

LAMPIRAN A
LISTING PROGRAM MENU DAFTAR

Form MenuDaftar

Option Explicit
Dim edit As Boolean

```
Sub validasi(X As Integer)
    Dim i As Integer
    Select Case X
    Case 0: tgl.Enabled = False
        For i = 2 To 9: Text(i).Enabled = False: Option1.Enabled = False: Option2.Enabled = False: status.Enabled = False: Next i
    Case 1: tgl.Enabled = True
        For i = 2 To 9: Text(i).Enabled = True: Option1.Enabled = True: Option2.Enabled = True: status.Enabled = True: Next i
    Case 2: tgl.Value = Now
        For i = 2 To 9: Text(i) = "": Option1.Value = False: Option2.Value = False: status.Text = "": Next i
    End Select
End Sub
```

```
Private Sub Form_Activate()
    tgl.Value = Now
    Dim a, b, c As Integer
    Query "SELECT Kartu.Id_Kartu,Tabungan.No_Rekening FROM Kartu,Tabungan"
    If Not rec.EOF Then
        rec.MoveLast: a = rec!Id_Kartu
        Text(1) = a + 1: validasi (1)
        rec.MoveLast: b = rec!No_Rekening
        Text(0) = b + 1: validasi (1)
        validasi (2): Text(2).SetFocus: tombol(0).Enabled = True
        tombol(1).Enabled = False: tombol(2).Enabled = False: tombol(3).Enabled = False
    End If
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
    MenuDaftar.Show
End Sub
```

```
Private Sub Text_KeyPress(index As Integer, KeyAscii As Integer)
    If (index >= 9) Then inputType Angka, KeyAscii
    If KeyAscii = 39 Then KeyAscii = 0
End Sub
```

```
Private Sub Text_LostFocus(index As Integer)
    If (index >= 9) Then _
    If Len(Trim(Text(index))) = 0 Then Text(index) = "0"
    SendKeys "{home}+{end}"
End Sub
```

```
Private Sub tombol_Click(index As Integer)
    Dim JenisKelamin As String
    If Option1.Value = True Then
        JenisKelamin = "L"
    Else
        JenisKelamin = "P"
    End If
    If index = 0 Then
        If Trim(Text(1)) = "" Or Trim(Text(2)) = "" Or Trim(Text(3)) = "" Or _
        Trim(Text(4)) = "" Or Trim(Text(5)) = "" Or Trim(Text(6)) = "" Or _
        Trim(Text(7)) = "" Or Trim(Text(8)) = "" Or Trim(Text(9)) = "" Or Trim(status.Text) = "" Then
            MsgBox "Data Nasabah belum lengkap!", 16, "Error"
        Else
            If Not edit Then
                con.Execute "INSERT INTO Nasabah VALUES (" & Text(2) & "," & Text(1) & "," & _
                "" & Text(3) & "," & Text(4) & "," & Text(5) & "," & _
                "" & Text(6) & "," & Text(7) & "," & Text(8) & "," & _
                "" & JenisKelamin & "," & status.Text & "," & Format(tgl.Value, "yyyy/mm/dd") & ")"
```

```

Dim bunga, pajak As Integer
bunga = 1000: pajak = 5000
con.Execute "INSERT INTO Tabungan VALUES (" & Text(0) & "," & Text(2) & "," & _
"" & Text(9) & "," & bunga & "," & pajak & ")"

con.Execute "INSERT INTO Kartu VALUES (" & Text(1) & ")"

Else
con.Execute "UPDATE Nasabah SET Id_Nasabah=" & _
"" & Text(2) & ",Nama=" & Text(3) & "," & _
"Alamat=" & Text(4) & ", Tempat_Lahir=" & Text(5) & ",Tanggal_Lahir=" & Text(6) & "," & _
"Telephone=" & Text(7) & ",Pekerjaan=" & Text(8) & ",Jenis_Kelamin=" & JenisKelamin & "," & _
"Status=" & status.Text & ",Tanggal_Daftar=" & Format(tgl.Value, "yyyy/mm/dd") & " WHERE Id_Kartu=" &
Text(1) & ""

con.Execute "UPDATE Tabungan SET Saldo=" & _
"" & Text(9) & ",Bunga=" & bunga & "," & _
"Pajak=" & pajak & " WHERE No_Rekening=" & Text(0) & ""

edit = False
End If
validasi (0): tombol(0).Enabled = False: tombol(1).Enabled = True
tombol(2).Enabled = True: tombol(1).SetFocus
tombol(3).Enabled = True
AET60Main.Text1.Text = Text(1).Text
AET60Main.Text2.Text = Text(0).Text
Load AET60Main: AET60Main.Show: Unload Me
End If
ElseIf index = 1 Then
validasi (1): Text(2).SetFocus: SendKeys "{home}+{end}": edit = True
tombol(0).Enabled = True: tombol(1).Enabled = False: tombol(2).Enabled = False: tombol(3).Enabled = False
ElseIf index = 2 Then
If MsgBox("Anda yakin untuk menghapus data Nasabah ini?", 4 + 32, "Konfirmasi") = 6 Then
con.Execute "DELETE FROM Tabungan WHERE No_Rekening=" & Text(0) & ""
con.Execute "DELETE FROM Nasabah WHERE Id_Nasabah=" & Text(2) & ""
con.Execute "DELETE FROM Kartu WHERE Id_Kartu=" & Text(1) & ""
validasi (1): tombol(0).Enabled = True: tombol(1).Enabled = False
validasi (2): tombol(2).Enabled = False: Text(2).SetFocus: tombol(3).Enabled = False

End If
ElseIf index = 3 Then
Dim a, b As Integer
Query "SELECT Kartu.Id_Kartu,Tabungan.No_Rekening FROM Kartu,Tabungan"
If Not rec.EOF Then
rec.MoveLast: a = rec!Id_Kartu
Text(1) = a + 1: validasi (1)
rec.MoveLast: b = rec!No_Rekening
Text(0) = b + 1: validasi (1)
validasi (2): Text(2).SetFocus: tombol(0).Enabled = True
tombol(1).Enabled = False: tombol(2).Enabled = False: tombol(3).Enabled = False
End If

Else
On Error GoTo a
Dim kode As Double
kode = InputBox("Masukkan No Rekening yang Anda cari:", "Pencarian")
Query "SELECT * FROM Tabungan WHERE No_Rekening = " & kode & ""
If Not rec.EOF Then
Text(2) = rec!Id_Nasabah
Text(9) = rec!Saldo
Query "SELECT * FROM Nasabah WHERE Id_Nasabah = " & Text(2) & ""
If rec.EOF = False Then
Text(1) = rec!Id_Kartu
Text(3) = rec!Nama
Text(4) = rec!Alamat
Text(5) = rec!Tempat_Lahir
Text(6) = rec!Tanggal_Lahir

