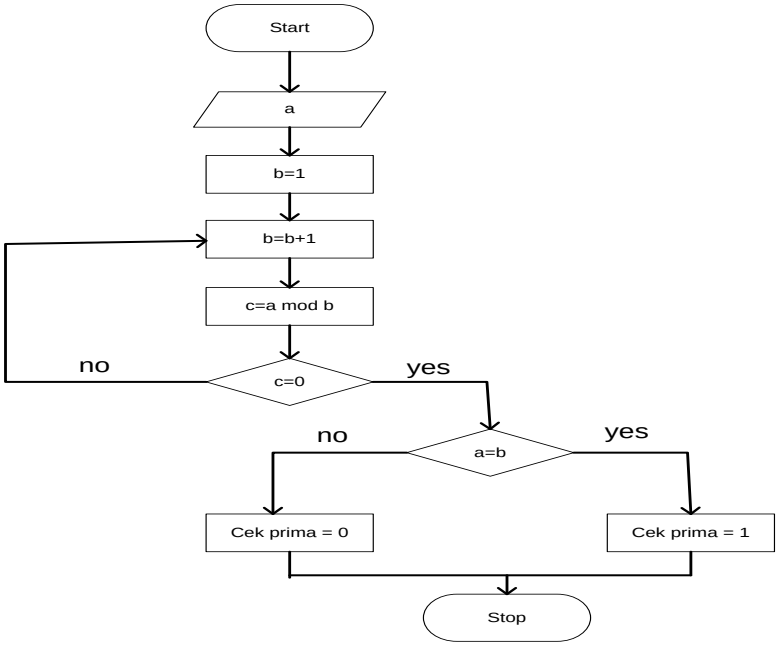
LAMPIRAN A: ALGORITMA DAN FLOWCHART PROSEDUR ELGAMALCRYPT

1. Algoritma dan *Flowchart* cek_prima.

1.1 Algoritma cek_prima

1. Mulai.
2. Masukkan nilai bilangan (a).
3. b = 1.
4. b = b + 1.
5. c = a mod b.
6. Periksa apakah c = 0.
7. Jika tidak, kembali ke langkah (4).
8. Jika iya, periksa apakah a = b.
9. Jika tidak, maka a bukan bilangan prima.
10. Jika iya, a adalah bilangan prima.
11. Berhenti.

1.2 Flowchart cek_prima

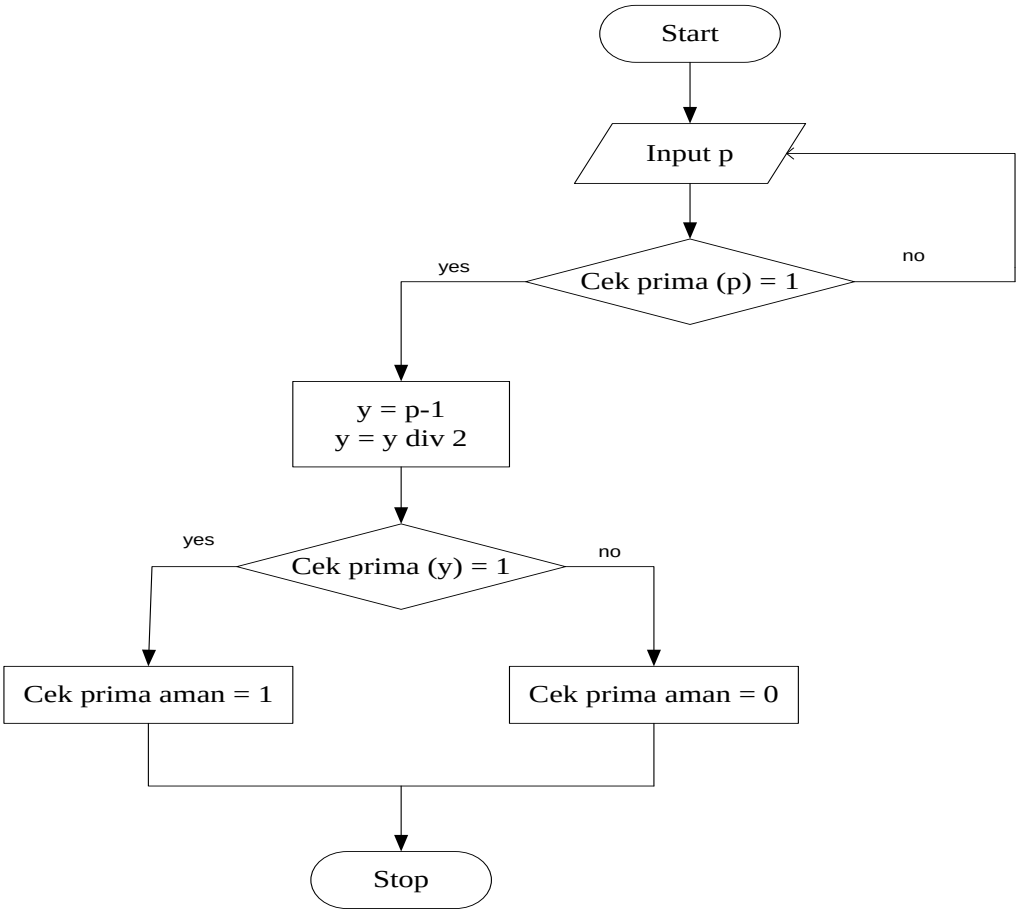


2. Algoritma dan *Flowchart* cek_prima_aman.

2.1 Algoritma cek_prima_aman

1. Mulai.
2. Masukkan bilangan prima (p).
3. Periksa apakah p merupakan bilangan prima.
4. Jika tidak kembali ke langkah (2).
5. Jika iya, lanjutkan ke langkah (6).
6. y = p – 1.
7. y = y div 2.
8. Periksa apakah y adalah bilangan prima.
9. Jika iya, maka p adalah bilangan prima aman.
10. Jika tidak, maka p bukanlah bilangan prima aman.

