LAMPIRAN A

LISTING PROGRAM

***Form Main Program (Program Pengaman Data)**

```
unit UMain;

interface

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

type
  TMain = class(TForm)
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    About1: TMenuItem;
    Exit1: TMenuItem;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    Label1: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    btBrowse: TButton;
    btGen: TButton;
    GroupBox1: TGroupBox;
    Label8: TLabel;
    Label9: TLabel;
    PrivD: TEdit;
    PrivN: TEdit;
    GroupBox2: TGroupBox;
    Label10: TLabel;
    Label11: TLabel;
    PubE: TEdit;
    PubN: TEdit;
    edStart: TEdit;
    edDuration: TEdit;
    edEnd: TEdit;
    edPath: TEdit;
    btEncrypt: TButton;
    RPlain: TRichEdit;
```

```
  edPass: TEdit;
  TabSheet2: TTabSheet;
  Label12: TLabel;
  Label13: TLabel;
  Label14: TLabel;
  Label15: TLabel;
  Label16: TLabel;
  Label17: TLabel;
  edPath1: TEdit;
  btBrowse1: TButton;
  edPass1: TEdit;
  edStart1: TEdit;
  edEnd1: TEdit;
  edDuration1: TEdit;
  GroupBox3: TGroupBox;
  Label18: TLabel;
  Label19: TLabel;
  PubE1: TEdit;
  PubN1: TEdit;
  btDecrypt: TButton;
  OpenDlgEnc: TOpenDialog;
  SaveDlgEnc: TSaveDialog;
  SaveDlgV: TSaveDialog;
  OpenDlgV: TOpenDialog;
  OpenDlgDec: TOpenDialog;
  SaveDlgDec: TSaveDialog;
  RichEdit1: TRichEdit;
  RCipher: TRichEdit;
  RichEdit2: TRichEdit;
  Button1: TButton;
  ESave: TSaveDialog;
  Button2: TButton;
  procedure Exit1Click(Sender: TObject);
  procedure btGenClick(Sender: TObject);
  procedure btEncryptClick(Sender: TObject);
  procedure btBrowseClick(Sender: TObject);
  procedure btBrowse1Click(Sender: TObject);
  procedure btDecryptClick(Sender: TObject);
  procedure About1Click(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
```

```
  end;

var
  Main: TMain;

implementation

uses RC2, RSATools, UAbout;

{$R *.dfm}

procedure TMain.Exit1Click(Sender: TObject);
begin
Application.Terminate;
end;

procedure TMain.btGenClick(Sender: TObject);
var p : Longint;  //random prime
    q : Longint;  //second random prime that not equal to p
    n : Longint;  //p * q
    pi : Longint; //(p - 1)(q - 1)
    e : Longint;  //e that relatively prime to pi but less than pi
    d : Longint;  //d that d*e congruent to 1 mod pi
    i1 : Longint; //counter

    c : Longint;
    temp2 : array of Longint; // temp dynamic array handler that hand selection of e
    temp3 : Longint;
    temp4 : Longint;
    temp5 : Longint; // temp handler
    temp6 : Longint; // temp handler 2
label lagi,ulang;

begin
lagi:
    p := RdmPrime;
    q := RdmPrime;

    //Trap handler if p = q
    If p = q Then
       GoTo lagi;

    n := p * q;

PrivN.Text := IntToStr(n);
```

```
PubN.Text := IntToStr(n);
   pi := (p - 1) * (q - 1);

//search for e
c := pi - 1;
SetLength(temp2,c);

For e := 2 To (pi - 1) do
  begin
   temp6 := gcd(pi, e);
   If temp6 = 1 Then
      begin
       temp2[c] := e;
       c := c - 1;
      end;
  end;

//random selection of e
ulang:
   Randomize;
   temp3 := Round((pi - 1) * (Random(10000)/10000));
   temp4 := temp2[temp3];
   If (temp4 = 0) Or (temp4 = Null) Then
      GoTo ulang;


//select e that is prime
For i1 := 2 To Round(Sqrt(temp4)) do
 begin
   temp5 := temp4 Mod i1;
   If temp5 = 0 Then
     GoTo ulang;
 end;

PubE.Text := IntToStr(temp4);

//determine d such that d*e congruent 1 mod pi and d > 0, d > e
