LAMPIRAN A

LISTING PROGRAM

FORM UTAMA

```
Private Sub cmdBrowse_Click()
CdbDialog.InitDir = "C:\"
CdbDialog.Filter = "Text File(*.txt)|*.txt"
CdbDialog.FileName = ""
CdbDialog.DialogTitle = "Open File"
CdbDialog.ShowOpen
txtSave.text = CdbDialog.FileName
If CdbDialog.FileName = "" Then
    Exit Sub
End If
    rtbplain.FileName = CdbDialog.FileName
    x = rtbplain.text
    pjg = Len(x)
    temp = Mid(x, pjg - 1, 2)
    For i = 1 To 2
    p = Mid(temp, i, 1)
    h = Hex(Asc(p))
    c = c & h
    Next i
    If c = "DA" Then
    hasil = Mid(x, 1, pjg - 2)
    Else
    hasil = x
    End If
txttime.text = ""
rtbcipher = ""
rtbplain.text = hasil
End Sub

Private Sub cmdBrowse2_Click()
CdbDialog.InitDir = "C:\"
CdbDialog.Filter = "Text File(*.txt.enc)|*.txt.enc"
CdbDialog.FileName = ""
CdbDialog.DialogTitle = "Save File"
CdbDialog.ShowOpen
txtSave2.text = CdbDialog.FileName
If CdbDialog.FileName = "" Then
    Exit Sub
End If
    rtbcipher.FileName = CdbDialog.FileName
    x = rtbcipher.text
    pjg = Len(x)
    temp = Mid(x, pjg - 1, 2)
    For i = 1 To 2
    p = Hex(Asc(Mid(temp, i, 1)))
    c = c & p
    Next i
    If c = "DA" Then
```

```
    hasil = Mid(x, 1, pjg - 2)
    Else
    hasil = x
    End If
txttime.text = ""
rtbplain = ""
rtbcipher.text = hasil
End Sub

Private Sub cmdCekSignVal_Click()
Dim signcek As Long
Dim vad As String

vad2 = CRCCheck(CStr(rtbplain.text))
vad = Hextobin(Mid(vad2, Len(vad2) - 1, 2))
h = CInt(BinTodes(vad))
Text9.text = h

If txtPublikE.text = "" Or txtPublikN.text = "" Then
    MsgBox "Input your publik key"
    Exit Sub
End If

AllText$ = ""
CommonDialog3.Filter = "Text files |*.TXT"
    CommonDialog3.ShowOpen
    If CommonDialog3.FileName <> "" Then
    frmMain.MousePointer = 11
    Open CommonDialog3.FileName For Input As #1
            Do Until EOF(1)          'then read lines from file
            Line Input #1, LineOfText$
            AllText$ = AllText$ & LineOfText$
        Loop
        txtValSign.text = ""
        txtValSign.text = AllText$ 'display file
        txtValSign.Enabled = True

    signcek    =    pangkatmod(txtValSign.text,    txtPublikE.text,
txtPublikN.text)
    Text10.text = signcek
CleanUp:
        CommonDialog3.FileName = ""
        frmMain.MousePointer = 0   'reset mouse
        Close #1                 'close file

    If Text9.text = Text10.text Then
        MsgBox "Data Verified", vbOKOnly, "ProjectFEAL"
        Exit Sub
    Else
        MsgBox "Data is not Valid or Wrong Signature", vbOKOnly,
"ProjectFEAL"
        Exit Sub

    End If
    End If
    Exit Sub
End Sub
```

```
Private Sub cmddekrip_Click()
Dim cteks, plain As String, kunci As String

cmdenkrip.Enabled = False
cmddekrip.Enabled = False

cteks = rtbcipher.text
kunci = txtkey2.text

If rtbcipher.text = "" Then
    S  =  MsgBox("Please  insert  text  into  CIPHERTEXT  box!",
vbExclamation, "ERROR")
    rtbcipher.SetFocus
    GoTo bawah
End If

vad1 = Mid(cteks, Len(cteks) - 7, 8)
cteks2 = Mid(cteks, 1, Len(cteks) - 8)

        If Not Len(kunci) = 8 Then
        S  =  MsgBox("Please  insert  keyword  in  8  character!",
vbExclamation, "ERROR")
        txtkey2.SetFocus
        GoTo bawah
        End If

        tawal = Timer

        plain = Dekripsi(CStr(kunci), CStr(cteks2))

        takhir = Timer
        txttime2.text = takhir - tawal

    vad2 = CRCCheck(CStr(plain))
    If Not vad1 = vad2 Then
        temp  =  MsgBox("Data  is  not  valid   ",  vbInformation,
"Information")
    End If

    rtbplain.text = plain

bawah:

cmdenkrip.Enabled = True
cmddekrip.Enabled = True
End Sub

Private Sub cmdenkrip_Click()
Dim pteks, kunci As String

cmdenkrip.Enabled = False
cmddekrip.Enabled = False

If rtbplain.text = "" Then
    S  =  MsgBox("Please  insert  text  into  PLAINTEXT  box!",
vbExclamation, "ERROR")
    rtbplain.SetFocus
    GoTo bawah
```

```
End If

pteks = rtbplain.text
kunci = txtkey.text
vad = CRCCheck(CStr(pteks))

        If Not Len(kunci) = 8 Then
        S  =  MsgBox("Please  insert  keyword  in  8  character!",
vbExclamation, "ERROR")
        txtkey.SetFocus
        GoTo bawah
        End If

        tawal = Timer

        cipher = Enkripsi(CStr(kunci), CStr(pteks))

        takhir = Timer
        txttime.text = takhir - tawal

    vcipher = cipher & vad

    On Error Resume Next
    rtbcipher.text = vcipher
    txtkey2.text = txtkey.text
    rtbplain.text = ""

bawah:

cmdenkrip.Enabled = True
cmddekrip.Enabled = True
End Sub

Private Sub cmdGen_Click()
Dim p As Long 'random prime
Dim q As Long 'second random prime that not equal to p
Dim n As Long 'p * q
Dim pi As Long '(p - 1)(q - 1)
Dim e As Long 'e that relatively prime to pi but less than pi
Dim d As Long 'd that d*e congruent to 1 mod pi
Dim i1 As Long 'counter
Dim c As Long
Dim temp1 As Long
Dim  temp2()  As  Long  '  temp  dynamic  array  handler  that  hand
selection of e
Dim temp3 As Long
Dim temp4 As Long
Dim temp5 As Long ' temp handler
Dim temp6 As Long ' temp handler 2

frmMain.MousePointer = 11
repeat:
    p = RdmPrime
    q = RdmPrime2

    'Trap handler if p = q
    If p = q Then
    GoTo repeat
```

```
    End If

    n = p * q

Text7.text = n
cpyText7.text = n
    pi = (p - 1) * (q - 1)

'search for e
c = pi - 1
ReDim temp2(c)

For e = 2 To (pi - 1)
    temp6 = gcd(pi, e)

    If temp6 = 1 Then
    temp2(c) = e
    c = c - 1
    End If

Next

'random selection of e
ulang:
    Randomize
    temp3 = Int((pi - 1) * Rnd)
    temp4 = temp2(temp3)
    If temp4 = 0 Or temp4 = Null Then
    GoTo ulang
    End If

'select e that is prime
For i1 = 2 To (Sqr(temp4))
temp5 = temp4 Mod i1
    If temp5 = 0 Then
      GoTo ulang
    End If
Next i1

Text5.text = 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
End If

Text6.text = d

frmMain.MousePointer = 0
End Sub

Private Sub cmdValSign_Click()
Dim vad As String
Dim vad2 As Long
Dim sign As Long

If Text6.text = "" Then
```

```
    MsgBox "Click the generate button"
    Exit Sub
End If
pteks = rtbplain.text
temp = CRCCheck(CStr(pteks))
vad = Hextobin(Mid(temp, Len(temp) - 1, 2))
vad2 = CInt(BinTodes(vad))

sign = pangkatmod(vad2, Text6.text, Text7.text)
Text8.text = sign

If Text8.text = "" Then
        MsgBox "Error signing the plain document"
        Exit Sub
    Else
        CommonDialog2.Filter = "Text files (*.TXT)|*.TXT"
        CommonDialog2.ShowSave
            If CommonDialog2.FileName <> "" Then
            Open CommonDialog2.FileName For Output As #1
            Print #1, Text8.text
            CommonDialog1.FileName = ""
            Close #1
            End If
        temp = MsgBox("Document has been signed", vbInformation,
"information")
End If
Text8.text = ""
Text6.text = ""
End Sub

Private Sub Form_Load()
Me.Move (Screen.Width - Me.Width) \ 2, _
            (Screen.Height - Me.Height) \ 2
End Sub

Private Sub keygen_Click()
 For i = 1 To 4
    Randomize
    dat = Hex(Rnd * 255)
    If Len(dat) = 1 Then dat = "0" & dat
    Key = Key & dat
    Next i
    txtkey.text = Key
End Sub

Private Sub mnAbout_Click()
    Load frmAbout
    frmAbout.Show
End Sub

Private Sub mnhowto_Click()
    Load FrmHow
    FrmHow.Show
End Sub

Private Sub mnbuka1_Click()
 CdbDialog.InitDir = "C:\"
CdbDialog.Filter = "Text File(*.txt)|*.txt"
```

```
CdbDialog.FileName = ""
CdbDialog.DialogTitle = "Open File"
CdbDialog.ShowOpen
If CdbDialog.FileName = "" Then
    Exit Sub
End If
    rtbplain.FileName = CdbDialog.FileName
    x = rtbplain.text
    pjg = Len(x)
    temp = Mid(x, pjg - 1, 2)
    For i = 1 To 2
    p = Mid(temp, i, 1)
    h = Hex(Asc(p))
    c = c & h
    Next i
    If c = "DA" Then
    hasil = Mid(x, 1, pjg - 2)
    Else
    hasil = x
    End If
txttime.text = ""
rtbcipher = ""
rtbplain.text = hasil
End Sub

Private Sub mnbuka2_Click()
CdbDialog.InitDir = "C:\"
CdbDialog.Filter = "Text File(*.txt.enc)|*.txt.enc"
CdbDialog.FileName = ""
CdbDialog.DialogTitle = "Save File"
CdbDialog.ShowOpen
If CdbDialog.FileName = "" Then
    Exit Sub
End If
    rtbcipher.FileName = CdbDialog.FileName
    x = rtbcipher.text
    pjg = Len(x)
    temp = Mid(x, pjg - 1, 2)
    For i = 1 To 2
    p = Hex(Asc(Mid(temp, i, 1)))
    c = c & p
    Next i
    If c = "DA" Then
    hasil = Mid(x, 1, pjg - 2)
    Else
    hasil = x
    End If
txttime.text = ""
rtbplain = ""
rtbcipher.text = hasil
End Sub

Private Sub mnexit_Click()
End
End Sub
```

```
Private Sub mnhapus1_Click()
rtbplain = ""
End Sub

Private Sub mnhapus2_Click()
rtbcipher = ""
End Sub

Private Sub mnreset_Click()
rtbplain = ""
rtbcipher = ""
txtkey = ""
txtkey2 = ""
txttime = ""
txttime2 = ""
txtSave = ""
txtSave2 = ""
Text6 = ""
Text5 = ""
Text7 = ""
cpyText7 = ""
txtValSign = ""
txtPublikE = ""
txtPublikN = ""
Text9 = ""
Text10 = ""
End Sub

Private Sub mnsimpan1_Click()
CdbDialog.DialogTitle = "Save File"
CdbDialog.Filter = "Text File(*.txt)|*.txt"
CdbDialog.FileName = ""
CdbDialog.ShowSave

simpan = CdbDialog.FileName
If simpan = "" Then
    Exit Sub
End If

    hasil = rtbplain.text

Open simpan For Output As #1
    Print #1, hasil
Close #1
End Sub

Private Sub mnsimpan2_Click()
CdbDialog.DialogTitle = "Save File"
CdbDialog.Filter = "Text File(*.txt.enc)|*.txt.enc"
CdbDialog.FileName = ""
CdbDialog.ShowSave

simpan = CdbDialog.FileName
If simpan = "" Then
    Exit Sub
End If

    hasil = rtbcipher.text
```

```
Open simpan For Output As #1
    Print #1, hasil
Close #1
End Sub
```