2.2 Flowchart cek_prima_aman

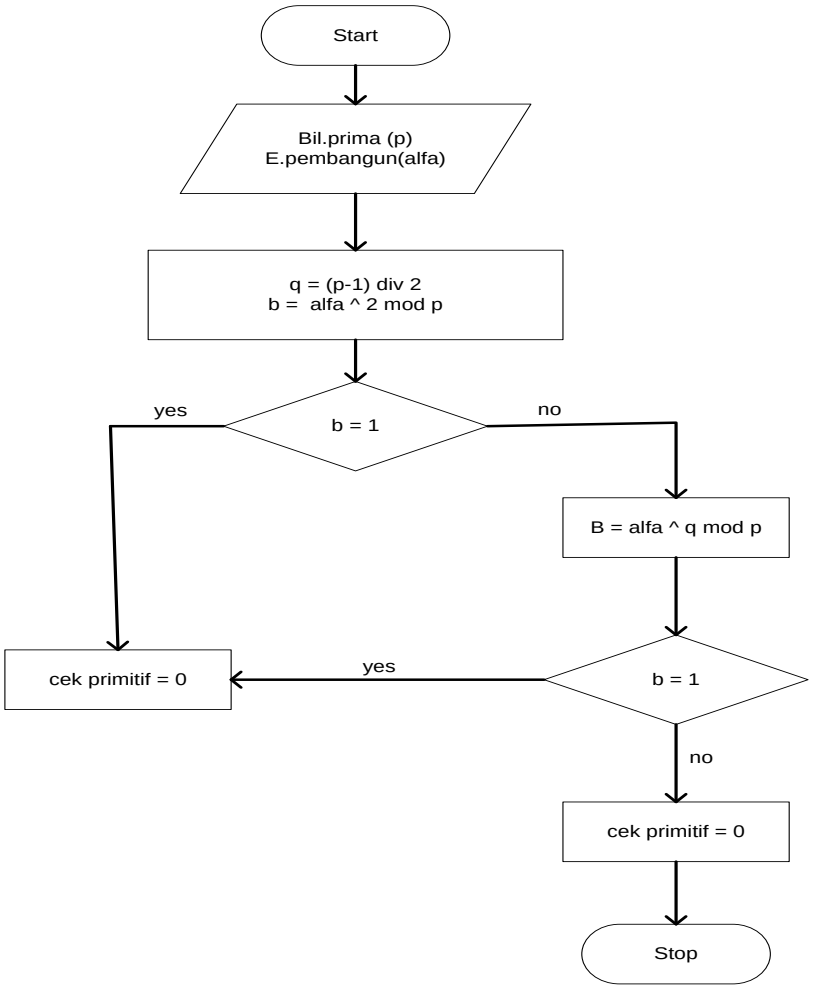


3. Algoritma dan *Flowchart* cek_elemen_primitif

3.1 Algoritma cek_ elemen_primitif

1. Mulai.
2. Masukkan bilangan prima (p) dan elemen pembangun (alfa).
3. Hitung q = (p-1) div 2.
4. b = alfa ^ 2 mod p.
5. Periksa apakah b = 1.
6. Jika iya, alfa bukanlah elemen primitif.
7. Jika tidak, b = (alfa ^ q) (mod p).

8. Periksa apakah b = 1,
9. Jika iya, maka alfa adalah elemen primitif.
10. Jika tidak, alfa bukanlah elemen primitif.
11. Berhenti.



LAMPIRAN B: LISTING PROGRAM

1. Modul Utama

```
program ProjectElgamal;

uses
  Forms,
  UUtama in 'UUtama.pas' {FUtama},
  UPilKunci in 'UPilKunci.pas' {FKOtomatis},
  UKunciManual in 'UKunciManual.pas' {FKManual},
  UEnkripsi in 'UEnkripsi.pas' {FEnkripsi},
  UDekripsi in 'UDekripsi.pas' {FDekripsi},
  UkEnkripsi in 'UkEnkripsi.pas' {FKEnkripsi},
  UKDekripsi in 'UKDekripsi.pas' {FKDekripsi},
  UAbout in 'UAbout.pas' {FAbout},
  Upilihan in 'uPilihan.pas' {FPilKunci},
  uGlobal in 'uGlobal.pas',
  uHelp in 'uHelp.pas' {FHelp};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TFUtama, FUtama);
  Application.CreateForm(TFHelp, FHelp);
  Application.Run;
end.
```