d := Euclid(pi, temp4);
If d < temp4 Then
  GoTo ulang;

PrivD.Text := IntToStr(d);
end;

procedure TMain.btEncryptClick(Sender: TObject);
```

```
var slama:String;
    Source,Dest:File;
    valid:TextFile;
    Buffer: array[0..7] of byte;
    temp:RC2MsgBlock;
    Read:Integer;
    awal,akhir,lama:TDateTime;
    i,d,n:Integer;
    s,x:longint;
begin
if edPass.Text = '' then
   MessageDlg('Kata Kunci Belum Diisi',mtError,mbOKCancel,0)
  else
    if RPlain.Text = '' then
       MessageDlg('Document masih kosong',mtError,mbOKCancel,0)
      else
       if PrivD.Text = '' then
          MessageDlg('Klik dahulu Generate Key',mtError,mbOKCancel,0)
         else
           begin
           if SaveDlgEnc.Execute then
             begin
               awal:=Time;
               edStart.Text:=TimeToStr(awal);
               AssignFile(Source,edPath.Text);
               AssignFile(Dest,SaveDlgEnc.FileName);
               Reset(source,1);
               ReWrite(Dest,1);
               repeat
                 for i:=0 to 7 do
                   Buffer[i]:=0;
                 temp[0]:=0;temp[1]:=0;
                 BlockRead(Source,Buffer,sizeof(Buffer),read);
                 if read <> 0 then
                   begin
                    move(Buffer,temp,read);
                    temp:=Encrypt(temp,edPass.Text);
                    move(temp,Buffer,sizeof(temp));
                    BlockWrite(Dest,Buffer,SizeOf(Buffer));
                   end;
               until read <> sizeof(Buffer);
               CloseFile(Source);
               CloseFile(Dest);
               akhir:=Time;
               edEnd.Text:=TimeToStr(akhir);
```

```
            lama:=(akhir-awal)*86000;
            str(lama:12:8,slama);
            edDuration.Text:=slama;
          end;
        if Application.MessageBox('  Masukan Nama File Validasi dan Digital
Signature'  ,'  Pesan'  ,MB_OK) = IDOK then
          begin
            if SaveDlgV.Execute then
              begin
                AssignFile(Source,SaveDlgEnc.FileName);
                Reset(Source,1);
                repeat
                  BlockRead(Source,Buffer,sizeof(Buffer),read);
                  for i:=0 to 7 do
                    s := s xor buffer[i];
                until read <> sizeof(buffer);
                CloseFile(Source);
                AssignFile(valid,SaveDlgV.FileName);
                Rewrite(valid);
                d:=StrToInt(PrivD.Text);
                n:=StrToInt(PrivN.Text);
                x:=pangkatmod(s,d,n);
                write(valid,inttostr(x));
                CloseFile(valid);
              end;
          end;
        end;
if OpenDlgDec.Execute then
  begin
    edPath1.Text:=OpenDlgDec.FileName;
    RichEdit1.Clear;
    RichEdit1.Lines.LoadFromFile(OpenDlgDec.FileName);
  end;
end;

procedure TMain.btBrowseClick(Sender: TObject);
begin
if OpenDlgEnc.Execute then
  begin
    edPath.Text:=OpenDlgEnc.FileName;
    RPlain.Clear;
    RPlain.Lines.LoadFromFile(OpenDlgEnc.FileName);
  end;
end;
```

```
procedure TMain.btBrowse1Click(Sender: TObject);
begin
if OpenDlgDec.Execute then
  begin
    RCipher.Clear;
    RCipher.Lines.LoadFromFile(OpenDlgDec.FileName);
    edPath1.Text:=OpenDlgDec.FileName;
    RCipher.Clear;
    RCipher.Lines.LoadFromFile(OpenDlgDec.FileName);
  end;
end;

procedure TMain.btDecryptClick(Sender: TObject);
var slama,temp1:String;
   Source,Dest:File;
   valid:TextFile;