FORM ABOUT

```
Private Sub cmdOK_Click()
Unload frmAbout
End Sub
```

FORM CARA KERJA

```
Private Sub cmdFEAL_Click()
Unload FrmHow
End Sub
```

FORM SPLASH

```
Private Sub Form_KeyPress(KeyAscii As Integer)
  Unload Me
End Sub

Private Sub Form_Load()
  imgLogo.Picture = LoadPicture(App.Path & "\Loading2.gif")
End Sub

Private Sub Timer1_Timer()
  Unload Me
  frmMain.Show
End Sub
```

MODUL DIGITAL SIGNATURE

```
Option Explicit
Function Euclid(ByVal nilai1, ByVal nilai2) As Long
Dim mex As Long
Dim bex As Long
Dim A1 As Long
Dim A2 As Long
Dim A3 As Long
Dim Qex As Long
Dim T1 As Long
Dim T2 As Long
Dim T3 As Long
Dim B1 As Long
Dim B2 As Long
Dim B3 As Long
Dim hasil As Long

mex = nilai1
```

```
bex = nilai2

A1 = 1
A2 = 0
A3 = mex

B1 = 0
B2 = 1
B3 = bex

itung:

If B3 = 0 Then
hasil = 0
GoTo selesai
End If

If B3 = 1 Then
hasil = B2
GoTo selesai
End If

Qex = A3 \ B3

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

A1 = B1
A2 = B2
A3 = B3

B1 = T1
B2 = T2
B3 = T3

GoTo itung

selesai:
Euclid = hasil
End Function

Function gcd(ByVal p, ByVal q) As Long
Dim A11 As Long
Dim B11 As Long
Dim R11 As Long

    A11 = p
    B11 = q

label:
    If B11 = 0 Then
    gcd = A11
    Else
    R11 = A11 Mod B11
    A11 = B11
    B11 = R11
    GoTo label
```

```
    End If
End Function

Function RdmPrime() As Long
Dim iRandom As Long ' holds random long result
Dim i2 As Long ' checkprime loop counter
Dim temp2a As Long 'swap var

Const iLowerBound = 30
Const iUpperBound = 300

    Randomize

110
    iRandom = (Int((iUpperBound - iLowerBound + 1) * Rnd() +
iLowerBound))
'trap handler

If iRandom = 0 Or iRandom = 1 Then
GoTo 110
End If

  'check number
  For i2 = 2 To (Sqr(iRandom))
    temp2a = iRandom Mod i2
    If temp2a = 0 Then
      GoTo 110
    End If
  Next i2

    RdmPrime = iRandom
End Function

Function RdmPrime2() As Long
Dim iRandom2 As Long ' holds random long result
Dim y As Long 'checkprime loop counter
Dim holder As Long 'swap var

   Randomize

120
    iRandom2 = (Int(Asc(Date) Xor 255 * Rnd))
'trap handler

If iRandom2 = 0 Or iRandom2 = 1 Then
GoTo 120
End If

'check number
For y = 2 To (Sqr(iRandom2))
    holder = iRandom2 Mod y
    If holder = 0 Then
      GoTo 120
    End If
Next y

    RdmPrime2 = iRandom2
End Function
```

```
Function pangkatmod(ByVal num1, ByVal num2, ByVal num3) As Long
'Dim pangkat As Long
Dim a22 As Long
Dim b22 As Long
Dim n22 As Long
Dim nilaimod As Long
Dim nilaic As Long
Dim nilaid As Long
Dim nilaii As Long
Dim nilaik As Long
Dim naik As Long
Dim temp() As Variant
Dim decbin As Variant

a22 = num1
b22 = num2
n22 = num3

decbin = DecimalToBinary(b22)
nilaic = 0
nilaid = 1
nilaik = Len(decbin)
ReDim temp(nilaik)
naik = 1

For nilaii = nilaik - 1 To 0 Step -1
temp(nilaii) = Mid(decbin, naik, 1)
naik = naik + 1
nilaic = 2 * nilaic
nilaid = (nilaid * nilaid) Mod n22
    If temp(nilaii) = 1 Then
        nilaic = nilaic + 1
        nilaid = (nilaid * a22) Mod n22
    End If
Next

pangkatmod = nilaid
End Function
```