2. Form Utama

```
unit UUtama;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, Menus, StdCtrls, Buttons, ExtCtrls, ActnMan,
ActnColorMaps,
  XPMan;

type
  TFUtama = class(TForm)
    lJudul1: TLabel;
    Label2: TLabel;
    BitBtn2: TBitBtn;
    Panel1: TPanel;
    bbPkunci: TBitBtn;
    bbEnkripsi: TBitBtn;
    bbDekripsi: TBitBtn;
```

```
    bbAbout: TBitBtn;
    StaticText1: TStaticText;
    XPManifest1: TXPManifest;
    Label1: TLabel;
    procedure BitBtn2Click(Sender: TObject);
    procedure bbPkunciClick(Sender: TObject);
    procedure bbEnkripsiClick(Sender: TObject);
    procedure bbDekripsiClick(Sender: TObject);
    procedure bbAboutClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FUtama: TFUtama;

implementation
uses uPilKunci,uEnkripsi,uDekripsi,uAbout, Upilihan;
{$R *.dfm}

procedure TFUtama.BitBtn2Click(Sender: TObject);
begin
  Close;
end;

procedure TFUtama.bbPkunciClick(Sender: TObject);
begin
  Application.CreateForm(TFPilKunci, FPilKunci);
  FPilKunci.ShowModal;
end;

procedure TFUtama.bbEnkripsiClick(Sender: TObject);
begin
  Application.CreateForm(TFEnkripsi, FEnkripsi);
  FEnkripsi.ShowModal;
end;

procedure TFUtama.bbDekripsiClick(Sender: TObject);
begin
    Application.CreateForm(TFDekripsi, FDekripsi);
    FDekripsi.ShowModal;
end;

procedure TFUtama.bbAboutClick(Sender: TObject);
begin
    Application.CreateForm(TFAbout, FAbout);
    FAbout.ShowModal;
```

```
end;

end.
```

3. Unit Global

```
unit uGlobal;

interface
  function cekprimitif(alfa:longint; p:longint):integer;
  function fastexp(r:longint; s:longint; t:longint):longint;
  function mrtest(p:longint; t:longint):integer;
  function primaaman():longint;
  function cekprimaaman(p:longint):integer;
  function cekprima(a:longint):integer;
  function gcd(a:longint; b:longint):longint;
  function primitif(p:longint):longint;

implementation

{Fungsi Untuk Mencari Elemen Primitif acak}
function primitif(p:longint):longint;
var
   alfa:longint;
begin
     repeat
           randomize;
           alfa:=random(p-2)+1;
     until
           cekprimitif(alfa,p)=1;
     primitif:=alfa;
end;

{Fungsi Untuk Menghitung gcd(a,b)}
function gcd(a:longint; b:longint):longint;
var
   r:longint;
begin
     if a<0 then a:=-a;
     if b<0 then b:=-b;
       while b <> 0 do
          begin
               r:=a mod b;
                   a:=b;
               b:=r;
            end;
```

```
     gcd:=a;
end;

{Fungsi Pengecekan Bilangan Prima}
function cekprima(a:longint):integer;
var
   b,c:longint;
begin
     b:=1;
     repeat
           b:=b+1;
           c:=a mod b;
     until c=0;
     if a=b then cekprima:=1
     else cekprima:=0;
end;

{Fungsi Pengecekan Bilangan Prima Aman}
function cekprimaaman(p:longint):integer;
var
   y:longint;
begin
     if cekprima(p) = 1 then
     begin
      y:=p-1;
      y:=y div 2;
      if cekprima(y) = 1 then Result  := 1;
      if cekprima(y) = 0 then Result  := 0;
     end
     else Result := 0;
end;

{Fungsi Mencari Bilangan Prima Aman}
function primaaman():longint;
var
  p    : Integer;
  rand :longint;
begin
     randomize;
     repeat
          repeat
                rand:=random(2000)+128+1;
                if rand mod 2 = 0 then rand:=rand+1;
          until
               cekprima(rand)=1;
     p:=(2*rand)+1;
     until
          cekprima(p)=1;
     primaaman:=p;
```

```
end;

{Fungsi Tes Keprimaan Miller-Rabbin}
function mrtest(p:longint; t:longint):integer;
var
   x,y,s,m,a,r,j,i:longint;
begin
    mrtest:=1;
      randomize;
      x:=p;
      m:=0;
      repeat
            x:=x div 2;
            m:=m+1;
      until
      (x mod 2) <> 0;
      s:=m;
      r:=x;
      for i:=1 to t do
    begin
            a:=random(p-3)+2;
            y:=fastexp(a,r,p) mod p;
    j:=0;
            if (y<>1) and (y<>(p-1)) then
                  begin
                  j:=1;
                  while (j<=(s-1)) and (y<>(p-1)) do
                        begin
                              y:=(y*y) mod p;
                              if (y=1) then mrtest:=0;
                    j:=j+1;
                        end;
            if y <> (p-1) then mrtest:=0;
                  end;
    end;
end;

{Metode Fast Exponentiation}
function fastexp(r:longint; s:longint; t:longint):longint;
var
      x,mtemp,atemp:longint;
begin
      atemp:=r;
      mtemp:=s;
      x:=1;
      while mtemp <> 0 do
                  begin
            while mtemp mod 2 = 0 do
                begin
```

```
                  mtemp:=mtemp div 2;
                  atemp:=(atemp*atemp) mod t;
               end;
            mtemp:=mtemp-1;
            x:=(x*atemp) mod t;
            end;
    if x<0 then x:=(x+t) mod t;
      fastexp:=x;
end;

function cekprimitif(alfa:longint; p:longint):integer;
var
   b,q:longint;
begin
     q:=(p-1) div 2;
     b:=fastexp(alfa,2,p);
     if b=1 then Result := 0
     else
         b:=fastexp(alfa,q,p);
         if b=1 then Result := 0
         else
             Result := 1;
end;

end.
```