```

```

Text(7) = rec!Telephone
Text(8) = rec!Pekerjaan
If rec!Jenis_Kelamin = "L" Then
    Option1.Value = True
Else
    Option2.Value = True
End If
status.Text = rec!status
tgl.Value = rec!Tanggal_Daftar
End If

validasi (0): Text(0) = kode
tombol(0).Enabled = False
tombol(1).Enabled = True
tombol(2).Enabled = True
tombol(3).Enabled = True
Else
a: MsgBox "Data yang dicari tidak ditemukan", 16, "Pencarian"
End If
End If
End Sub

```

```

Private Sub tombol1_Click(index As Integer)
Unload Me
End Sub

```

Form AET60Main

```

Option Explicit
Dim scNit, rdrLocked As Boolean
Dim retCode, numDev, numRdr, hContext, hCard, Protocol, RetLength As Long
Dim dwActProtocol, RecvLen, SendLen As Long
Dim ioRequest As SCARD_IO_REQUEST
Dim FpTemplate(0 To 767) As Byte
Dim SendBuff(0 To 262) As Byte
Dim RecvBuff(0 To 262) As Byte
Dim MatchResult As Integer
Dim apdu As APDURec
Dim Transaction As SCARD_IO_REQUEST
Dim rName, scannerName As String
Dim fpTemplateLength, procTimeOut As Long
Dim rList As String * 256
Dim desFar, desFrr, achFar, achFrr As Long
Const MAX_BUFFER_LEN = 256
Const INVALID_SW1SW2 = -450

```

```

Private Sub cmdEnroll_Click()
Dim tmpStr As String
Dim ctr, indx, fileNo, byteNo As Long
Dim tmpArray(0 To 31) As Byte

Call clearMessages
Call clearImage

'Format kartu ACOS2
retCode = SubmitIC()
If retCode <> SCARD_S_SUCCESS Then
Exit Sub
End If

'Pilih FF 02
retCode = selectFile(&HFF, &H2)
If retCode <> SCARD_S_SUCCESS Then
Exit Sub
End If
tmpStr = ""