   Buffer: array[0..7] of byte;
   temp:RC2MsgBlock;
   Read:Integer;
   awal,akhir,lama:TDateTime;
   i,e,n:Integer;
   s,p:Longint;
begin
if edPass1.Text = ' '  then
  MessageDlg('  Kata Kunci Belum Diisi'  ,mtError,mbOKCancel,0)
 else
   if RCipher.Text = ' '  then
     MessageDlg('  Document masih kosong'  ,mtError,mbOKCancel,0)
    else
      if (PubE1.Text = ' '  ) or (PubN1.Text = ' '  ) then
         MessageDlg('  Public Key Belum Di Isi'  ,mtError,mbOKCancel,0)
        else
         begin
           MessageDlg('  Buka File Validasi'  ,mtInformation,mbOKCancel,0);
           if OpenDlgV.Execute then
             begin
               e:=StrToInt(PubE1.Text);
               n:=StrToInt(PubN1.Text);
               AssignFile(valid,OpenDlgV.FileName);
               Reset(valid);
               Readln(valid,temp1);
               CloseFile(valid);
               AssignFile(Source,edPath1.Text);
               Reset(Source,1);
               repeat
```

```
                BlockRead(Source,Buffer,sizeof(Buffer),read);
                for i:=0 to 7 do
                   s := s xor buffer[i];
              until read <> sizeof(buffer);
              CloseFile(Source);
              p:=pangkatmod(StrToInt(temp1),e,n);
              if p <> s then
                begin
                 MessageDlg('  File Chipertext tidak valid proses
berhenti'  ,mtError,mbOKCancel,0);
                  exit;
                end
              else
              ShowMessage('  Data Sudah Valid'  );
               if SaveDlgDec.Execute then
                 begin
                  awal:=Time;
                  edStart1.Text:=TimeToStr(awal);
                  AssignFile(Source,edPath1.Text);
                  AssignFile(Dest,SaveDlgDec.FileName);
                  Reset(Source,1);
                  Rewrite(Dest,1);
                  repeat
                   for i:=0 to 7 do
                     Buffer[i]:=0;
                   temp[0]:=0;temp[1]:=0;
                   BlockRead(Source,Buffer,sizeof(Buffer),read);
                   if read <> 0 then
                     begin
                      move(Buffer,temp,read);
                      temp:=Decrypt(temp,edPass.Text);
                      move(temp,Buffer,sizeof(temp));
                      BlockWrite(Dest,Buffer,read);
                     end;
                  until read <> sizeof(Buffer);
                  CloseFile(Source);
                  CloseFile(Dest);
                  akhir:=Time;
                  edEnd1.Text:=TimeToStr(akhir);
                  lama:=(akhir-awal)*86000;
                  str(lama:12:8,slama);
                  edDuration1.Text:=slama;
                 end;
            end;
     end;
```

```
if OpenDlgEnc.Execute then
  begin
    edPath.Text:=OpenDlgEnc.FileName;
    RichEdit2.Clear;
    RichEdit2.Lines.LoadFromFile(OpenDlgEnc.FileName);
  end;
end;

procedure TMain.About1Click(Sender: TObject);
begin
frmAbout.ShowModal;
end;

procedure TMain.Button1Click(Sender: TObject);
begin
if ESave.Execute then
  begin
    RPlain.Lines.SaveToFile(ESave.FileName);
    ShowMessage('  Plain Text telah disimpan dengan
nama'  +chr(10)+chr(13)+ESave.FileName);
  end;
end;

procedure TMain.Button2Click(Sender: TObject);
begin
if ESave.Execute then
  begin
    RCipher.Lines.SaveToFile(ESave.FileName);
    ShowMessage('  Plain Text telah disimpan dengan
nama'  +chr(10)+chr(13)+ESave.FileName);
  end;
end;

end.
```

