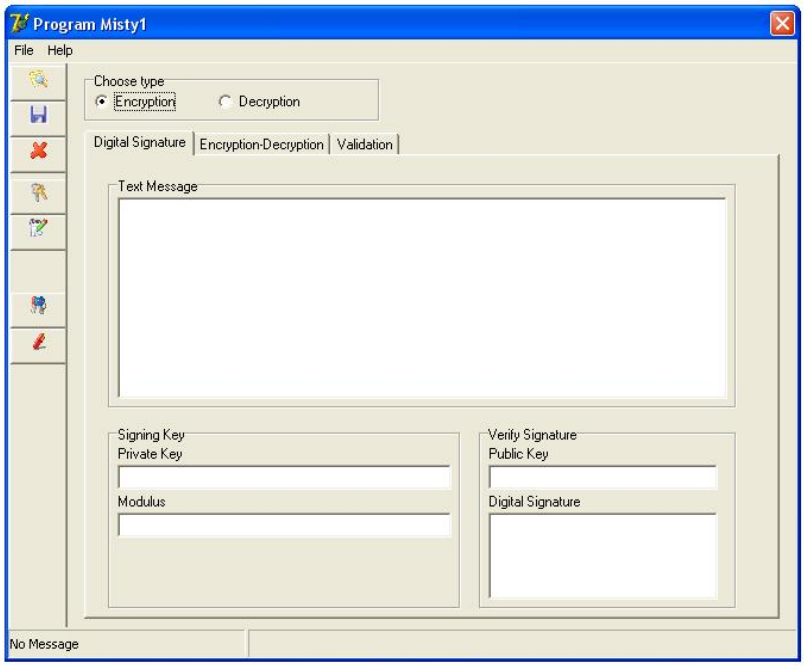

Tampilan akhir proses enkripsi file teks



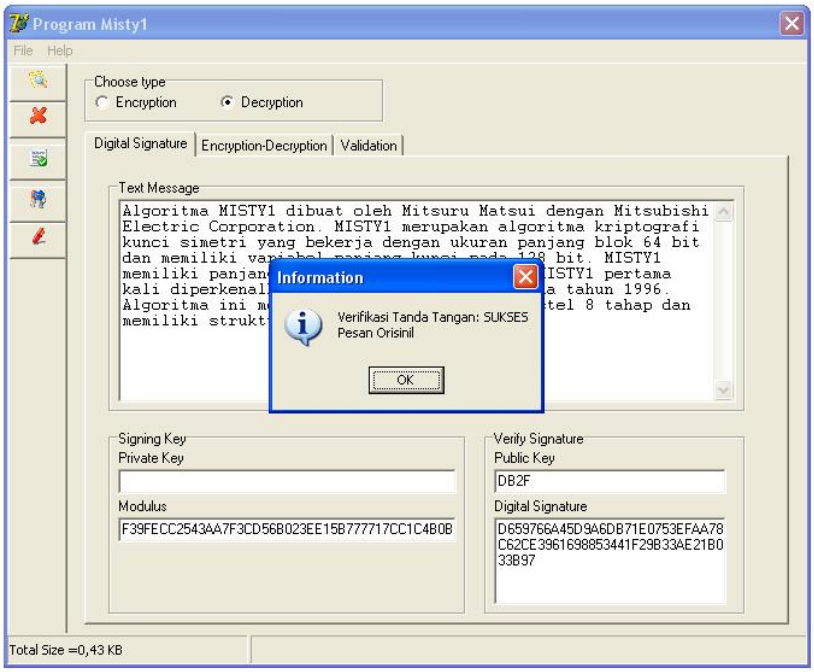

Tampilan akhir proses dekripsi file



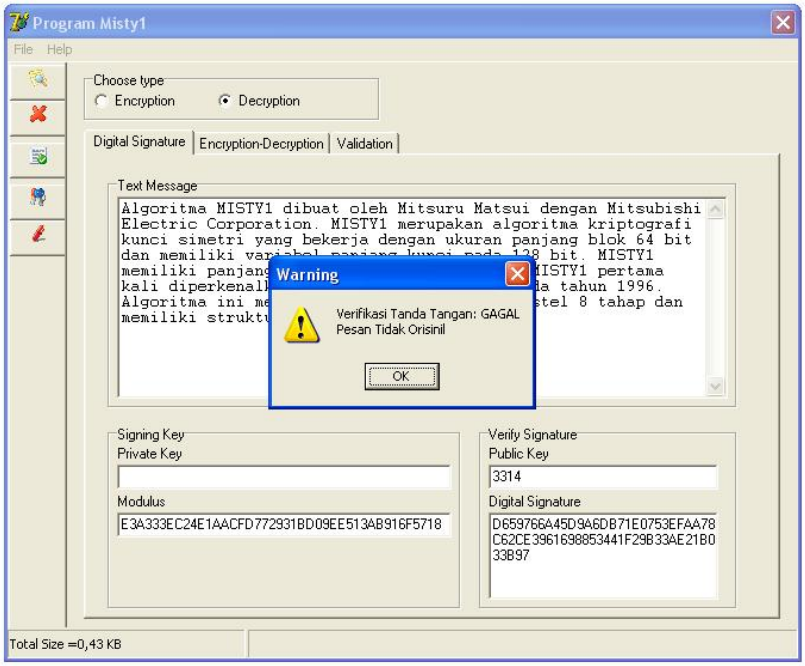

Tampilan proses enkripsi file gambar bitmap



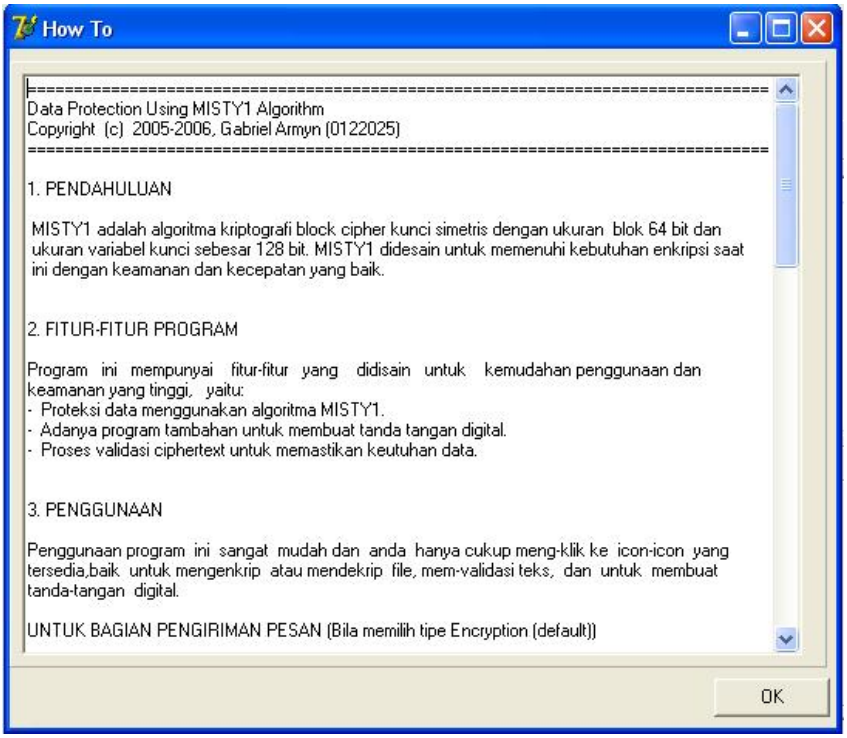

Tampilan proses digital signature



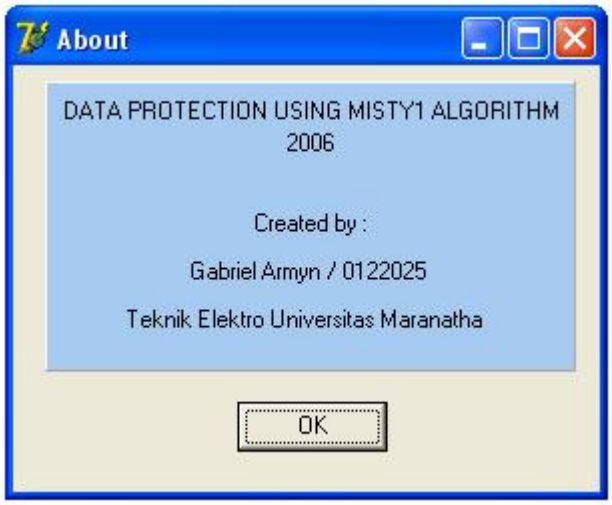

Tampilan akhir proses digital signature




Tampilan akhir proses digital signature




Tampilan menu How To




Tampilan menu About

# LAMPIRAN C

# TABEL CRC

- **Tabel untuk CRC-32**

```
CONST
    table:  ARRAY[0..255] OF DWORD =
   ($00000000, $77073096, $EE0E612C, $990951BA,
    $076DC419, $706AF48F, $E963A535, $9E6495A3,
    $0EDB8832, $79DCB8A4, $E0D5E91E, $97D2D988,
    $09B64C2B, $7EB17CBD, $E7B82D07, $90BF1D91,
    $1DB71064, $6AB020F2, $F3B97148, $84BE41DE,
    $1ADAD47D, $6DDDE4EB, $F4D4B551, $83D385C7,
    $136C9856, $646BA8C0, $FD62F97A, $8A65C9EC,
    $14015C4F, $63066CD9, $FA0F3D63, $8D080DF5,
    $3B6E20C8, $4C69105E, $D56041E4, $A2677172,
    $3C03E4D1, $4B04D447, $D20D85FD, $A50AB56B,
    $35B5A8FA, $42B2986C, $DBBBC9D6, $ACBCF940,
    $32D86CE3, $45DF5C75, $DCD60DCF, $ABD13D59,