```

```

For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    Exit Sub
End If

```

```

'Menulis record pertama dari FF 02
tmpArray(0) = &H0 '00 Option registers
tmpArray(1) = &H0 '00 Security option register
tmpArray(2) = &H3 '01 No of user files
tmpArray(4) = &H0 '00 Personalization bit
retCode = writeRecord(0, &H0, &H4, &H4, tmpArray())
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
End If

```

```

'Melakukan reset
retCode = SCardDisconnect(hCard, SCARD_UNPOWER_CARD)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If
scInit = False
retCode = SCardConnect(hContext, _
    cbReader.Text, _
    SCARD_SHARE_SHARED, _
    SCARD_PROTOCOL_T0 Or SCARD_PROTOCOL_T1, _
    hCard, _
    Protocol)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If
scInit = True

```

```

'Mengirim kode IC
retCode = SubmitIC()
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
End If

```

```

'Pilih FF 04
retCode = selectFile(&HFF, &H4)
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    Exit Sub
End If

```

```

'Menulis record pertama dari FF 04
tmpArray(0) = &H20 '32 Panjang record
tmpArray(1) = &H18 '24 24 record
tmpArray(2) = &H0 '00 Atribut baca
tmpArray(3) = &H0 '00 Atribut Tulis
tmpArray(4) = &HAA 'AA Nama file
tmpArray(5) = &HAA 'AA Nama file
retCode = writeRecord(0, &H0, &H6, &H6, tmpArray())
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub

```

End If

```
'Menulis record kedua dari FF 04
tmpArray(0) = &H20 ' 32 Panjang record
tmpArray(1) = &H2 ' 2 2 record
tmpArray(2) = &H0 ' 00 Atribut baca
tmpArray(3) = &H0 ' 00 Atribut Tulis
tmpArray(4) = &HBB ' BB Nama file
tmpArray(5) = &HBB ' BB Nama file
retCode = writeRecord(0, &H1, &H6, &H6, tmpArray())
If retCode <> SCARD_S_SUCCESS Then
  Exit Sub
End If
```

```
'Mengambil data dari fp scanner
procTimeOut = 15000
fpTemplateLength = 768
retCode = AET60_Enroll(FpTemplate(0), _
  fpTemplateLength, _
  procTimeOut)
If retCode <> 0 Then
  Call displayOut(1, retCode, "")
  Exit Sub
End If
Call displayOut(2, 0, "Proses enroll sukses.")
```

```
'Menulis data ke kartu
'Pilih file AA AA
retCode = selectFile(&HAA, &HAA)
If retCode <> SCARD_S_SUCCESS Then
  Exit Sub
End If
tmpStr = ""
For indx = 0 To 1
  tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "91 00 " Then
  Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
  Exit Sub
End If
```

```
'Menulis data fp scanner ke file AA AA
Call displayOut(3, 0, "Tunggu sebentar, data sedang disimpan pada kartu...")
userAction.Refresh
ctr = 0
For fileNo = 0 To 23 ' 24 record
  For byteNo = 0 To 31 ' 32 bytes per record
    tmpArray(byteNo) = FpTemplate(ctr)
    ctr = ctr + 1
  Next byteNo
  retCode = writeRecord(0, fileNo, &H20, &H20, tmpArray())
  If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
  End If
Next fileNo
```

```
'Pilih file BB BB
retCode = selectFile(&HBB, &HBB)
If retCode <> SCARD_S_SUCCESS Then
  Exit Sub
End If
tmpStr = ""
For indx = 0 To 1
  tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "91 01 " Then
  Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
```

```
Exit Sub
End If

'Menulis Id Kartu dan No Rekening ke File BB BB
tmpStr = Text1.Text
For indx = 0 To Len(tmpStr) - 1
    tmpArray(indx) = Asc(Mid(tmpStr, indx + 1, 1))
Next indx
retCode = writeRecord(1, &H0, &H20, Len(tmpStr), tmpArray())
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
End If

tmpStr = Text2.Text
For indx = 0 To Len(tmpStr) - 1
    tmpArray(indx) = Asc(Mid(tmpStr, indx + 1, 1))
Next indx
retCode = writeRecord(1, &H1, &H20, Len(tmpStr), tmpArray())
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
End If

Call displayOut(3, 0, "")
Call displayOut(2, 0, "Proses Penyimpanan Sidik Jari sukses..!")
End Sub
```

```
Private Sub cmdQuit_Click()
Unload Me: Load MenuDaftar: MenuDaftar.Show
Module1.Main
End Sub
```

```
Private Sub cmdReset_Click()
Call ResetInitConnect
End Sub
```

```
Private Sub ResetInitConnect()
'Menon-aktifkan FP scanner
If rdrLocked Then
    retCode = AET60_UnlockReader()
End If
retCode = AET60_Close()

'Menon-aktifkan SC reader
If scInit Then
    retCode = SCardDisconnect(hCard, SCARD_UNPOWER_CARD)
End If
retCode = SCardReleaseContext(hContext)
scInit = False
rdrLocked = False

Dim indx As Long
rList = String(255, vbNullChar)
numRdr = 255

'Inisialisasi SC Reader
retCode = SCardEstablishContext(SCARD_SCOPE_USER, 0, 0, hContext)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If

retCode = SCardListReaders(hContext, rName, rList, numRdr)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
```

```

End If
Call LoadListToControl(cbReader, rList)
cbReader.ListIndex = 0

'Inialisasi FP Scanner
retCode = AET60_Open()
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If
numDev = AET60_GetNumDevices()
If numDev > 0 Then
    cbScanner.Clear
    scannerName = String(255, vbNullChar)
    RetLength = Len(scannerName)
    For indx = 0 To (numDev - 1)
        retCode = AET60_GetDeviceName(indx, _
            scannerName, _
            RetLength)
        cbScanner.AddItem scannerName
        cbScanner.ListIndex = 0
    Next indx
Else
    Call displayOut(1, retCode, "")
    Exit Sub
End If

    Call displayOut(2, 0, "")
Call displayOut(3, 0, "")

' 1. Connect to smartcard
retCode = SCardConnect(hContext, _
    cbReader.Text, _
    SCARD_SHARE_SHARED, _
    SCARD_PROTOCOL_T0 Or SCARD_PROTOCOL_T1, _
    hCard, _
    Protocol)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If
scInit = True

' 2. Lock scanner and set callback
retCode = AET60_LockReader(cbScanner.ListIndex)
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If
rdrLocked = True

retCode = AET60_SetCallback(AddressOf CALLBACK_FUNCTION)
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If

Call displayOut(3, 0, "Silahkan simpan Sidik Jari Nasabah")
Call displayOut(2, 0, "Smart card telah terhubung.")
cbReader.Enabled = False
cbScanner.Enabled = False
End Sub