Listing RC2 enkripsi, dan deskripsi

```
unit rc2;

interface

uses
  Classes, Sysutils;

type
    RC2MsgBlock = array[0..3] of word;
    KeyData = array[0..63] of word;
```

```
   function Encrypt(Indata : RC2MsgBlock; Key : String): Rc2MsgBlock;
   function Decrypt(Indata : RC2MsgBlock; Key : String): Rc2MsgBlock;

implementation
{$R-}{$Q-}

{$I RC2.inc}

function LRot16(a, n: word): word;
begin
  Result:= (a shl n) or (a shr (16-n));
end;

function RRot16(a, n: word): word;
begin
  Result:= (a shr n) or (a shl (16-n));
end;
```

Listing enkripsi

```
function Encrypt(Indata : RC2MsgBlock; Key : String): Rc2MsgBlock;
var
  i, j: longword;
  w: array[0..3] of word;
  KeyB: array[0..127] of byte;
  KeyData: array[0..63] of word;
  size: longword;
begin
  //Init key
  size := length(key)*8;
  FillChar(KeyData,Sizeof(KeyData),0);
  FillChar(KeyB,Sizeof(KeyB),0);
  for i:=1 to length(key) do
      KeyB[i-1]:= ord(key[i]);
  for i:= (size div 8) to 127 do
    KeyB[i]:= sBox[(KeyB[i-(size div 8)]+KeyB[i-1]) and $FF];
  KeyB[0]:= sBox[KeyB[0]];
  Move(KeyB,KeyData,Sizeof(KeyData));

  //Encryption
  Move(InData,w,Sizeof(w));
  for i:= 0 to 15 do
  begin
    j:= i*4;

    w[0]:=LRot16((w[0]+(w[1]and (not w[3]))+(w[2] and w[3])+KeyData[j+0]),1);
```

```
    w[1]:= LRot16((w[1]+(w[2] and (not w[0]))+(w[3] and [0])+KeyData[j+1]),2);
    w[2]:= LRot16((w[2]+(w[3] and (not w[1]))+(w[0] and [1])+KeyData[j+2]),3);
    w[3]:= LRot16((w[3]+(w[0] and (not w[2]))+(w[1] and [2])+KeyData[j+3]),5);
    if (i= 4) or (i= 10) then
    begin
      w[0]:= w[0]+KeyData[w[3] and 63];
      w[1]:= w[1]+KeyData[w[0] and 63];
      w[2]:= w[2]+KeyData[w[1] and 63];
      w[3]:= w[3]+KeyData[w[2] and 63];
    end;
  end;
  move(w,result,sizeof(w));
end;
```

Listing Deskripsi

```
function Decrypt(Indata : RC2MsgBlock; Key : String): Rc2MsgBlock;
var
  i, j: longword;
  w: array[0..3] of word;
  KeyB: array[0..127] of byte;
  KeyData: array[0..63] of word;
  size: longword;
begin
  //Init key
  size := length(key)*8;
  FillChar(KeyData,Sizeof(KeyData),0);
  FillChar(KeyB,Sizeof(KeyB),0);
  for i:=1 to length(key) do
      KeyB[i-1]:= ord(key[i]);
  for i:= (size div 8) to 127 do
    KeyB[i]:= sBox[(KeyB[i-(size div 8)]+KeyB[i-1]) and $FF];
  KeyB[0]:= sBox[KeyB[0]];
  Move(KeyB,KeyData,Sizeof(KeyData));

  //Decryption
  Move(InData,w,Sizeof(w));
  for i:= 15 downto 0 do
  begin
    j:= i*4;
    w[3]:= RRot16(w[3],5)-(w[0] and (not w[2]))-(w[1] and w[2])-KeyData[j+3];
    w[2]:= RRot16(w[2],3)-(w[3] and (not w[1]))-(w[0] and w[1])-KeyData[j+2];
    w[1]:= RRot16(w[1],2)-(w[2] and (not w[0]))-(w[3] and w[0])-KeyData[j+1];
    w[0]:= RRot16(w[0],1)-(w[1] and (not w[3]))-(w[2] and w[3])-KeyData[j+0];
    if (i= 5) or (i= 11) then
    begin
```

```
    w[3]:= w[3]-KeyData[w[2] and 63];
    w[2]:= w[2]-KeyData[w[1] and 63];
    w[1]:= w[1]-KeyData[w[0] and 63];
    w[0]:= w[0]-KeyData[w[3] and 63];
   end;
  end;
  move(w,result,sizeof(w));
end;

end.
```