MODUL FEAL

```
Private CRC32Table(255) As Long

Public Function Enkripsi(ByVal kunci As String, ByVal pteks As
String) As String
Dim k(0 To 15, 1 To 2) As Byte
Dim a(0 To 8) As Integer, b(0 To 3) As Integer, c(0 To 3) As
Integer, d(0 To 3) As Integer, e(0 To 3) As Integer, f(1 To 4) As
Integer, g(1 To 4) As Integer

pteks2 = CStr(Pad(pteks))
For i = 1 To Len(kunci)
a(i - 1) = Asc(Mid(kunci, i, 1))
Next i
```

```
b(0) = a(4)
b(1) = a(5)
b(2) = a(6)
b(3) = a(7)

c(1) = ShiftLeft2Bits(((a(0) Xor a(1)) + (b(0) Xor (a(2) Xor
a(3))) + 1) Mod 256)
c(0) = ShiftLeft2Bits(((c(1) Xor b(2)) + a(0)) Mod 256)
c(2) = ShiftLeft2Bits(((c(1) Xor b(1)) + (a(2) Xor a(3))) Mod 256)
c(3) = ShiftLeft2Bits((((c(2) Xor b(3)) + a(3)) + 1) Mod 256)

k(0, 1) = c(0)
k(0, 2) = c(1)
k(1, 1) = c(2)
k(1, 2) = c(3)

For h = 2 To 15 Step 2
d(0) = b(0)
d(1) = b(1)
d(2) = b(2)
d(3) = b(3)

b(0) = (a(0) Xor c(0))
b(1) = (a(1) Xor c(1))
b(2) = (a(2) Xor c(2))
b(3) = (a(3) Xor c(3))

a(0) = d(0)
a(1) = d(1)
a(2) = d(2)
a(3) = d(3)

c(1) = ShiftLeft2Bits(((a(0) Xor a(1)) + (b(0) Xor (a(2) Xor
a(3))) + 1) Mod 256)
c(0) = ShiftLeft2Bits(((c(1) Xor b(2)) + a(0)) Mod 256)
c(2) = ShiftLeft2Bits(((c(1) Xor b(1)) + (a(2) Xor a(3))) Mod 256)
c(3) = ShiftLeft2Bits((((c(2) Xor b(3)) + a(3)) + 1) Mod 256)

k(h, 1) = c(0)
k(h, 2) = c(1)
k((h + 1), 1) = c(2)
k((h + 1), 2) = c(3)
Next h

'Awal Enkripsi terhadap Plaintext

For j = 1 To Len(pteks2) Step 8
    For m = 1 To 8
    a(m) = Asc(Mid(pteks2, m + j - 1, 1))
    Next m

    a(1) = a(1) Xor k(8, 1)
    a(2) = a(2) Xor k(8, 2)
    a(3) = a(3) Xor k(9, 1)
    a(4) = a(4) Xor k(9, 2)
    a(5) = a(5) Xor k(10, 1)
    a(6) = a(6) Xor k(10, 2)
    a(7) = a(7) Xor k(11, 1)
```

```
 a(8) = a(8) Xor k(11, 2)

 a(5) = a(1) Xor a(5)
 a(6) = a(2) Xor a(6)
 a(7) = a(3) Xor a(7)
 a(8) = a(4) Xor a(8)

 e(1) = ShiftLeft2Bits((((a(6) Xor k(0, 1)) Xor a(5)) + ((a(7)
        Xor k(0, 2)) Xor a(8)) + 1) Mod 256)
 e(0) = ShiftLeft2Bits((a(5) + e(1)) Mod 256)
 e(2) = ShiftLeft2Bits((((a(7) Xor k(0, 2)) Xor a(8)) + e(1))
        Mod 256)
 e(3) = ShiftLeft2Bits((a(8) + e(2) + 1) Mod 256)

 a(1) = a(1) Xor e(0)
 a(2) = a(2) Xor e(1)
 a(3) = a(3) Xor e(2)
 a(4) = a(4) Xor e(3)

For u = 1 To 7
 f(1) = a(1)
 f(2) = a(2)
 f(3) = a(3)
 f(4) = a(4)

 e(1) = ShiftLeft2Bits((((a(2) Xor k(u, 1)) Xor a(1)) + ((a(3)
        Xor k(u, 2)) Xor a(4)) + 1) Mod 256)
 e(0) = ShiftLeft2Bits((a(1) + e(1)) Mod 256)
 e(2) = ShiftLeft2Bits((((a(3) Xor k(u, 2)) Xor a(4)) + e(1))
        Mod 256)
 e(3) = ShiftLeft2Bits((a(4) + e(2) + 1) Mod 256)

 a(1) = a(5) Xor e(0)
 a(2) = a(6) Xor e(1)
 a(3) = a(7) Xor e(2)
 a(4) = a(8) Xor e(3)

 a(5) = f(1)
 a(6) = f(2)
 a(7) = f(3)
 a(8) = f(4)
 Next u

 a(5) = a(1) Xor a(5)
 a(6) = a(2) Xor a(6)
 a(7) = a(3) Xor a(7)
 a(8) = a(4) Xor a(8)

 'a(1-8) dibalik

 g(1) = a(1)
 g(2) = a(2)
 g(3) = a(3)
 g(4) = a(4)

 a(1) = a(5)
 a(2) = a(6)
 a(3) = a(7)
```

```
    a(4) = a(8)

    a(5) = g(1)
    a(6) = g(2)
    a(7) = g(3)
    a(8) = g(4)

    a(1) = a(1) Xor k(12, 1)
    a(2) = a(2) Xor k(12, 2)
    a(3) = a(3) Xor k(13, 1)
    a(4) = a(4) Xor k(13, 2)
    a(5) = a(5) Xor k(14, 1)
    a(6) = a(6) Xor k(14, 2)
    a(7) = a(7) Xor k(15, 1)
    a(8) = a(8) Xor k(15, 2)

    For r = 1 To 8
    Mid(pteks2, r + j - 1, 1) = Chr(a(r))
    Next r

Next j

Enkripsi = pteks2
End Function

Public Function Pad(text As String) As String
x = Len(text) Mod 8
For i = (7 - x) To 1 Step -1
    text = text & "0"
Next i
y = 8 - x
text = text & y
    Pad = text
End Function

Public Function ShiftLeft2Bits(huruf As String) As String
Dim temp As String
Dim pjghuruf As Integer, ordebit As Integer
Dim tempdes As Byte

    temp = huruf
    temp = Hex(temp)
    If Len(temp) = 1 Then temp = "0" & temp
    huruf = ""

    ' Konversi ke biner

    For pjghuruf = 1 To Len(temp)
        tempdes = Val("&H" & Mid(temp, pjghuruf, 1))
        For ordebit = 3 To 0 Step -1
            If tempdes And 2 ^ ordebit Then
                huruf = huruf + "1"
            Else
                huruf = huruf + "0"
            End If
        Next ordebit
    Next pjghuruf
```

```
        y = Left(huruf, 2)
        Z = Right(huruf, Len(huruf) - 2)
        huruf = Z + y

    temp = huruf

     ' Konversi ke desimal

    tempdes = 0
    For pjghuruf = 0 To Len(temp) - 1
        If Val(Mid(temp, pjghuruf + 1, 1)) Then
        tempdes = tempdes + 2 ^ (Len(temp) - 1 - pjghuruf)
        End If
    Next pjghuruf

    ShiftLeft2Bits = tempdes
End Function

Public Function Dekripsi(ByVal kunci As String, ByVal cteks As
String) As String
Dim k(0 To 15, 1 To 2) As Byte
Dim a(0 To 8) As Integer, b(0 To 3) As Integer, c(0 To 3) As
Integer, d(0 To 3) As Integer, e(0 To 3) As Integer, f(1 To 4) As
Integer, g(1 To 4) As Integer, l(1 To 4) As Integer

For i = 1 To Len(kunci)
a(i - 1) = Asc(Mid(kunci, i, 1))
Next i

b(0) = a(4)
b(1) = a(5)
b(2) = a(6)
b(3) = a(7)

c(1) = ShiftLeft2Bits(((a(0) Xor a(1)) + (b(0) Xor (a(2) Xor
a(3))) + 1) Mod 256)
c(0) = ShiftLeft2Bits(((c(1) Xor b(2)) + a(0)) Mod 256)
c(2) = ShiftLeft2Bits(((c(1) Xor b(1)) + (a(2) Xor a(3))) Mod 256)
c(3) = ShiftLeft2Bits((((c(2) Xor b(3)) + a(3)) + 1) Mod 256)

k(0, 1) = c(0)
k(0, 2) = c(1)
k(1, 1) = c(2)
k(1, 2) = c(3)

For h = 2 To 15 Step 2
d(0) = b(0)
d(1) = b(1)
d(2) = b(2)
d(3) = b(3)

b(0) = (a(0) Xor c(0))
b(1) = (a(1) Xor c(1))
b(2) = (a(2) Xor c(2))
b(3) = (a(3) Xor c(3))

a(0) = d(0)
a(1) = d(1)
```

```
a(2) = d(2)
a(3) = d(3)

c(1) = ShiftLeft2Bits(((a(0) Xor a(1)) + (b(0) Xor (a(2) Xor
a(3))) + 1) Mod 256)
c(0) = ShiftLeft2Bits(((c(1) Xor b(2)) + a(0)) Mod 256)
c(2) = ShiftLeft2Bits(((c(1) Xor b(1)) + (a(2) Xor a(3))) Mod 256)
c(3) = ShiftLeft2Bits((((c(2) Xor b(3)) + a(3)) + 1) Mod 256)

k(h, 1) = c(0)
k(h, 2) = c(1)
k((h + 1), 1) = c(2)
k((h + 1), 2) = c(3)
Next h

'Awal Dekripsi terhadap chipertext

For j = 1 To Len(cteks) Step 8
    For m = 1 To 8
    If Not (Mid(cteks, m + j - 1, 1)) = "" Then
    a(m) = Asc(Mid(cteks, m + j - 1, 1))
    End If
    Next m

    a(1) = a(1) Xor k(12, 1)
    a(2) = a(2) Xor k(12, 2)
    a(3) = a(3) Xor k(13, 1)
    a(4) = a(4) Xor k(13, 2)
    a(5) = a(5) Xor k(14, 1)
    a(6) = a(6) Xor k(14, 2)
    a(7) = a(7) Xor k(15, 1)
    a(8) = a(8) Xor k(15, 2)

    l(1) = a(1)
    l(2) = a(2)
    l(3) = a(3)
    l(4) = a(4)

    a(1) = a(5)
    a(2) = a(6)
    a(3) = a(7)
    a(4) = a(8)

    a(5) = l(1)
    a(6) = l(2)
    a(7) = l(3)
    a(8) = l(4)

    a(5) = a(1) Xor a(5)
    a(6) = a(2) Xor a(6)
    a(7) = a(3) Xor a(7)
    a(8) = a(4) Xor a(8)

    e(1) = ShiftLeft2Bits((((a(6) Xor k(7, 1)) Xor a(5)) + ((a(7)
           Xor k(7, 2)) Xor a(8)) + 1) Mod 256)
    e(0) = ShiftLeft2Bits((a(5) + e(1)) Mod 256)
    e(2) = ShiftLeft2Bits((((a(7) Xor k(7, 2)) Xor a(8)) + e(1))
           Mod 256)
```

```
    e(3) = ShiftLeft2Bits((a(8) + e(2) + 1) Mod 256)

    a(1) = a(1) Xor e(0)
    a(2) = a(2) Xor e(1)
    a(3) = a(3) Xor e(2)
    a(4) = a(4) Xor e(3)

 For u = 6 To 0 Step -1
    f(1) = a(1)
    f(2) = a(2)
    f(3) = a(3)
    f(4) = a(4)

    e(1) = ShiftLeft2Bits((((a(2) Xor k(u, 1)) Xor a(1)) + ((a(3)
           Xor k(u, 2)) Xor a(4)) + 1) Mod 256)
    e(0) = ShiftLeft2Bits((a(1) + e(1)) Mod 256)
    e(2) = ShiftLeft2Bits((((a(3) Xor k(u, 2)) Xor a(4)) + e(1))
           Mod 256)
    e(3) = ShiftLeft2Bits((a(4) + e(2) + 1) Mod 256)

    a(1) = a(5) Xor e(0)
    a(2) = a(6) Xor e(1)
    a(3) = a(7) Xor e(2)
    a(4) = a(8) Xor e(3)

    a(5) = f(1)