```

```

Private Sub Form_Activate()
Call ResetInitConnect
End Sub

```

```
Private Sub clearImage()
    Dim dib As BITMAPINFO_256
    Dim tmpArray(0 To 91259) As Byte
    Dim indx As Long

    For indx = 0 To 91259
        tmpArray(indx) = &H0
    Next indx

    dib.bmiHeader.biSize = Len(dib.bmiHeader)
    dib.bmiHeader.biWidth = 256
    dib.bmiHeader.biHeight = 360
    dib.bmiHeader.biPlanes = 1
    dib.bmiHeader.biBitCount = 8
    dib.bmiHeader.biCompression = BI_RGB
    dib.bmiHeader.biSizeImage = 0
    dib.bmiHeader.biXPelsPerMeter = 0
    dib.bmiHeader.biYPelsPerMeter = 0
    dib.bmiHeader.biClrUsed = 0
    dib.bmiHeader.biClrImportant = 0
    dib.bmiColors(0).rgbBlue = 0
    dib.bmiColors(0).rgbGreen = 64
    dib.bmiColors(0).rgbRed = 128
    dib.bmiColors(0).rgbReserved = 0

    Call StretchDIBits(AET60Main.fpImage.hdc, _
        0, 0, _
        AET60Main.fpImage.Width / 16, AET60Main.fpImage.Height / 16, _
        0, 0, _
        256, 360, _
        tmpArray(0), _
        dib, _
        DIB_RGB_COLORS, _
        SRCCOPY)
    AET60Main.fpImage.Refresh
End Sub
```

```
Private Sub clearMessages()
    Call displayOut(2, 0, "")
    Call displayOut(3, 0, "")
    userAction.Refresh
    MsgBox.Refresh
End Sub
```

```
Private Sub ClearBuffers()
    Dim indx As Long
    For indx = 0 To 262
        SendBuff(indx) = &H0
        RecvBuff(indx) = &H0
    Next indx
End Sub
```

```
Private Sub displayOut(ByVal errType As Integer, ByVal retVal As Long, ByVal msgText As String)
    Select Case (errType)
        Case 0 ' Smartcard error
            userAction.Text = "Error encountered. Please click on Reset button."
            MsgBox.Text = GetScardErrMsg(retVal)
        Case 1 ' BioAPI error
            Dim StatusMsg As String
            StatusMsg = String(255, vbNullChar)
            userAction.Text = "Error encountered. Please click on Reset button."
            Call AET60_GetStatus(retVal, StatusMsg)
            MsgBox.Text = StatusMsg
        Case 2 ' Message only
            MsgBox.Text = msgText
        Case 3 ' Required user action
```

```
    userAction.Text = msgText
End Select
End Sub
```