Listing Alogaritma RSA

```
unit RSATools;

interface
uses SysUtils, StrUtils, IdGlobal, Math;

  Function Euclid(nilai1 : Longint; nilai2 : Longint) : Longint ;
  Function gcd(p : longint; q : longint) : Longint;
  Function RdmPrime() : Longint;
  Function pangkatmod(num1 : longint;num2 : longint;num3 : longint) : longint;

implementation

Function Euclid(nilai1 : Longint; nilai2 : Longint) : Longint ;
var mex,bex,A1,A2,A3,Qex,T1,T2,T3,B1,B2,B3,hasil:Longint;
label itung, selesai;

begin
  mex := nilai1;
  bex := nilai2;

  A1 := 1;     B1 := 0;
  A2 := 0;     B2 := 1;
  A3 := mex;   B3 := bex;

  itung:
  if B3 = 0 then
    begin
      hasil := 0;
      GoTo selesai;
    end;
  if B3 = 1 then
    begin
       hasil := B2;
       GoTo selesai;
```

```
    end;

  Qex := A3 div B3;
  T1 := A1 - Qex * B1;
  T2 := A2 - Qex * B2;
  T3 := A3 - Qex * B3;

  A1 := B1;  B1 := T1;
  A2 := B2;  B2 := T2;
  A3 := B3;  B3 := T3;

  GoTo itung;
  selesai:
  Euclid := hasil;
end;

Function gcd(p : longint; q : longint) : Longint;
var A11,B11,R11 : Longint;
label balik;

begin
  A11 := p;
  B11 := q;

balik:
  If B11 = 0 Then
     gcd := A11
    else
     begin
      R11 := A11 Mod B11;
      A11 := B11;
      B11 := R11;
      GoTo balik;
     end;
end;

Function RdmPrime() : Longint;
var iRandom : integer; // holds random long result
    i2 : longint; // checkprime loop counter
    temp2a : Longint; //swap var
    iLowerBound : Longint;
    iUpperBound : Longint;
label i110;

begin
```

```
iLowerBound := 30;
iUpperBound := 300;
Randomize;
i110:
 iRandom := (Round((iUpperBound - iLowerBound + 1) *
(Random(10000)/10000)) + iLowerBound);

//trap handler
If (iRandom = 0) Or (iRandom = 1) Then
GoTo i110;

//check number
for i2:= 2 to (iRandom-1) do
  begin
   temp2a := iRandom Mod i2;
   If temp2a = 0 Then
     GoTo i110;
  end;

   RdmPrime := iRandom;
End;


Function pangkatmod(num1 : longint;num2 : longint;num3 : longint) : longint;
var a22,b22,n22,nilaimod,nilaic,nilaid,nilaii,nilaik,naik : longint;
    barray : array of Variant;
    decbin : Variant;

begin
a22 := num1;
b22 := num2;
n22 := num3;

decbin := inttobin(b22);
nilaic := 0;
nilaid := 1;
nilaik := Length(decbin);
SetLength(barray,nilaik);
naik := 1;

For nilaii := nilaik - 1 downto 0 do
  begin
    barray[nilaii] := Midstr(decbin, naik, 1);
    naik := naik + 1;
    nilaic := 2 * nilaic;
```

```
    nilaid := (nilaid * nilaid) Mod n22;
    If barray[nilaii] = 1 Then
      begin
        nilaic := nilaic + 1;
        nilaid := (nilaid * a22) Mod n22;
      end;
  end;

pangkatmod := nilaid;
End;

end.
```