    a(6) = f(2)
    a(7) = f(3)
    a(8) = f(4)
    Next u

    a(5) = a(1) Xor a(5)
    a(6) = a(2) Xor a(6)
    a(7) = a(3) Xor a(7)
    a(8) = a(4) Xor a(8)

    a(1) = a(1) Xor k(8, 1)
    a(2) = a(2) Xor k(8, 2)
    a(3) = a(3) Xor k(9, 1)
    a(4) = a(4) Xor k(9, 2)
    a(5) = a(5) Xor k(10, 1)
    a(6) = a(6) Xor k(10, 2)
    a(7) = a(7) Xor k(11, 1)
    a(8) = a(8) Xor k(11, 2)

    For r = 1 To 8
    If Not (Mid(cteks, r + j - 1, 1)) = "" Then
    Mid(cteks, r + j - 1, 1) = Chr(a(r))
    End If
    Next r

Next j

Dekripsi = unPad(CStr(cteks))
End Function
```

```
Public Function unPad(inp As String) As String
On Error GoTo ErrunPad
    x = Len(inp)
    y = Mid(inp, x, 1)
    b = CLng(x - y)
    Z = CStr(Mid(inp, 1, b))
    unPad = Z
Exit Function

ErrunPad:
    unPad = inp
End Function

Public Function Hextobin(Hex As String) As String
Dim temp As String, hasil As String
Dim i As Integer, j As Integer, tempdes As Integer

temp = Hex
hasil = ""
If Len(temp) = 1 Then temp = "0" & temp

For i = 1 To Len(temp)
 tempdes = Val("&H" + Mid$(temp, i, 1))
 For j = 3 To 0 Step -1
    If tempdes And 2 ^ j Then
        hasil = hasil + "1"
       Else
        hasil = hasil + "0"
       End If
  Next j
Next i
Hextobin = hasil
End Function

Public Function BinTodes(Binary As String) As String
Dim Bit As String, temphex As String
Dim i As Long, j As Long, tempdes As Long
Bit = Binary
tempdes = 0
For i = 0 To Len(Bit) - 1
    j = Val(Mid$(Bit, i + 1, 1))
    If j = 1 Then
    tempdes = tempdes + (2 ^ (Len(Bit) - 1 - i))
    Else
    tempdes = tempdes + 0
    End If
    Next i

  BinTodes = CStr(tempdes)
End Function

Public Function DecimalToBinary(Des As Long) As String
Dim temp As String
Dim n As Long

n = Des

temp = Trim(Str(n Mod 2))
```

```
n = n \ 2

Do While n <> 0
temp = Trim(Str(n Mod 2)) & temp
n = n \ 2
Loop

DecimalToBinary = temp
End Function

Public Function CRCCheck(sMessage As String) As String
    Dim iCRC As Long
    Dim bytT As Byte
    Dim bytC As Byte
    Dim lngA As Long

    Call CRC32Setup

    iCRC = &HFFFFFFFF
    For i = 1 To Len(sMessage)
        bytC = Asc(Mid(sMessage, i, 1))
        bytT = (iCRC And &HFF) Xor bytC
        lngA = ShiftRight8(iCRC)
        iCRC = lngA Xor CRC32Table(bytT)
    Next

    CRC = iCRC Xor &HFFFFFFFF
    CRCCheck = Hex(CRC)

    If Not (Len(CRCCheck) / 8) = (Len(CRCCheck) \ 8) Then
        For i = Len(CRCCheck) To 7
        CRCCheck = CRCCheck & "0"
        Next i
    End If
End Function

Public Function ShiftRight8(x As Long) As Long
    Dim iNew As Long
    iNew = (x And &H7FFFFFFF) \ 256
    If (x And &H80000000) <> 0 Then
        iNew = iNew Or &H800000
    End If
    ShiftRight8 = iNew
End Function

Public Function CRC32Setup()
    Static bDone As Boolean
    Dim vntA As Variant
    Dim i As Integer, iOffset As Integer
    Dim nLen As Integer

    If bDone Then
        Exit Function
    End If

    iOffset = 0
    nLen = 32
    vntA = Array( _
```

```
    &H0, &H77073096, &HEE0E612C, &H990951BA, _
    &H76DC419, &H706AF48F, &HE963A535, &H9E6495A3, _
    &HEDB8832, &H79DCB8A4, &HE0D5E91E, &H97D2D988, _
    &H9B64C2B, &H7EB17CBD, &HE7B82D07, &H90BF1D91, _
    &H1DB71064, &H6AB020F2, &HF3B97148, &H84BE41DE, _
    &H1ADAD47D, &H6DDDE4EB, &HF4D4B551, &H83D385C7, _
    &H136C9856, &H646BA8C0, &HFD62F97A, &H8A65C9EC, _
    &H14015C4F, &H63066CD9, &HFA0F3D63, &H8D080DF5)

For i = iOffset To iOffset + nLen - 1
    CRC32Table(i) = vntA(i - iOffset)
Next
iOffset = iOffset + nLen

vntA = Array( _
    &H3B6E20C8, &H4C69105E, &HD56041E4, &HA2677172, _
    &H3C03E4D1, &H4B04D447, &HD20D85FD, &HA50AB56B, _
    &H35B5A8FA, &H42B2986C, &HDBBBC9D6, &HACBCF940, _
    &H32D86CE3, &H45DF5C75, &HDCD60DCF, &HABD13D59, _
    &H26D930AC, &H51DE003A, &HC8D75180, &HBFD06116, _
    &H21B4F4B5, &H56B3C423, &HCFBA9599, &HB8BDA50F, _
    &H2802B89E, &H5F058808, &HC60CD9B2, &HB10BE924, _
    &H2F6F7C87, &H58684C11, &HC1611DAB, &HB6662D3D)

For i = iOffset To iOffset + nLen - 1
    CRC32Table(i) = vntA(i - iOffset)
Next
iOffset = iOffset + nLen

vntA = Array( _
    &H76DC4190, &H1DB7106, &H98D220BC, &HEFD5102A, _
    &H71B18589, &H6B6B51F, &H9FBFE4A5, &HE8B8D433, _
    &H7807C9A2, &HF00F934, &H9609A88E, &HE10E9818, _
    &H7F6A0DBB, &H86D3D2D, &H91646C97, &HE6635C01, _
    &H6B6B51F4, &H1C6C6162, &H856530D8, &HF262004E, _
    &H6C0695ED, &H1B01A57B, &H8208F4C1, &HF50FC457, _
    &H65B0D9C6, &H12B7E950, &H8BBEB8EA, &HFCB9887C, _
    &H62DD1DDF, &H15DA2D49, &H8CD37CF3, &HFBD44C65)

For i = iOffset To iOffset + nLen - 1
    CRC32Table(i) = vntA(i - iOffset)
Next
iOffset = iOffset + nLen

vntA = Array( _
    &H4DB26158, &H3AB551CE, &HA3BC0074, &HD4BB30E2, _
    &H4ADFA541, &H3DD895D7, &HA4D1C46D, &HD3D6F4FB, _
    &H4369E96A, &H346ED9FC, &HAD678846, &HDA60B8D0, _
    &H44042D73, &H33031DE5, &HAA0A4C5F, &HDD0D7CC9, _
    &H5005713C, &H270241AA, &HBE0B1010, &HC90C2086, _
    &H5768B525, &H206F85B3, &HB966D409, &HCE61E49F, _
    &H5EDEF90E, &H29D9C998, &HB0D09822, &HC7D7A8B4, _
    &H59B33D17, &H2EB40D81, &HB7BD5C3B, &HC0BA6CAD)

For i = iOffset To iOffset + nLen - 1
    CRC32Table(i) = vntA(i - iOffset)
Next
iOffset = iOffset + nLen
```

```
vntA = Array( _
    &HEDB88320, &H9ABFB3B6, &H3B6E20C, &H74B1D29A, _
    &HEAD54739, &H9DD277AF, &H4DB2615, &H73DC1683, _
    &HE3630B12, &H94643B84, &HD6D6A3E, &H7A6A5AA8, _
    &HE40ECF0B, &H9309FF9D, &HA00AE27, &H7D079EB1, _
    &HF00F9344, &H8708A3D2, &H1E01F268, &H6906C2FE, _
    &HF762575D, &H806567CB, &H196C3671, &H6E6B06E7, _
    &HFED41B76, &H89D32BE0, &H10DA7A5A, &H67DD4ACC, _
    &HF9B9DF6F, &H8EBEEFF9, &H17B7BE43, &H60B08ED5)

For i = iOffset To iOffset + nLen - 1
    CRC32Table(i) = vntA(i - iOffset)
Next
iOffset = iOffset + nLen

vntA = Array( _
    &HD6D6A3E8, &HA1D1937E, &H38D8C2C4, &H4FDFF252, _
    &HD1BB67F1, &HA6BC5767, &H3FB506DD, &H48B2364B, _
    &HD80D2BDA, &HAF0A1B4C, &H36034AF6, &H41047A60, _
    &HDF60EFC3, &HA867DF55, &H316E8EEF, &H4669BE79, _
    &HCB61B38C, &HBC66831A, &H256FD2A0, &H5268E236, _
    &HCC0C7795, &HBB0B4703, &H220216B9, &H5505262F, _
    &HC5BA3BBE, &HB2BD0B28, &H2BB45A92, &H5CB36A04, _
    &HC2D7FFA7, &HB5D0CF31, &H2CD99E8B, &H5BDEAE1D)

For i = iOffset To iOffset + nLen - 1
    CRC32Table(i) = vntA(i - iOffset)
Next
iOffset = iOffset + nLen

vntA = Array( _
    &H9B64C2B0, &HEC63F226, &H756AA39C, &H26D930A, _
    &H9C0906A9, &HEB0E363F, &H72076785, &H5005713, _
    &H95BF4A82, &HE2B87A14, &H7BB12BAE, &HCB61B38, _
    &H92D28E9B, &HE5D5BE0D, &H7CDCEFB7, &HBDBDF21, _
    &H86D3D2D4, &HF1D4E242, &H68DDB3F8, &H1FDA836E, _
    &H81BE16CD, &HF6B9265B, &H6FB077E1, &H18B74777, _
    &H88085AE6, &HFF0F6A70, &H66063BCA, &H11010B5C, _
    &H8F659EFF, &HF862AE69, &H616BFFD3, &H166CCF45)

For i = iOffset To iOffset + nLen - 1
    CRC32Table(i) = vntA(i - iOffset)
Next
iOffset = iOffset + nLen

vntA = Array( _
    &HA00AE278, &HD70DD2EE, &H4E048354, &H3903B3C2, _
    &HA7672661, &HD06016F7, &H4969474D, &H3E6E77DB, _
    &HAED16A4A, &HD9D65ADC, &H40DF0B66, &H37D83BF0, _
    &HA9BCAE53, &HDEBB9EC5, &H47B2CF7F, &H30B5FFE9, _
    &HBDBDF21C, &HCABAC28A, &H53B39330, &H24B4A3A6, _
    &HBAD03605, &HCDD70693, &H54DE5729, &H23D967BF, _
    &HB3667A2E, &HC4614AB8, &H5D681B02, &H2A6F2B94, _
    &HB40BBE37, &HC30C8EA1, &H5A05DF1B, &H2D02EF8D)

For i = iOffset To iOffset + nLen - 1
    CRC32Table(i) = vntA(i - iOffset)
```

```
    Next
    iOffset = iOffset + nLen

    bDone = True
End Function
```