Private Function validIDCard() As Boolean

```
Dim tmpStr As String
Dim indx As Integer
```

```
validIDCard = False
```

```
' 1. Select CC CC
```

```
retCode = selectFile(&HCC, &HCC)
```

```
If retCode <> SCARD_S_SUCCESS Then
```

```
    Exit Function
```

```
End If
```

```
tmpStr = ""
```

```
For indx = 0 To 1
```

```
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
```

```
Next indx
```

```
If tmpStr <> "91 02 " Then
```

```
    Call displayOut(2, 0, "Inserted ID card may be invalid. Return value: " + tmpStr)
```

```
    Exit Function
```

```
End If
```

```
' 2. Read ID Flag
```

```
retCode = readRecord(&H0, &H6)
```

```
If retCode <> SCARD_S_SUCCESS Then
```

```
    Exit Function
```

```
End If
```

```
tmpStr = ""
```

```
For indx = 0 To 3
```

```
    tmpStr = tmpStr + Chr(RecvBuff(indx + 2))
```

```
Next indx
```

```
If tmpStr <> "KNEF" Then
```

```
    Call displayOut(2, 0, "The user is not using a valid ID card.")
```

```
    Exit Function
```

```
End If
```

```
validIDCard = True
```

```
End Function
```

Private Function SubmitIC() As Long

```
Dim indx As Integer
```

```
Dim tmpStr As String
```

```
Call ClearBuffers
```

```
SendBuff(0) = &H80 ' CLA
```

```
SendBuff(1) = &H20 ' INS
```

```
SendBuff(2) = &H7 ' P1
```

```
SendBuff(3) = &H0 ' P2
```

```
SendBuff(4) = &H8 ' P3
```

```
SendBuff(5) = &H41 ' A
```

```
SendBuff(6) = &H43 ' C
```

```
SendBuff(7) = &H4F ' O
```

```
SendBuff(8) = &H53 ' S
```

```
SendBuff(9) = &H54 ' T
```

```
SendBuff(10) = &H45 ' E
```

```
SendBuff(11) = &H53 ' S
```

```
SendBuff(12) = &H54 ' T
```

```
SendLen = &HD
```

```
RecvLen = &H2
```

```
retCode = SendAPDU()
```

```
If retCode <> SCARD_S_SUCCESS Then
```

```
    Call displayOut(0, retCode, "")
```

```
    SubmitIC = retCode
```

```
    Exit Function
```

```
End If
```

```
tmpStr = ""
```

```
For indx = 0 To 1
```

```
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
```

```

Next indx
If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    retCode = INVALID_SW1SW2
End If
SubmitIC = retCode
End Function

```

```

Private Function selectFile(ByVal HiAddr As Byte, ByVal LoAddr As Byte) As Long
    Call ClearBuffers
    SendBuff(0) = &H80 ' CLA
    SendBuff(1) = &HA4 ' INS
    SendBuff(2) = &H0 ' P1
    SendBuff(3) = &H0 ' P2
    SendBuff(4) = &H2 ' P3
    SendBuff(5) = HiAddr ' Value of High Byte
    SendBuff(6) = LoAddr ' Value of Low Byte
    SendLen = &H7
    RecvLen = &H2
    retCode = SendAPDU()
    If retCode <> SCARD_S_SUCCESS Then
        Call displayOut(0, retCode, "")
    End If
    selectFile = retCode
End Function