Listing Tools

```
unit Tools;

interface
uses
  Sysutils;

type
{$IFDEF VER120}
  dword= longword;
{$ELSE}
  dword= longint;
{$ENDIF}

function LRot16(X: word; c: integer): word; assembler;
function RRot16(X: word; c: integer): word; assembler;
function LRot32(X: dword; c: integer): dword; assembler;
function RRot32(X: dword; c: integer): dword; assembler;
procedure XorBlock(I1, I2, O1: PByteArray; Len: integer);
procedure IncBlock(P: PByteArray; Len: integer);
function StrToDWord(s : string) : DWord;
function DWordToStr(wo : DWord) : string;

implementation

function LRot16(X: word; c: integer): word; assembler;
asm
  mov ecx,&c
  mov ax,&X
  rol ax,cl
  mov &Result,ax
```

```
end;

function RRot16(X: word; c: integer): word; assembler;
asm
  mov ecx,&c
  mov ax,&X
  ror ax,cl
  mov &Result,ax
end;

function LRot32(X: dword; c: integer): dword; register; assembler;
asm
  mov ecx, edx
  rol eax, cl
end;

function RRot32(X: dword; c: integer): dword; register; assembler;
asm
  mov ecx, edx
  ror eax, cl
end;

procedure XorBlock(I1, I2, O1: PByteArray; Len: integer);
var
  i: integer;
begin
  for i:= 0 to Len-1 do
    O1[i]:= I1[i] xor I2[i];
end;

procedure IncBlock(P: PByteArray; Len: integer);
begin
  Inc(P[Len-1]);
  if (P[Len-1]= 0) and (Len> 1) then
    IncBlock(P,Len-1);
end;

function StrToDWord(s : string) : DWord;
var cely : Dword;
    Si1,Si2,Si3,Si4 : byte;
begin
Si1:=ord(s[1]);
Si2:=ord(s[2]);
Si3:=ord(s[3]);
Si4:=ord(s[4]);
```

```
cely:=Si1;
cely:=cely shl 8;
cely:=cely + Si2;
cely:=cely shl 8;
cely:=cely + Si3;
cely:=cely shl 8;
cely:=cely + Si4;
StrToDWord:=cely;
end;

function DWordToStr(wo : DWord) : string;
var vysl : string;
    rr,ll : word;
    Si1,Si2,Si3,Si4 : byte;
begin
rr:=(wo shl 16) shr 16;
ll:=(wo shr 16);
Si2:=(ll shl 8) shr 8;
Si1:=(ll shr 8);
Si4:=(rr shl 8) shr 8;
Si3:=(rr shr 8);
vysl:=chr(Si1)+chr(Si2)+chr(Si3)+chr(Si4);
DWordToStr:=vysl;
end;

end.
```