LAMPIRAN B
DIAGRAM ALIR

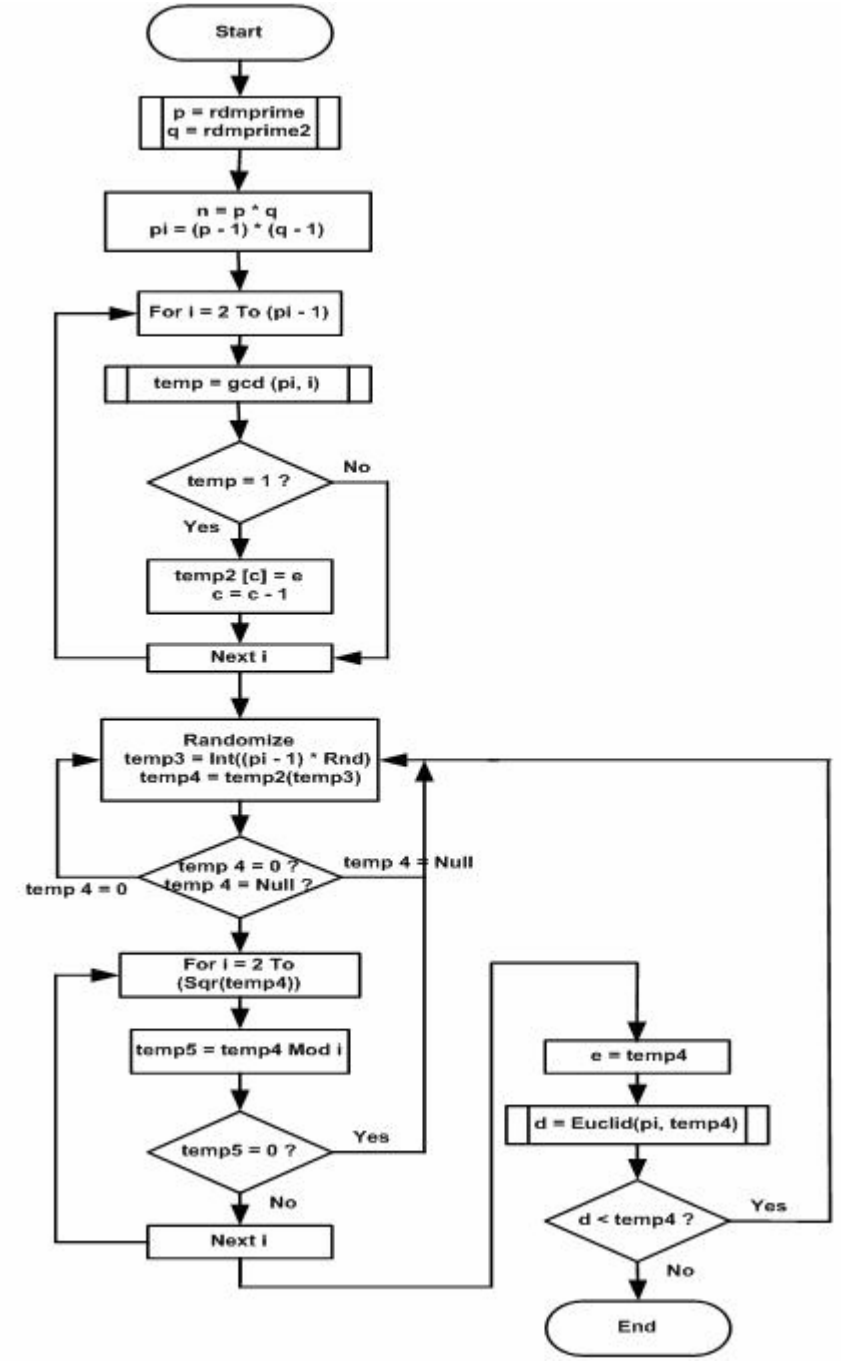


Diagram Alir Program Pembangkit Kunci Tanda Tangan Digital

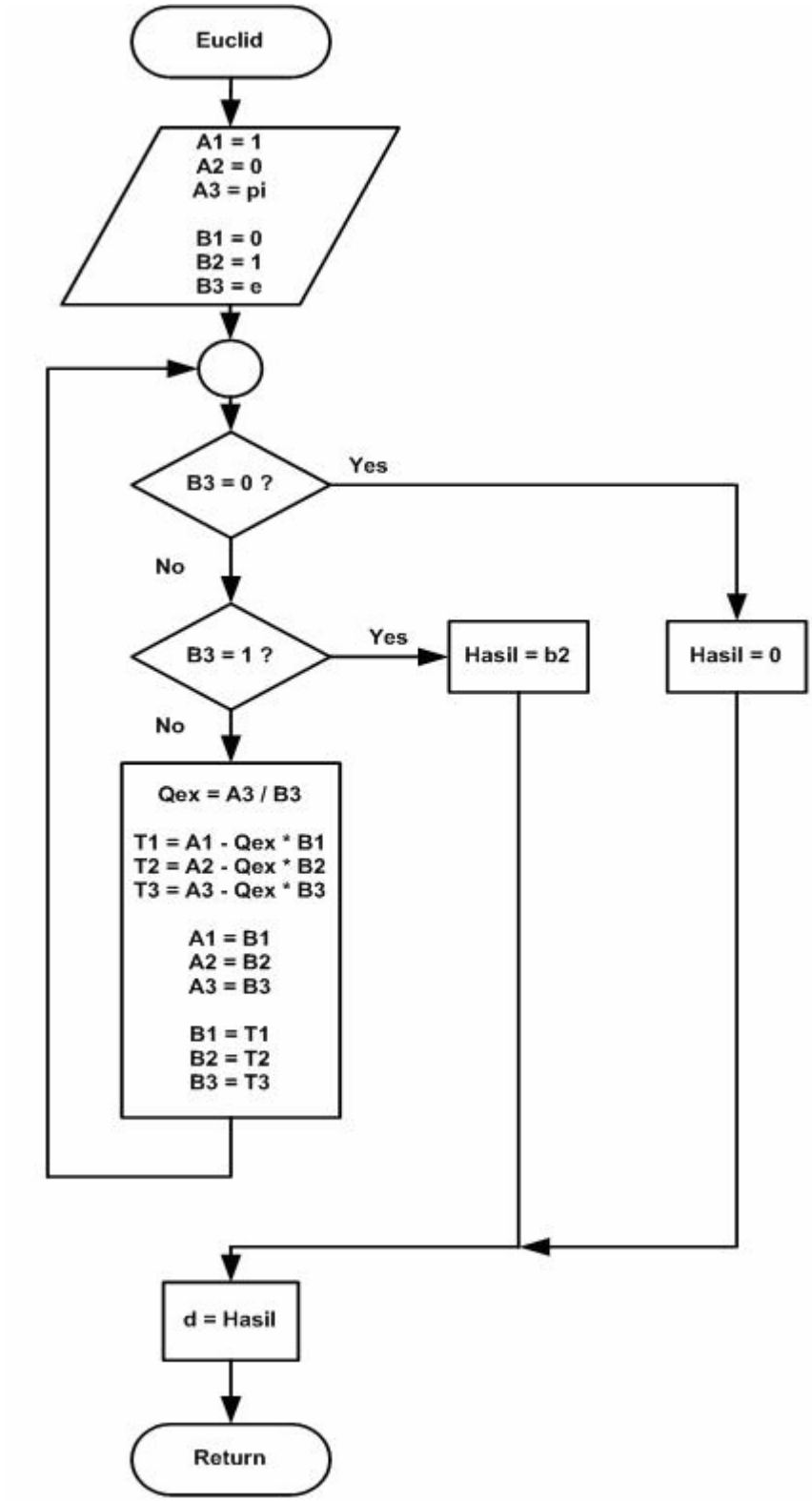


Diagram Alir Sub Program Euclid