4. Form Pilihan Pembentukan Kunci

```
unit Upilihan;

interface

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

type
  TFPilKunci = class(TForm)
    bbOtomatis: TBitBtn;
    bbManual: TBitBtn;
    bbCancel: TBitBtn;
    Bevel1: TBevel;
    procedure bbOtomatisClick(Sender: TObject);
    procedure bbManualClick(Sender: TObject);
    procedure bbCancelClick(Sender: TObject);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
    Tutup     : Boolean;
  end;

var
  FPilKunci : TFPilKunci;

implementation
uses uKunciManual,uPilKunci;
{$R *.dfm}

procedure TFPilKunci.bbOtomatisClick(Sender: TObject);
begin
   Tutup  := False;
   Application.CreateForm(TFKOtomatis, FKOtomatis);
   FKOtomatis.ShowModal;
   if Tutup = True then
    begin
      ModalResult := mrCancel;
      FPilKunci.Release;
    end;
end;

procedure TFPilKunci.bbManualClick(Sender: TObject);
begin
  Tutup := False;
  Application.CreateForm(TFKManual, FKManual);
  FKManual.ShowModal;
  if Tutup = True then
    begin
      ModalResult := mrCancel;
      FPilKunci.Release;
    end;
end;

procedure TFPilKunci.bbCancelClick(Sender: TObject);
begin
  ModalResult := mrCancel;
  FPilKunci.Release;
end;

end.
```

5. Form Pembentukan Kunci Otomatis

```
unit UPilKunci;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls, IniFiles {nanda};

type
  TFKOtomatis = class(TForm)
    lBPrima: TLabel;
    lEPrimitif: TLabel;
    Label1: TLabel;
    eBPrima: TEdit;
    ePrimitif: TEdit;
    eRahasia: TEdit;
    bbGenerate: TBitBtn;
    lKPublik: TLabel;
    eKPublik: TEdit;
    bbSave: TBitBtn;
    bbkembali: TBitBtn;
    SaveDialog1: TSaveDialog;
    Memo1: TMemo;
    Bevel1: TBevel;
    Bevel2: TBevel;
    Bevel3: TBevel;
    Bevel4: TBevel;
    SaveDialog2: TSaveDialog;
    Memo2: TMemo;
    Label2: TLabel;
    eBeta: TEdit;
    procedure bbkembaliClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure bbGenerateClick(Sender: TObject);
    procedure bbSaveClick(Sender: TObject);
    procedure eBPrimaKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }

  end;
var
  FKOtomatis: TFKOtomatis;
```

```
implementation

uses Upilihan, uGlobal, Math;

{$R *.dfm}

procedure TFKOtomatis.bbkembaliClick(Sender: TObject);
begin
    Close;
end;

procedure TFKOtomatis.FormCreate(Sender: TObject);
Var Ini:TIniFile;
begin
  ini             :=
TIniFile.Create(ExtractFilePath(Application.ExeName) + '\data.ini');
  eBPrima.Text    := ini.ReadString('KUNCI','Prima','');
  ePrimitif.Text  := ini.ReadString('KUNCI','Pembangun','');
  eBeta.Text      := ini.ReadString('KUNCI','Beta','');
  eRahasia.Text   := ini.ReadString('KUNCI','Rahasia','');
  eKPublik.Text   := ini.ReadString('KUNCI','Publik','');
  ini.Free;
end;

procedure TFKOtomatis.bbGenerateClick(Sender: TObject);
Var Beta  : Integer;
begin
  Randomize;
  eBPrima.Text    := IntToStr(primaaman);
  ePrimitif.Text  := IntToStr(primitif(StrToInt(eBPrima.Text)));
  eRahasia.Text   := IntToStr(RandomRange(1,StrToInt(eBPrima.Text)-
2));
  beta            :=
fastexp(StrToInt(ePrimitif.Text),StrToInt(eRahasia.Text),StrToInt(eBP
rima.Text));
  eBeta.Text      := IntToStr(beta);
  eKPublik.Text   := eBPrima.Text + ', ' + ePrimitif.Text + ', ' +
eBeta.Text;
end;

procedure TFKOtomatis.bbSaveClick(Sender: TObject);
var
  Ini   : TIniFile;
begin
  Ini := TIniFile.Create(ExtractFilePath(Application.ExeName) +
'\data.ini');
  ini.WriteString('KUNCI','Prima',Trim(eBPrima.Text));
  ini.WriteString('KUNCI','Pembangun',Trim(ePrimitif.Text));
  ini.WriteString('KUNCI','Rahasia',Trim(eRahasia.Text));
```

```
  ini.WriteString('KUNCI','Beta',Trim(eBeta.Text));
  ini.WriteString('KUNCI','Publik',Trim(eKPublik.Text));
  ini.Free;
  FPilKunci.Tutup := True;
  ModalResult     := mrCancel;
  FKOtomatis.Release;
end;
procedure TFKOtomatis.eBPrimaKeyPress(Sender: TObject; var Key:
Char);
begin
  if not (key in ['0'..'9',#8,#13]) then key := #0;
  if key = #13 then Perform(WM_NEXTDLGCTL,0,0);
end;

end.
```