Listing About

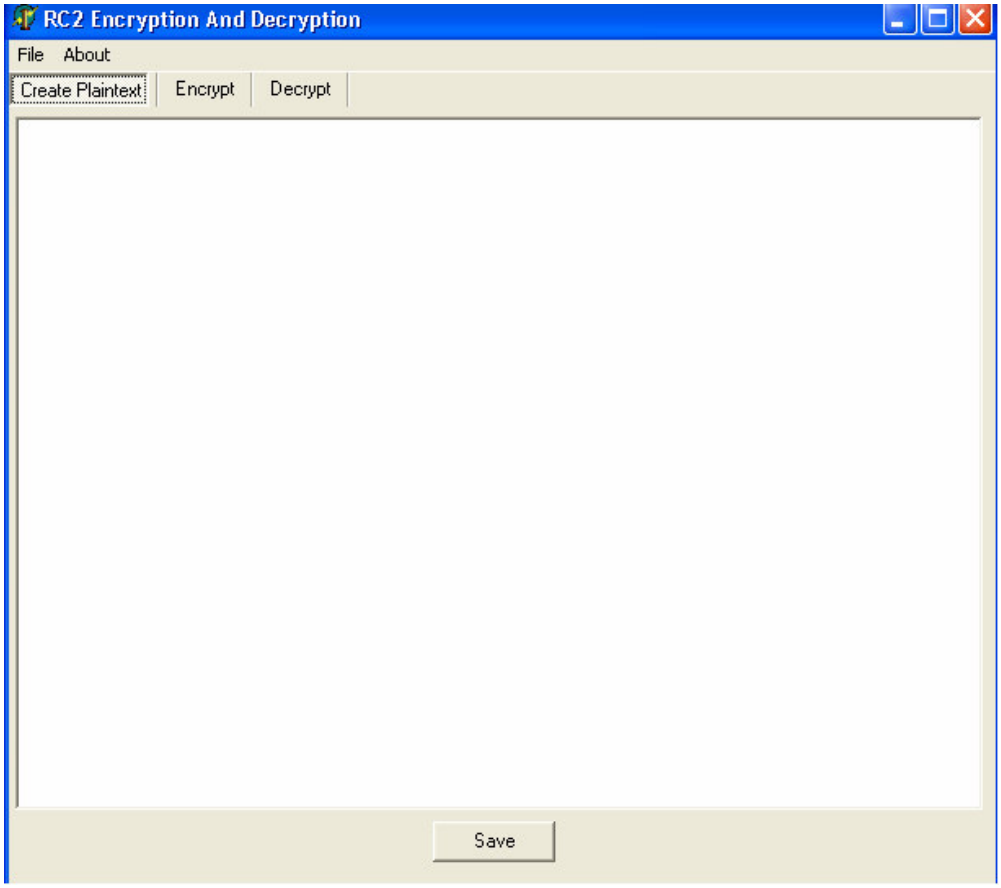
```
unit UAbout;

interface

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

type
  TfrmAbout = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label5: TLabel;
    Label2: TLabel;
    Label3: TLabel;
```

```
    Label4: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmAbout: TfrmAbout;

implementation

{$R *.dfm}

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

end.
```