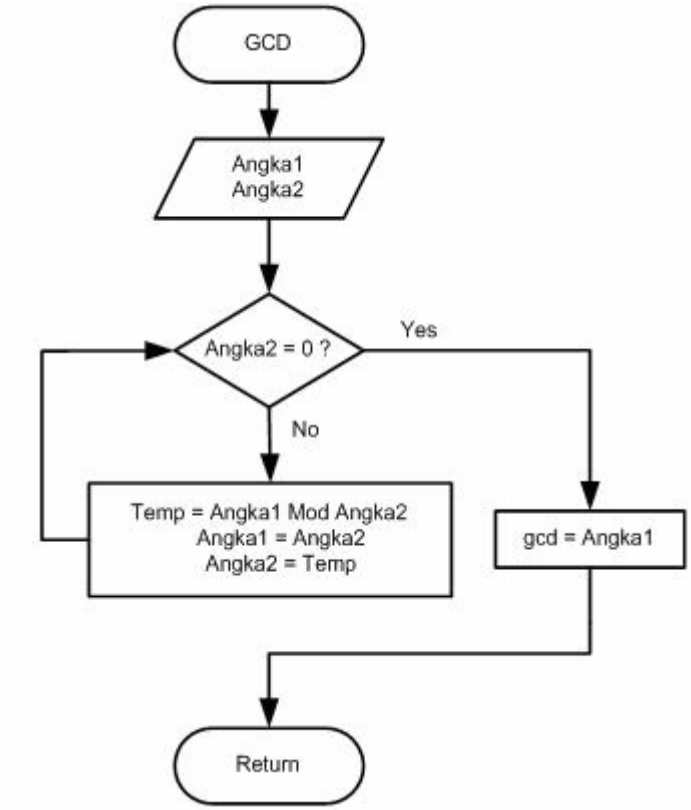


Diagram Alir Sub Program gcd

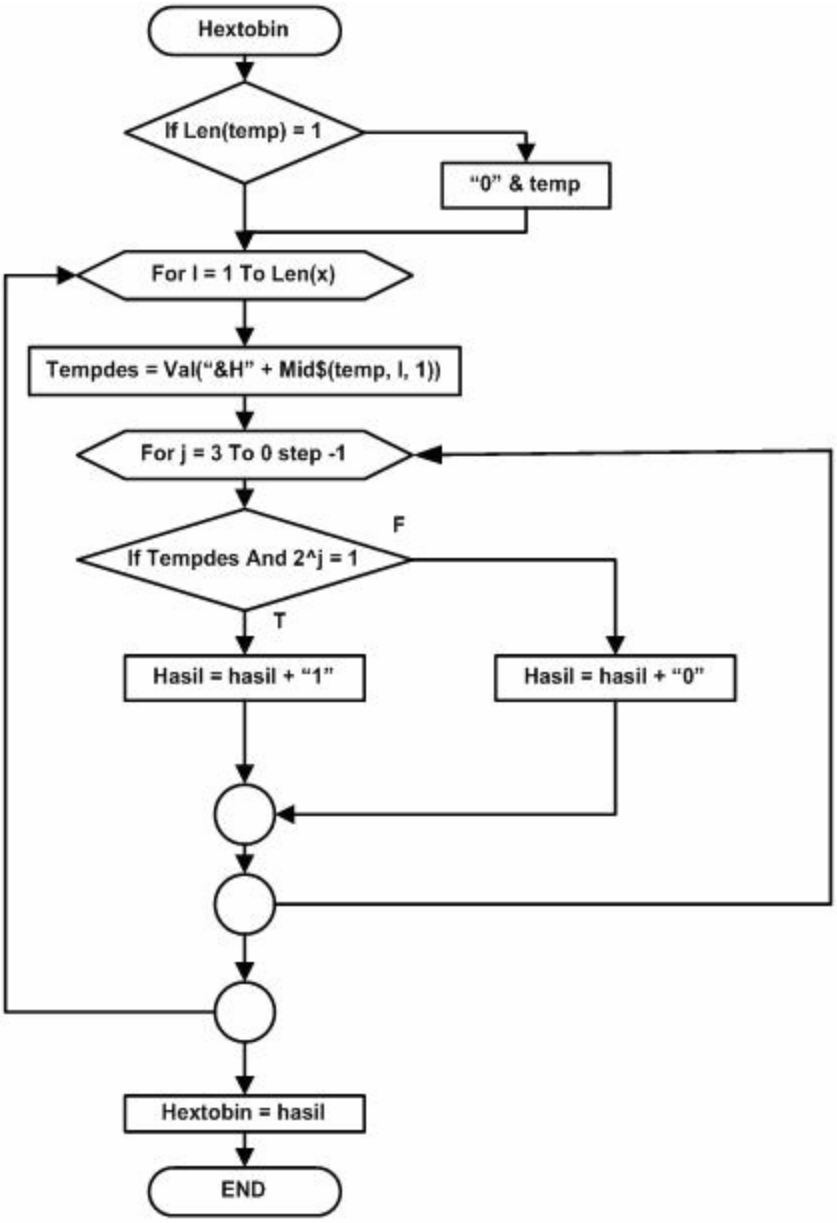
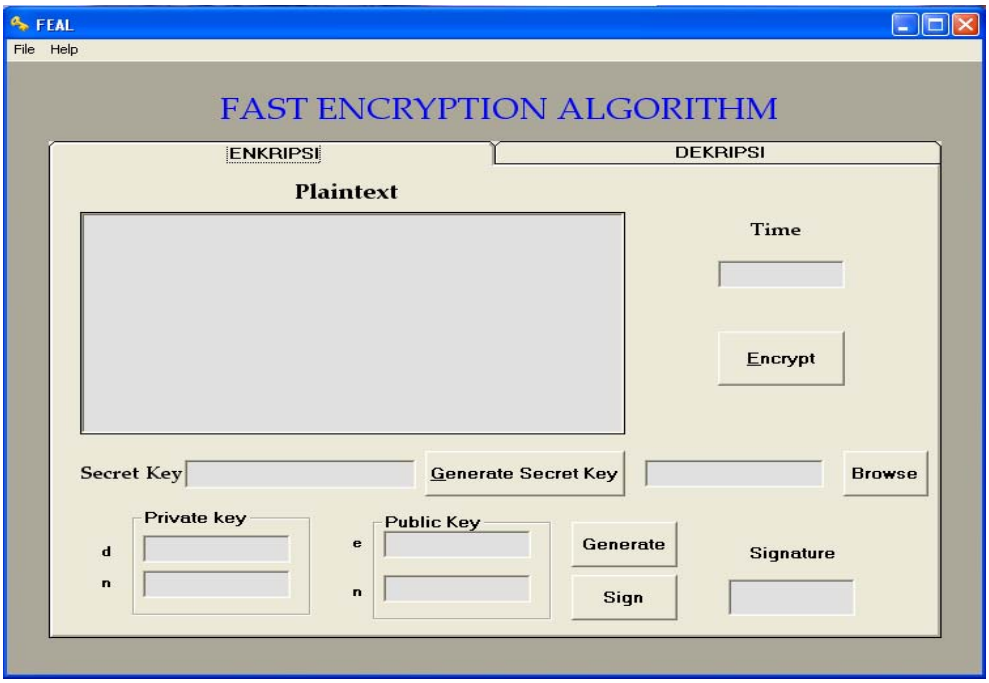
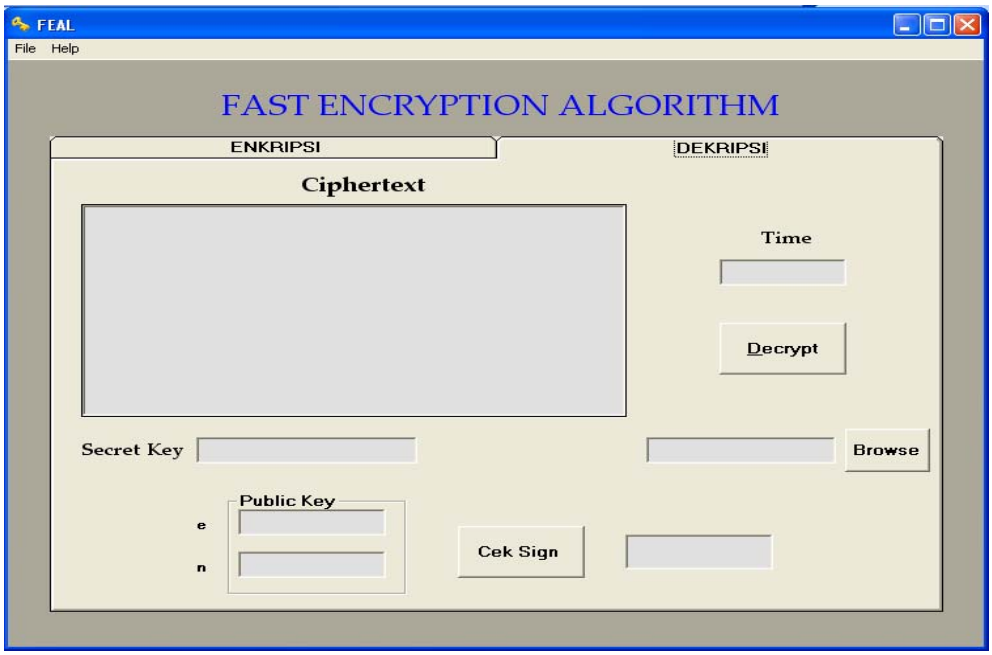