6. Form Pembentukan Kunci Manual

```
unit UKunciManual;

interface

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

type
  TFKManual = class(TForm)
    bbBack: TBitBtn;
    bbProses: TBitBtn;
    Bevel5: TBevel;
    Bevel6: TBevel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Bevel7: TBevel;
    Bevel8: TBevel;
    Label6: TLabel;
    eBPrima: TEdit;
```

```
    ePrimitif: TEdit;
    eRahasia: TEdit;
    bbValidate: TBitBtn;
    eKPublik: TEdit;
    bbSave: TBitBtn;
    eBeta: TEdit;
    BitBtn1: TBitBtn;
    bbHelp: TBitBtn;

    procedure bbBackClick(Sender: TObject);

    procedure bbSaveClick(Sender: TObject);
    procedure bbValidateClick(Sender: TObject);
    procedure eBPrimaKeyPress(Sender: TObject; var Key: Char);
    procedure BitBtn1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure bbHelpClick(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var FKManual  : TFKManual;
    ini       : TIniFile;

implementation
uses uPilihan, UPilKunci, UUtama, uGlobal, uHelp;
{$R *.dfm}

procedure TFKManual.bbBackClick(Sender: TObject);
begin
  ModalResult := mrCancel;
  FKManual.Release;
end;

procedure TFKManual.bbSaveClick(Sender: TObject);
begin
  Ini := TIniFile.Create(ExtractFilePath(Application.ExeName) +
'\data.ini');
  ini.WriteString('KUNCI','Prima',Trim(eBPrima.Text));
  ini.WriteString('KUNCI','Pembangun',Trim(ePrimitif.Text));
  ini.WriteString('KUNCI','Rahasia',Trim(eRahasia.Text));
  ini.WriteString('KUNCI','Beta',Trim(eBeta.Text));
  ini.WriteString('KUNCI','Publik',Trim(eKPublik.Text));
  ini.Free;
  bbSave.Enabled  := False;
end;
```

```
procedure TFKManual.bbValidateClick(Sender: TObject);
begin
  { Validasi Bil. Prima }
  if Trim(eBPrima.Text) = '' then
    begin
      MessageDlg('Kolom Bilangan Prima Tidak Boleh Kosong
!',mtError,[mbok],0);
      eBPrima.SetFocus;
      Exit;
    end;

  if StrtoInt(eBPrima.Text) < 255 then
    begin
      MessageDlg('Nilai Bilangan  Harus Lebih Besar Dari 255
!',mtError,[mbok],0);
      eBPrima.SetFocus;
      Exit;
    end;

  if cekprimaaman(StrtoInt(eBPrima.Text)) = 0 then
    begin
      MessageDlg('Bukan Bilangan Prima Aman',MtError,[MbOK],0);
      eBPrima.SetFocus;
      eBPrima.Text := '';
      Exit;
    end;
  { End Of Validasi Bil. Prima }

  { Validasi Elemen Pembangun }
  if trim(ePrimitif.Text) = '' then
    begin
      MessageDlg('Maaf, Kolom Elemen Pembangun Harus
Diisi.',mtError,[mbok],0);
      ePrimitif.SetFocus;
      Exit;
    end;

  if StrToInt(ePrimitif.Text) > StrtoInt(eBPrima.Text) then
    begin
      MessageDlg('Maaf, Nilai Elemen Pembangun Harus Lebih Kecil Dari
Nilai Bilangan Prima',mtError,[mbok],0);
      ePrimitif.SetFocus;
      Exit;
    end;

  if cekprimitif(StrToInt(ePrimitif.Text),StrtoInt(eBPrima.Text)) = 0
then
    begin
```

```
      MessageDlg('Bukan Elemen Pembangun',MtError,[MbOK],0);
      ePrimitif.setfocus;
      ePrimitif.Text := '';
    end;
  { End Of Validasi Elemen Pembangun }

  { Validasi Bil. Rahasia }
  if trim(eRahasia.Text) = '' then
    begin
      MessageDlg('Maaf, Kolom Bilangan Rahasia Harus
Diisi.',mtError,[mbok],0);
      eRahasia.SetFocus;
      Exit;
    end;

   if (StrToInt(eRahasia.Text) < 0 ) Or (StrToInt(eRahasia.Text) >
(StrtoInt(eBPrima.Text) - 2)) then
      begin
           MessageDlg('Di Luar Interval',MtError,[MbOK],0);
           eRahasia.SetFocus;
           eRahasia.Text := '';
      end;
  { End Of Validasi Bil. Rahasia }

  eBeta.Text      := IntToStr( fastexp(
StrToInt(ePrimitif.Text),StrToInt(eRahasia.Text),StrtoInt(eBPrima.Tex
t) ) );
  eKPublik.Text   := eBPrima.Text + ', ' + ePrimitif.Text + ', ' +
eBeta.Text;
  bbSave.Enabled  := True;
end;
procedure TFKManual.eBPrimaKeyPress(Sender: TObject; var Key: Char);
begin
  if not (key in ['0'..'9',#8,#13]) then key := #0;
  if key = #13 then Perform(WM_NEXTDLGCTL,0,0);
end;

procedure TFKManual.BitBtn1Click(Sender: TObject);
begin
  eBPrima.Text    := '';
  ePrimitif.Text  := '';
  eRahasia.Text   := '';
  eBeta.Text      := '';
  eKPublik.Text   := '';
  bbSave.Enabled  := False;
  eBPrima.SetFocus;
end;

procedure TFKManual.FormCreate(Sender: TObject);
```

```
begin
  ini             :=
TIniFile.Create(ExtractFilePath(Application.ExeName) + '\data.ini');
  eBPrima.Text    := ini.ReadString('KUNCI','Prima','');
  ePrimitif.Text  := ini.ReadString('KUNCI','Pembangun','');
  eBeta.Text      := ini.ReadString('KUNCI','Beta','');
  eRahasia.Text   := ini.ReadString('KUNCI','Rahasia','');
  eKPublik.Text   := ini.ReadString('KUNCI','Publik','');
  ini.Free;
end;

procedure TFKManual.bbHelpClick(Sender: TObject);
begin
  FHelp.Show;
end;

end.
```