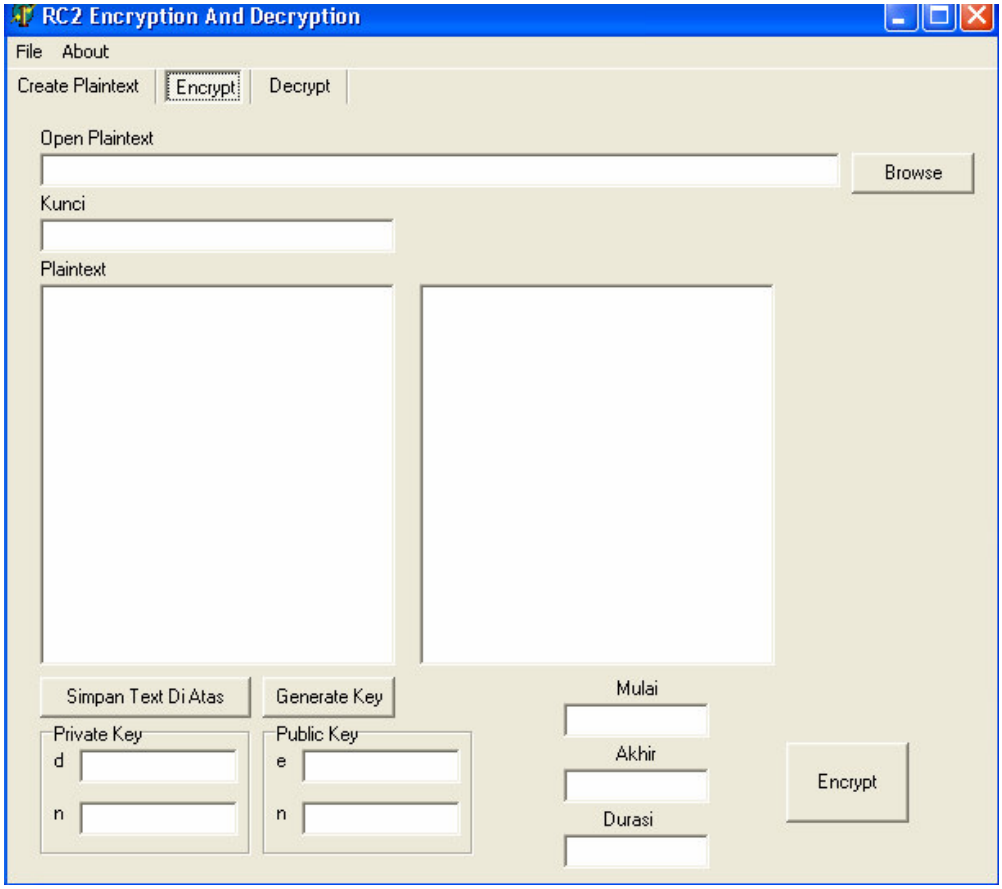
LAMPIRAN B

TAMPILAN PROGRAM PENGAMAN DATA

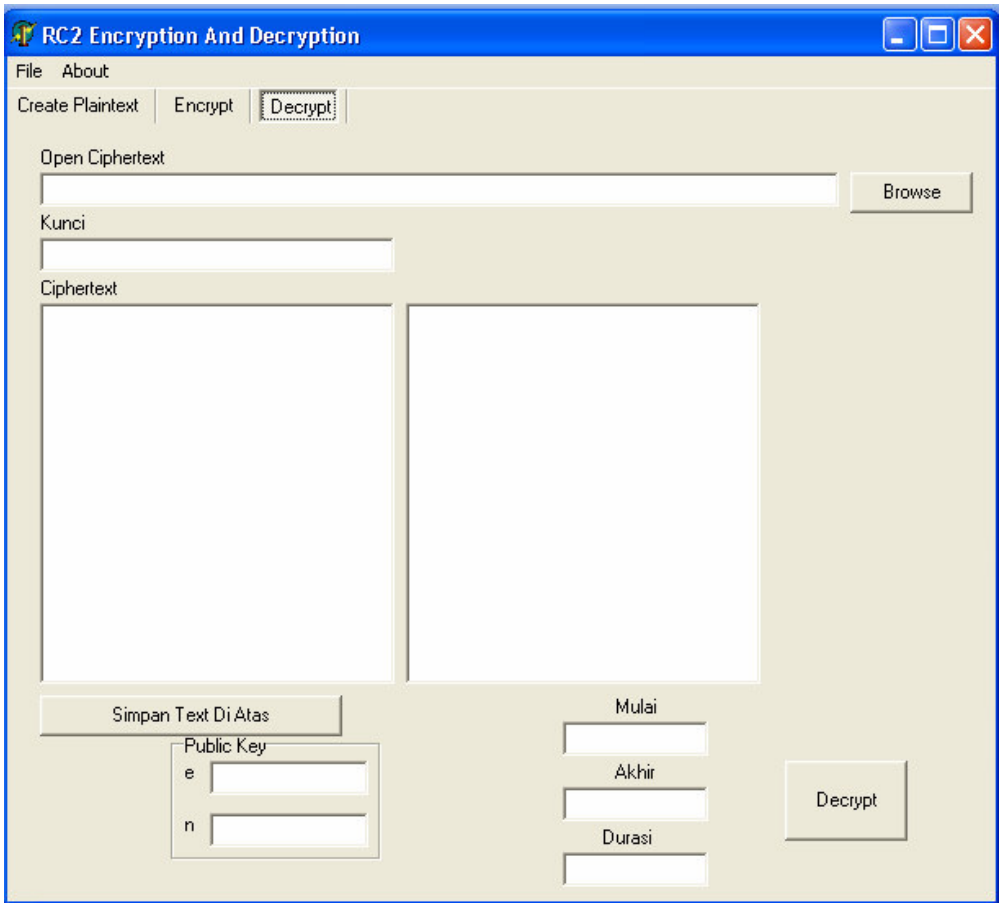
Tampilan Menu About



Tampilan menu Create plaintext



Tampilan Menu Program Enkripsi



Tampilan Menu Dekripsi