Diagram Alir Fungsi Hextobin

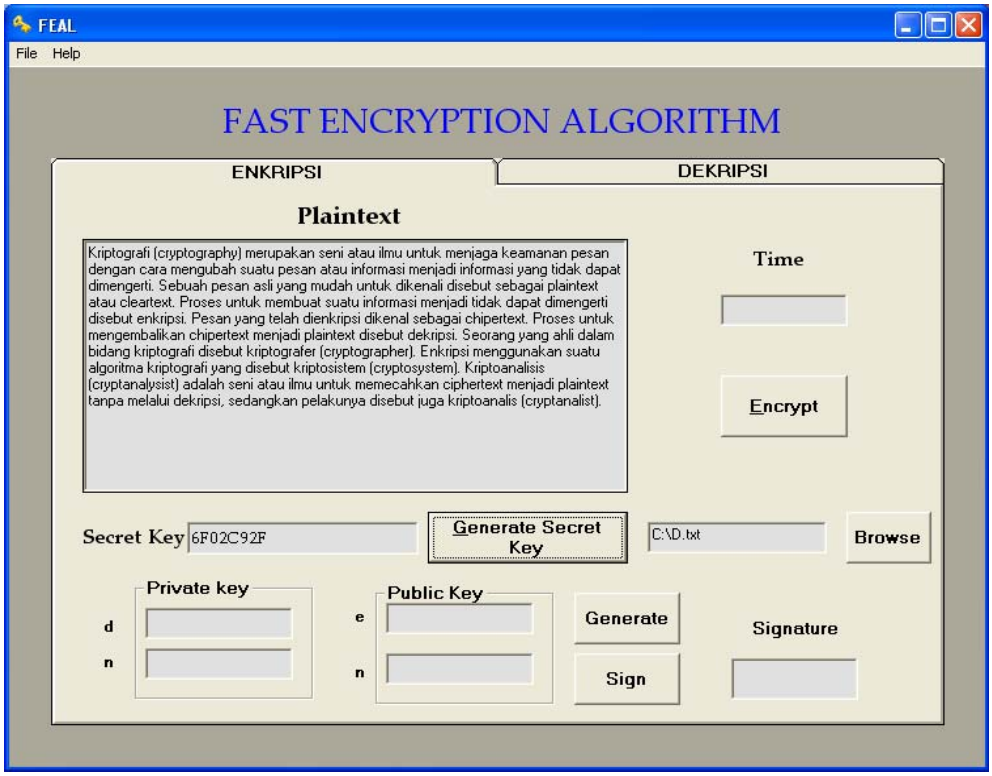
LAMPIRAN C
TAMPILAN PROGRAM



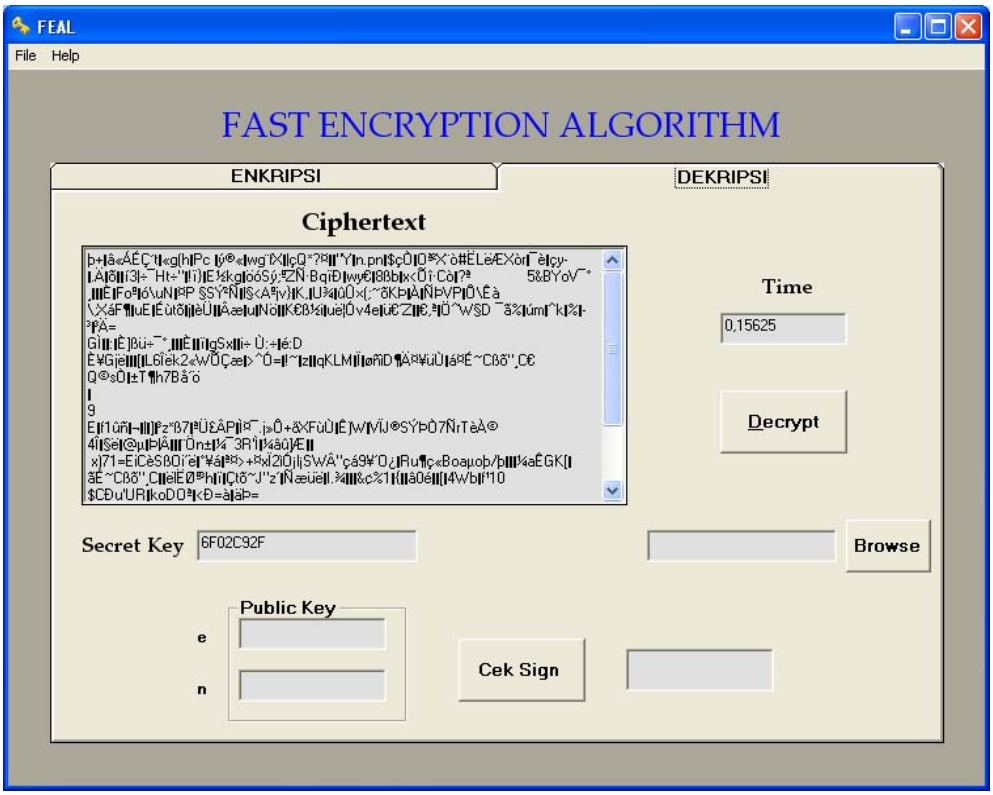
Tampilan Awal Program Enkripsi Pengamanan Data



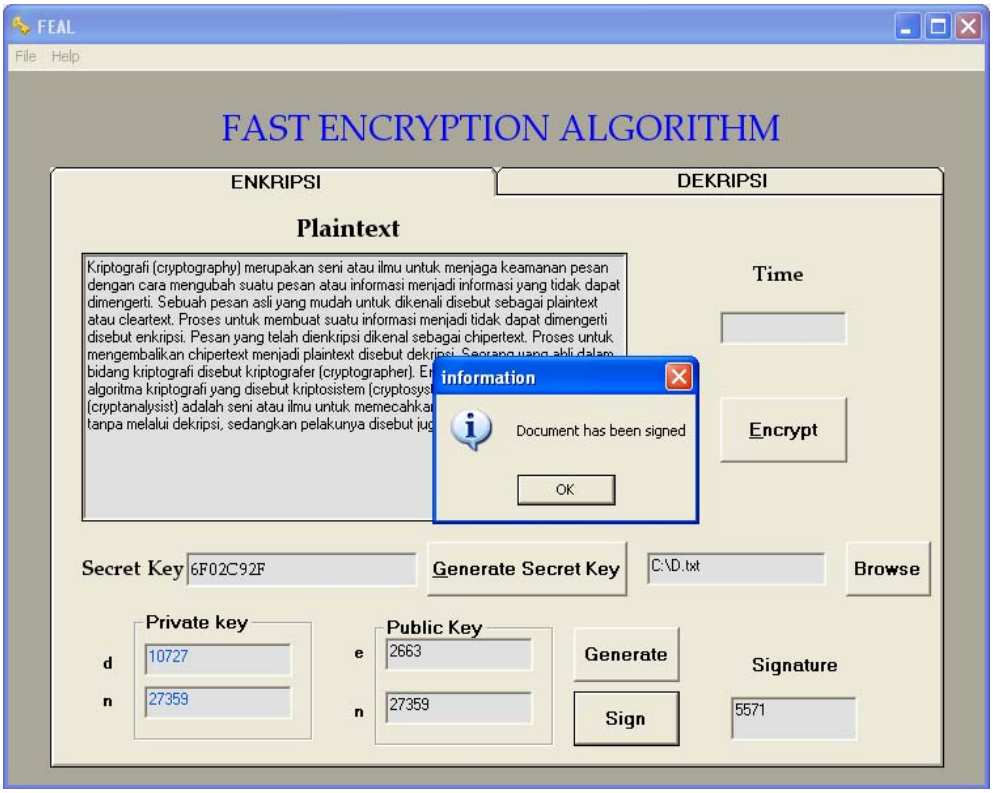
Tampilan Awal Program Dekripsi Pengamanan Data