7. Form Enkripsi

```
unit UEnkripsi;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, Menus, StdCtrls, FileCtrl, Grids, Outline, DirOutln,
Buttons,
  ExtCtrls;

type
  TFEnkripsi = class(TForm)
    DirectoryOutline1: TDirectoryOutline;
    FileListBox1: TFileListBox;
    DriveComboBox1: TDriveComboBox;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    bbTulis: TBitBtn;
    bbBatal: TBitBtn;
    bbEnkrip: TBitBtn;
    Bevel1: TBevel;
    Bevel2: TBevel;
    Bevel3: TBevel;
    Bevel5: TBevel;
    Bevel6: TBevel;
    Bevel7: TBevel;
```

```
    Panel1: TPanel;
    MPesan: TMemo;
    procedure DriveComboBox1Change(Sender: TObject);
    procedure DirectoryOutline1Change(Sender: TObject);
    procedure FileListBox1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure bbTulisClick(Sender: TObject);
    procedure bbBatalClick(Sender: TObject);
    procedure bbEnkripClick(Sender: TObject);


  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FEnkripsi   : TFEnkripsi;
  Tulis_Baru  : Boolean;

implementation
uses uKEnkripsi;
{$R *.dfm}

procedure TFEnkripsi.DriveComboBox1Change(Sender: TObject);
begin
  DirectoryOutline1.Drive := DriveComboBox1.Drive;
end;

procedure TFEnkripsi.DirectoryOutline1Change(Sender: TObject);
begin
  FileListBox1.Directory := DirectoryOutline1.Directory;
  FileListBox1.Mask := '*.txt';
end;

procedure TFEnkripsi.FileListBox1Click(Sender: TObject);
begin
  MPesan.Lines.LoadFromFile(Filelistbox1.FileName);
end;

procedure TFEnkripsi.FormCreate(Sender: TObject);
begin
 MPesan.Lines.Clear;
 Tulis_Baru := False;
end;

procedure TFEnkripsi.bbTulisClick(Sender: TObject);
begin
```

```
  MPesan.Lines.Clear;
  MPesan.ReadOnly := False;
  Tulis_Baru      := True;
  MPesan.SetFocus;
end;

procedure TFEnkripsi.bbBatalClick(Sender: TObject);
begin
  Mpesan.Lines.Clear;
  ModalResult := mrCancel;
  FEnkripsi.Release;
end;

procedure TFEnkripsi.bbEnkripClick(Sender: TObject);
begin
  if (Filelistbox1.FileName = '') AND (Tulis_Baru = False) then
    begin
      MessageDlg('Maaf, Data Kosong. Proses Dibatalkan
!!',mtError,[mbok],0);
      Exit;
    end;

  if (Tulis_Baru = True) AND (MPesan.Lines.Text = '') then
    begin
      MessageDlg('Maaf, Data Kosong. Proses Dibatalkan
!!',mtError,[mbok],0);
      Exit;
    end;
  Application.CreateForm(TFKEnkripsi, FKEnkripsi);
  FKEnkripsi.ShowModal;
end;

end.
```

8. Form Konfirmasi Kunci Enkripsi

```
unit UkEnkripsi;

interface

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

type
  TFKEnkripsi = class(TForm)
    Label1: TLabel;
```

```
    Label2: TLabel;
    Label3: TLabel;
    eBprima: TEdit;
    eEprimitif: TEdit;
    bbProses: TBitBtn;
    bbBatal: TBitBtn;
    eKpublik: TEdit;
    Bevel1: TBevel;
    Memo1: TMemo;
    SaveDialog1: TSaveDialog;
    mTampung: TMemo;
    StaticText1: TStaticText;
    Bevel2: TBevel;
    procedure bbBatalClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure bbProsesClick(Sender: TObject);
    procedure eBprimaChange(Sender: TObject);
    procedure eEprimitifChange(Sender: TObject);
    procedure eKpublikChange(Sender: TObject);
    procedure eBprimaKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var FKEnkripsi: TFKEnkripsi;

implementation

uses UEnkripsi, UDekripsi, Math, uGlobal;

{$R *.dfm}

procedure TFKEnkripsi.bbBatalClick(Sender: TObject);
begin
  ModalResult := mrCancel;
  FKEnkripsi.Release;
end;

procedure TFKEnkripsi.FormCreate(Sender: TObject);
Var Ini : TIniFile;
begin
  Ini := TIniFile.Create(ExtractFilePath(Application.ExeName) +
'\data.ini');
  eBprima.Text    := ini.ReadString('KUNCI','Prima','');
  eEprimitif.Text := ini.ReadString('KUNCI','Pembangun','');
  eKpublik.Text   := ini.ReadString('KUNCI','Beta','');
  ini.Free;
```

```
end;

procedure TFKEnkripsi.bbProsesClick(Sender: TObject);
var
  i, c, p, k, Alfa, K_Publik    : Integer;
  G_amma,D_elta                 : integer;
  nama, pesan, gamma1, delta1   : string;
Label Lanjut;
begin
  pesan := '';
  mTampung.Clear;
  for i := 0 to FEnkripsi.MPesan.Lines.Count - 1 do
    begin
      pesan := Trim(FEnkripsi.MPesan.Lines.Strings[i]);
      for c:= 1 to length(pesan) do
        begin
          mTampung.Lines.Add(IntToStr(ord(pesan[c])));
        end;
      mTampung.Lines.Add(IntToStr(19));
    end;

  { Inisialisasi Variable }
  memo1.Lines.Clear;
  p         := StrToInt(eBprima.Text);
  alfa      := StrToInt(eEPrimitif.Text);
  K_Publik  := StrToInt(eKpublik.Text);
  { End of Inisialisasi Variable }

  for i := 0 to mTampung.Lines.Count - 1 do
    begin
      { Proses Enkripsi }
      k       := Random(p-2);
      c       := StrToInt(mTampung.Lines.Strings[i]);
      G_amma  := fastexp(alfa,k,p) mod p;
      D_elta  :=(c*fastexp(K_Publik,k,p)) mod p;
      gamma1  := Format('%.*d', [4, G_amma]); // Leading Zero - 4
digit, 1 jd 0001 dst;
      delta1  := Format('%.*d', [4, D_elta]);
      memo1.Lines.Add(gamma1 + delta1);
      { End of Proses Enkripsi }
    end;

  saveDialog1.DefaultExt  := 'elg';
  SaveDialog1.InitialDir  := ExtractFilePath(Application.ExeName) +
'data';

  if saveDialog1.Execute then
    begin
      nama:=saveDialog1.FileName;
```

```
      if fileExists(nama) then
        begin
          if MessageDlg('file '+nama+' sudah ada, tetap
disimpan?',MtConfirmation,
          [MByes,MBNo],0) = mrNo then exit
          else memo1.Lines.SaveToFile(nama);
        end;
      memo1.Lines.SaveToFile(nama);
    end;

  ModalResult := mrCancel;
  FKEnkripsi.Release;
end;

procedure TFKEnkripsi.eBprimaChange(Sender: TObject);
begin
if ((eBprima.Text <> '') and (eEprimitif.Text <> '') and
(eKpublik.Text <>'')) then
bbProses.Enabled:=true;
end;

procedure TFKEnkripsi.eEprimitifChange(Sender: TObject);
begin
if ((eBprima.Text <> '') and (eEprimitif.Text <> '') and
(eKpublik.Text <>'')) then
bbProses.Enabled:=true;

end;

procedure TFKEnkripsi.eKpublikChange(Sender: TObject);
begin
if ((eBprima.Text <> '') and (eEprimitif.Text <> '') and
(eKpublik.Text <>'')) then
bbProses.Enabled:=true;

end;

procedure TFKEnkripsi.eBprimaKeyPress(Sender: TObject; var Key:
Char);
begin
  if not (key in ['0'..'9',#8,#13]) then key := #0;
  if key = #13 then Perform(WM_NEXTDLGCTL,0,0);
end;

end.
```