    $26D930AC, $51DE003A, $C8D75180, $BFD06116,
    $21B4F4B5, $56B3C423, $CFBA9599, $B8BDA50F,
    $2802B89E, $5F058808, $C60CD9B2, $B10BE924,
    $2F6F7C87, $58684C11, $C1611DAB, $B6662D3D,

    $76DC4190, $01DB7106, $98D220BC, $EFD5102A,
    $71B18589, $06B6B51F, $9FBFE4A5, $E8B8D433,
    $7807C9A2, $0F00F934, $9609A88E, $E10E9818,
    $7F6A0DBB, $086D3D2D, $91646C97, $E6635C01,
    $6B6B51F4, $1C6C6162, $856530D8, $F262004E,
    $6C0695ED, $1B01A57B, $8208F4C1, $F50FC457,
    $65B0D9C6, $12B7E950, $8BBEB8EA, $FCB9887C,
    $62DD1DDF, $15DA2D49, $8CD37CF3, $FBD44C65,
    $4DB26158, $3AB551CE, $A3BC0074, $D4BB30E2,
    $4ADFA541, $3DD895D7, $A4D1C46D, $D3D6F4FB,
    $4369E96A, $346ED9FC, $AD678846, $DA60B8D0,
    $44042D73, $33031DE5, $AA0A4C5F, $DD0D7CC9,
    $5005713C, $270241AA, $BE0B1010, $C90C2086,
    $5768B525, $206F85B3, $B966D409, $CE61E49F,
    $5EDEF90E, $29D9C998, $B0D09822, $C7D7A8B4,
    $59B33D17, $2EB40D81, $B7BD5C3B, $C0BA6CAD,

    $EDB88320, $9ABFB3B6, $03B6E20C, $74B1D29A,
    $EAD54739, $9DD277AF, $04DB2615, $73DC1683,
    $E3630B12, $94643B84, $0D6D6A3E, $7A6A5AA8,
    $E40ECF0B, $9309FF9D, $0A00AE27, $7D079EB1,
    $F00F9344, $8708A3D2, $1E01F268, $6906C2FE,
    $F762575D, $806567CB, $196C3671, $6E6B06E7,