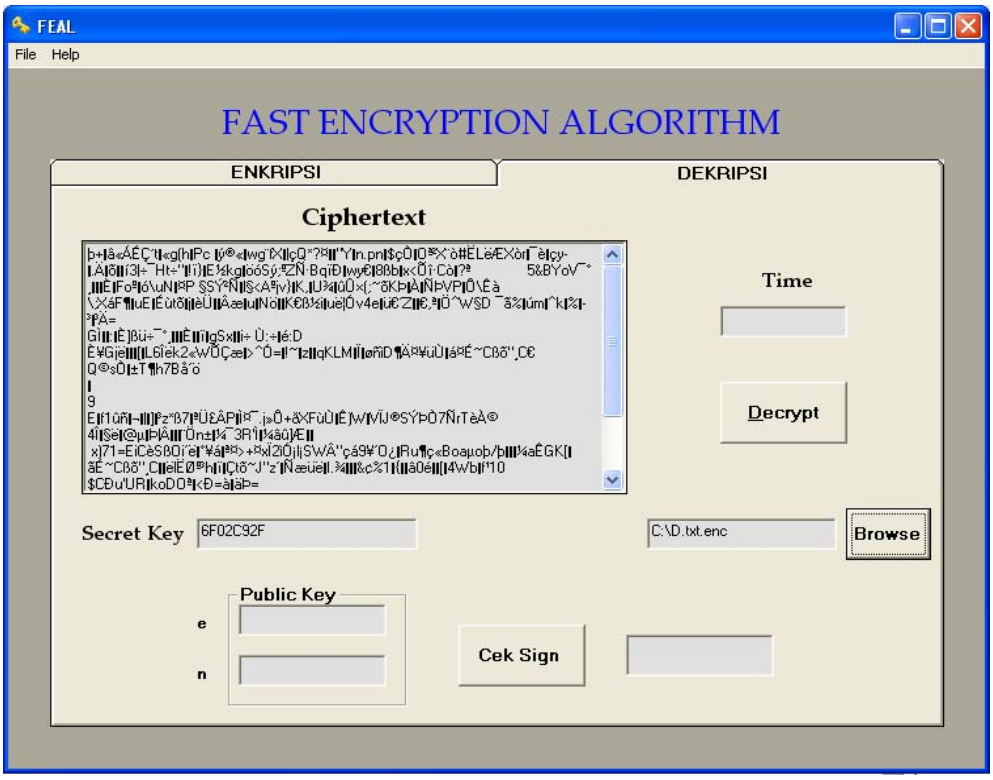
Tampilan Awal Proses Enkripsi



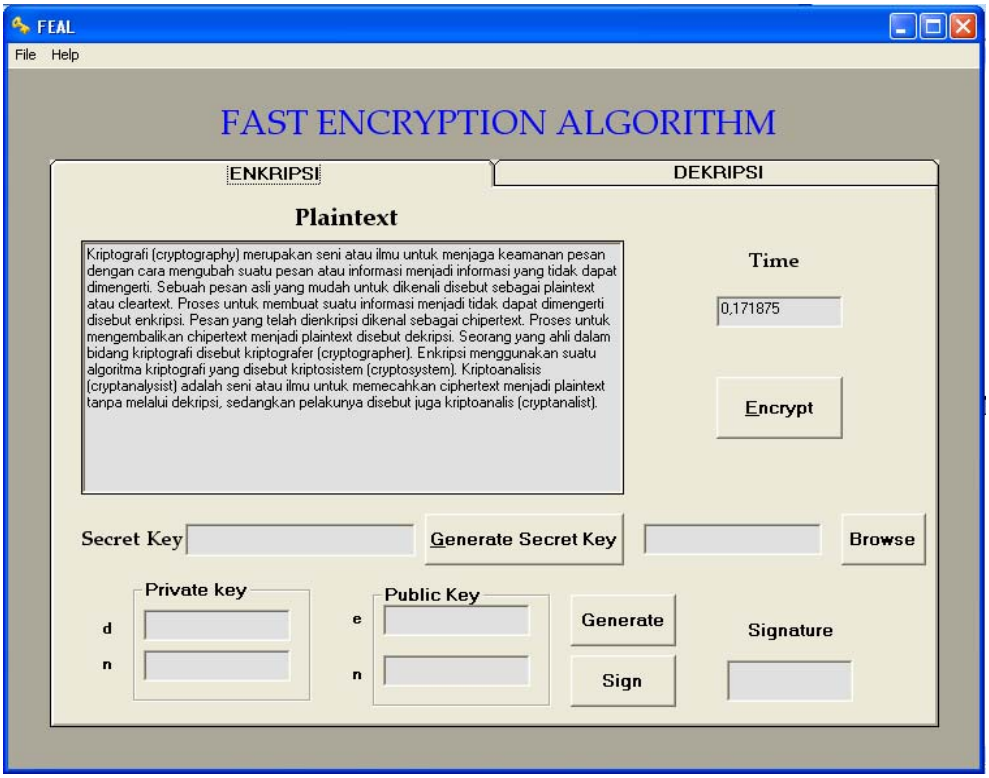
Tampilan Akhir Proses Enkripsi



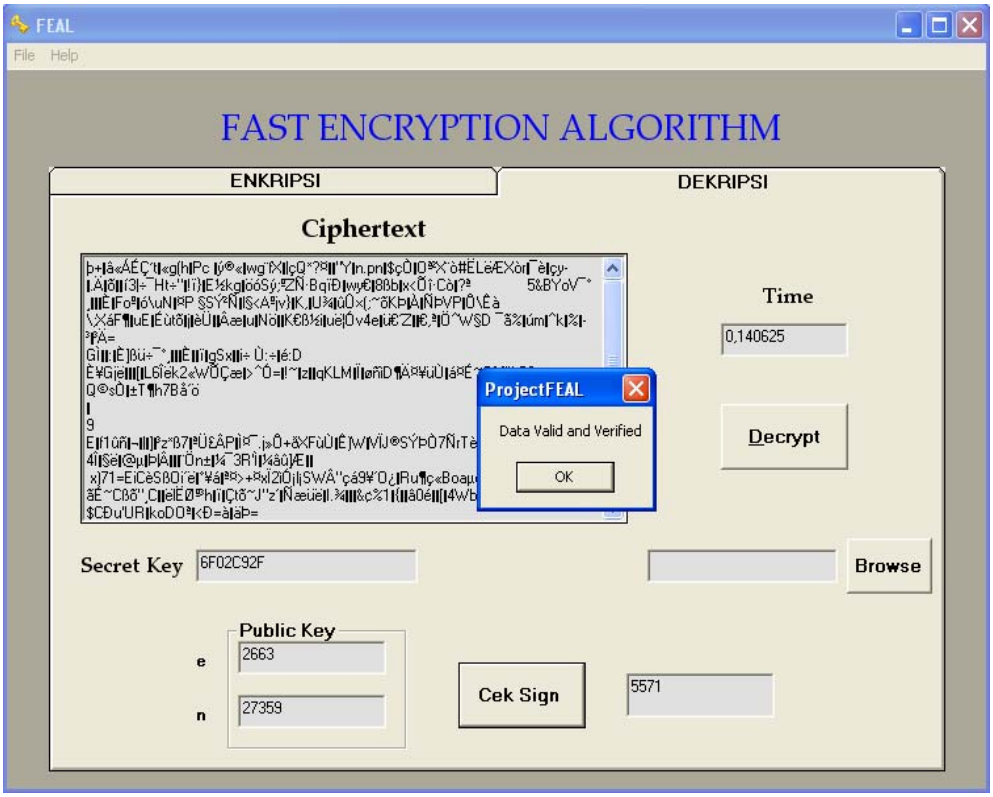
Tampilan Proses Signature



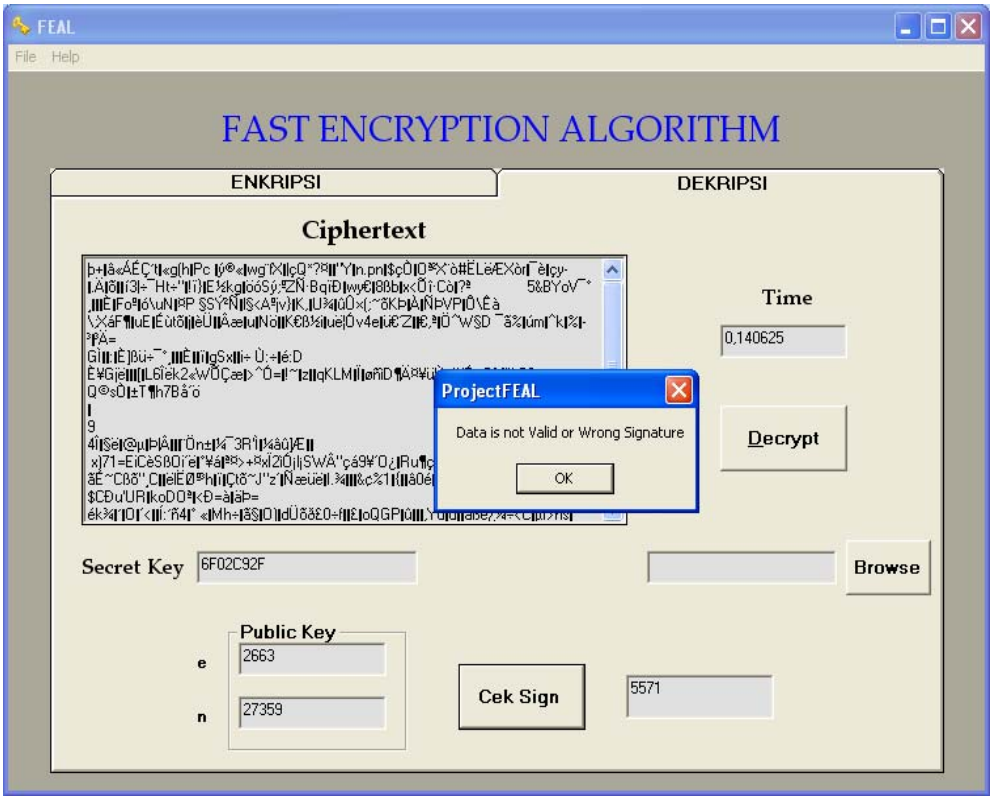
Tampilan Awal Proses Dekripsi



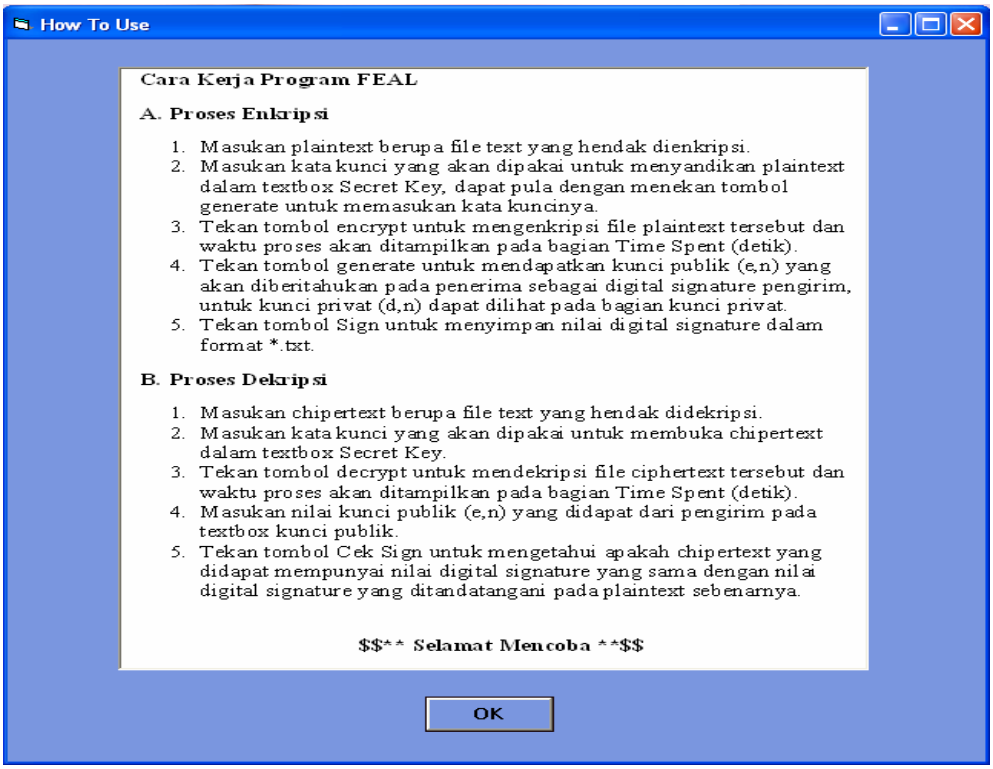
Tampilan Akhir Proses Dekripsi



Tampilan Proses Validasi dan Verifikasi



Tampilan Gagalnya Proses Validasi dan Verifikasi



Tampilan Form Cara Kerja

Tampilan Form About