9. Form Dekripsi

```
unit UDekripsi;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms,
  Dialogs, StdCtrls, Buttons, FileCtrl, Grids, Outline, DirOutln,
Menus,
  ExtCtrls;

type
  TFDekripsi = class(TForm)
    Label1: TLabel;
    DriveComboBox1: TDriveComboBox;
    DirectoryOutline1: TDirectoryOutline;
    Label2: TLabel;
    FileListBox1: TFileListBox;
    Label3: TLabel;
    ScrollBox1: TScrollBox;
    bbBatal: TBitBtn;
    BitBtn1: TBitBtn;
    Bevel1: TBevel;
    Bevel2: TBevel;
    Bevel3: TBevel;
    Bevel4: TBevel;
    MPesan: TMemo;
    procedure DriveComboBox1Change(Sender: TObject);
    procedure DirectoryOutline1Change(Sender: TObject);
    procedure FileListBox1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure bbBatalClick(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FDekripsi: TFDekripsi;

implementation

uses UUtama, UKDekripsi;
```

```
{$R *.dfm}

procedure TFDekripsi.DriveComboBox1Change(Sender: TObject);
begin
 DirectoryOutline1.Drive := DriveComboBox1.Drive;
end;

procedure TFDekripsi.DirectoryOutline1Change(Sender: TObject);
begin
  FileListBox1.Directory := DirectoryOutline1.Directory;
  FileListBox1.Mask := '*.elg';
end;

procedure TFDekripsi.FileListBox1Click(Sender: TObject);
begin
   MPesan.Lines.LoadFromFile(Filelistbox1.FileName);
end;

procedure TFDekripsi.FormCreate(Sender: TObject);
begin
  MPesan.Lines.Clear;
end;

procedure TFDekripsi.bbBatalClick(Sender: TObject);
begin
  ModalResult := mrCancel;
  FDekripsi.Release;
end;

procedure TFDekripsi.BitBtn1Click(Sender: TObject);
begin
  if MPesan.Text = '' then
    begin
      MessageDlg('Maaf, Data Kosong. Proses
Dibatalkan.',mtError,[mbok],0 );
      Exit;
    end;
  Application.CreateForm(TFKDekripsi, FKDekripsi);
  FKDekripsi.ShowModal;
end;

end.
```

10. Form Konfirmasi Kunci Dekripsi

```
unit UKDekripsi;

interface
```

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

type
  TFKDekripsi = class(TForm)
    Label1: TLabel;
    eBprima: TEdit;
    lKRahasia: TLabel;
    eKRahasia: TEdit;
    bbProses: TBitBtn;
    bbBatal: TBitBtn;
    Bevel1: TBevel;
    Memo1: TMemo;
    SaveDialog1: TSaveDialog;
    Bevel2: TBevel;
    procedure bbBatalClick(Sender: TObject);
    procedure bbProsesClick(Sender: TObject);
    procedure eBprimaKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var FKDekripsi: TFKDekripsi;

implementation
uses uDekripsi, UEnkripsi, StrUtils, uGlobal;
{$R *.dfm}

procedure TFKDekripsi.bbBatalClick(Sender: TObject);
begin
  FDekripsi.Show;
  Close;

end;

procedure TFKDekripsi.bbProsesClick(Sender: TObject);
var
  i,j,k,a,p           : integer;
  D_elta,G_amma       : Integer;
  nama,gamma1, delta1 : string;
  Hasil               : String;
begin
  { Inisialisasi }
  p     := StrToInt(eBprima.Text);
  a     := StrToInt(eKRahasia.Text);
```