    $FED41B76, $89D32BE0, $10DA7A5A, $67DD4ACC,
    $F9B9DF6F, $8EBEEFF9, $17B7BE43, $60B08ED5,
    $D6D6A3E8, $A1D1937E, $38D8C2C4, $4FDFF252,
    $D1BB67F1, $A6BC5767, $3FB506DD, $48B2364B,
    $D80D2BDA, $AF0A1B4C, $36034AF6, $41047A60,
    $DF60EFC3, $A867DF55, $316E8EEF, $4669BE79,
    $CB61B38C, $BC66831A, $256FD2A0, $5268E236,
```

```
$CC0C7795, $BB0B4703, $220216B9, $5505262F,
$C5BA3BBE, $B2BD0B28, $2BB45A92, $5CB36A04,
$C2D7FFA7, $B5D0CF31, $2CD99E8B, $5BDEAE1D,

$9B64C2B0, $EC63F226, $756AA39C, $026D930A,
$9C0906A9, $EB0E363F, $72076785, $05005713,
$95BF4A82, $E2B87A14, $7BB12BAE, $0CB61B38,
$92D28E9B, $E5D5BE0D, $7CDCEFB7, $0BDBDF21,
$86D3D2D4, $F1D4E242, $68DDB3F8, $1FDA836E,
$81BE16CD, $F6B9265B, $6FB077E1, $18B74777,
$88085AE6, $FF0F6A70, $66063BCA, $11010B5C,
$8F659EFF, $F862AE69, $616BFFD3, $166CCF45,
$A00AE278, $D70DD2EE, $4E048354, $3903B3C2,
$A7672661, $D06016F7, $4969474D, $3E6E77DB,
$AED16A4A, $D9D65ADC, $40DF0B66, $37D83BF0,
$A9BCAE53, $DEBB9EC5, $47B2CF7F, $30B5FFE9,
$BDBDF21C, $CABAC28A, $53B39330, $24B4A3A6,
$BAD03605, $CDD70693, $54DE5729, $23D967BF,
$B3667A2E, $C4614AB8, $5D681B02, $2A6F2B94,
$B40BBE37, $C30C8EA1, $5A05DF1B, $2D02EF8D);
```