```
  k     := p - 1 - a;
  Hasil := '';
  Memo1.Clear;
  { End Of Inisialisasi }

  for i := 0 to FDekripsi.MPesan.Lines.Count - 1 do
    begin
      gamma1  := LeftStr(FDekripsi.MPesan.Lines.Strings[i],4);  //
potong 4 digit dr kiri
      G_amma  := StrToInt(gamma1);
      delta1  := RightStr(FDekripsi.MPesan.Lines.Strings[i],4); //
Potong 4 digit dr kanan
      D_elta  := StrToInt(delta1);
      j       := (D_elta * fastexp(G_amma,k,p)) mod p;
      if j = 19 then
        Hasil   := Hasil + #13 + #10
        else Hasil   := Hasil + Chr(j);
    end;
  Memo1.Text  := Hasil;

  saveDialog1.DefaultExt  := 'txt';
  SaveDialog1.InitialDir  := ExtractFilePath(Application.ExeName) +
'data';

  if saveDialog1.Execute then
    begin
      nama:=saveDialog1.FileName;
      if fileExists(nama) then
        begin
          if MessageDlg('file '+nama+' sudah ada, tetap
disimpan?',MtConfirmation,
          [MByes,MBNo],0) = mrNo then exit
          else memo1.Lines.SaveToFile(nama);
        end;
      memo1.Lines.SaveToFile(nama);
    end;

 ModalResult  := mrCancel;
 FKDekripsi.Release;
end;

procedure TFKDekripsi.eBprimaKeyPress(Sender: TObject; var Key:
Char);
begin
  if not (key in ['0'..'9',#8,#13]) then key := #0;
  if key = #13 then Perform(WM_NEXTDLGCTL,0,0);
end;

end.
```

11. Form About

```
unit UAbout;

interface

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

type
  TFAbout = class(TForm)
    lJudul1: TLabel;
    Label2: TLabel;
    Label1: TLabel;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FAbout: TFAbout;

implementation

{$R *.dfm}

procedure TFAbout.Button1Click(Sender: TObject);
begin
  ModalResult := mrCancel;
  FAbout.Release;
end;

end.
```

12. Form Help

```
unit uHelp;

interface

uses
```

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

type
  TFHelp = class(TForm)
    StaticText1: TStaticText;
    StaticText2: TStaticText;
    StaticText3: TStaticText;
    StaticText4: TStaticText;
    StaticText5: TStaticText;
    StaticText6: TStaticText;
    StaticText7: TStaticText;
    StaticText8: TStaticText;
    StaticText9: TStaticText;
    StaticText10: TStaticText;
    StaticText11: TStaticText;
    StaticText12: TStaticText;
    StaticText13: TStaticText;
    StaticText14: TStaticText;
    StaticText15: TStaticText;
    StaticText16: TStaticText;
    StaticText17: TStaticText;
    StaticText18: TStaticText;
    bbOk: TBitBtn;
    procedure bbOkClick(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FHelp: TFHelp;

implementation

{$R *.dfm}



procedure TFHelp.bbOkClick(Sender: TObject);
begin
 Close;
end;

end.
```

LAMPIRAN C: TABEL KODE ASCII

Kode ASCII (0 – 127)

No.	Kode
0	NULL (null)
1	SOH (start of heading)
2	STX (start of text)
3	ETX (end of text)
4	EOT (end of transmission)
5	ENQ (enquiry
6	ACK (acknowledge)
7	BEL (bell)
8	BS (backspace)
9	TAB (horizontal tab)
10	LF (new line)
11	VT (vertical tab)
12	FF (new page)
13	CR (carriage return)
14	SO (shift out)
15	SI (shift in)
16	DLE (data link escape)
17	DC1 (device control 1)
18	DC1 (device control 2)
19	DC1 (device control 3)
20	DC1 (device control 4)
21	NAK (negative acknowledge)
22	SYN (synchronus idle)
23	ETB (end of trans. blok)
24	CAN (cancel)
25	EM (end of medium)
26	SUB (substitute)
27	ESC (escape)
28	FS (file separator)
29	GS (group separator)
30	RS (record separator)
31	US (unit separator)
32	Space
33	!
34	"
35	#
36	$
37	%
38	&
39	'
40	(
41	)
42	*
43	+

No.	Kode
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W

No.	Kode
88	X
89	Y
90	Z
91	[
92	\
93	]
94	^
95	_
96	`
97	a
98	b
99	c
100	d
101	e
102	f
103	g
104	h
105	i
106	j
107	k
108	l
109	m
110	n
111	o
112	p
113	q
114	r
115	s
116	t
117	u
118	v
119	w
120	x
121	y
122	z
123	{
124	|
125	}
126	~
127	DEL

Kode ASCII *Extended* (128 – 255)

128	Ç	144	É	161	í	177	▒	193	┴	209	╤	225	ß	241	±
129	ü	145	æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
130	é	146	Æ	163	ú	179	│	195	├	211	╙	227	π	243	≤
131	â	147	ô	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠
132	ä	148	ö	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
133	à	149	ò	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷
134	å	150	û	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
135	ç	151	ù	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
136	ê	152	ÿ	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙
137	ë	153	Ö	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
138	è	154	Ü	171	½	187	╗	203	╦	219	█	235	δ	251	√
139	ï	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	ⁿ
140	î	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²
141	ì	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■
142	Ä	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	
143	Å	160	á	176	░	192	└	208	╨	224	α	240	≡		