```

```

Private Function readRecord(ByVal RecNo As Byte, ByVal DataLen As Byte) As Long
    Dim indx As Integer
    Dim tmpStr As String

    Call ClearBuffers
    SendBuff(0) = &H80 ' CLA
    SendBuff(1) = &HB2 ' INS
    SendBuff(2) = RecNo ' P1 Record No
    SendBuff(3) = &H0 ' P2
    SendBuff(4) = DataLen ' P3 Length of data to be read
    SendLen = &H5
    RecvLen = SendBuff(4) + 2
    retCode = SendAPDU()
    If retCode <> SCARD_S_SUCCESS Then
        Call displayOut(0, retCode, "")
        readRecord = retCode
        Exit Function
    End If
    tmpStr = ""
    For indx = 0 To 1
        tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx + SendBuff(4))), 2) + " "
    Next indx
    If tmpStr <> "90 00 " Then
        Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
        retCode = INVALID_SW1SW2
    End If
    readRecord = retCode
End Function

```

```

Private Function writeRecord(ByVal caseType As Integer, ByVal RecNo As Byte, _
    ByVal maxDataLen As Byte, ByVal DataLen As Byte, _
    ByRef DataIn() As Byte) As Long
    Dim tmpStr As String
    Dim indx As Integer

    If caseType = 1 Then
        ' 1. Re-initialize card values to $00
        Call ClearBuffers
        SendBuff(0) = &H80 ' CLA
        SendBuff(1) = &HD2 ' INS
    End If

```

```

SendBuff(2) = RecNo      ' P1  Record to be written
SendBuff(3) = &H0      ' P2
SendBuff(4) = maxDataLen  ' P3  Length
For indx = 0 To maxDataLen - 1
  SendBuff(indx + 5) = &H0
Next indx
SendLen = maxDataLen + 5
RecvLen = &H2
retCode = SendAPDU()
If retCode <> SCARD_S_SUCCESS Then
  Call displayOut(0, retCode, "")
  writeRecord = retCode
  Exit Function
End If
tmpStr = ""
For indx = 0 To 1
  tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "90 00 " Then
  Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
  retCode = INVALID_SW1SW2
  writeRecord = retCode
  Exit Function
End If
End If

```

```

' 2. Write data to card
  Call ClearBuffers
  SendBuff(0) = &H80      ' CLA
  SendBuff(1) = &HD2      ' INS
  SendBuff(2) = RecNo      ' P1  Record to be written
  SendBuff(3) = &H0      ' P2
  SendBuff(4) = DataLen  ' P3  Length
  For indx = 0 To DataLen - 1
    SendBuff(indx + 5) = DataIn(indx)
  Next indx
  SendLen = DataLen + 5
  RecvLen = &H2
  retCode = SendAPDU()
  If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    writeRecord = retCode
    Exit Function
  End If
  tmpStr = ""
  For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
  Next indx
  If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    retCode = INVALID_SW1SW2
  End If
  writeRecord = retCode
End Function

```

Private Function SendAPDU() As Long
 ioRequest.dwProtocol = dwActProtocol
 ioRequest.cbPciLength = Len(ioRequest)

```

retCode = SCardTransmit(hCard, _
  ioRequest, _
  SendBuff(0), _
  SendLen, _
  ioRequest, _
  RecvBuff(0), _
  RecvLen)

```

SendAPDU = retCode

End Function

```
Public Function GetScardErrMsg(ByVal ReturnCode As Long) As String
Select Case ReturnCode
'Smartcard Reader Errors
Case SCARD_E_CANCELLED
GetScardErrMsg = "The action was canceled by an SCardCancel request."
Case SCARD_E_CANT_DISPOSE
GetScardErrMsg = "The system could not dispose of the media in the requested manner."
Case SCARD_E_CARD_UNSUPPORTED
GetScardErrMsg = "The smart card does not meet minimal requirements for support."
Case SCARD_E_DUPLICATE_READER
GetScardErrMsg = "The reader driver didn't produce a unique reader name."
Case SCARD_E_INSUFFICIENT_BUFFER
GetScardErrMsg = "The data buffer for returned data is too small for the returned data."
Case SCARD_E_INVALID_ATR
GetScardErrMsg = "An ATR string obtained from the registry is not a valid ATR string."
Case SCARD_E_INVALID_HANDLE
GetScardErrMsg = "The supplied handle was invalid."
Case SCARD_E_INVALID_PARAMETER
GetScardErrMsg = "One or more of the supplied parameters could not be properly interpreted."
Case SCARD_E_INVALID_TARGET
GetScardErrMsg = "Registry startup information is missing or invalid."
Case SCARD_E_INVALID_VALUE
GetScardErrMsg = "One or more of the supplied parameter values could not be properly interpreted."
Case SCARD_E_NOT_READY
GetScardErrMsg = "The reader or card is not ready to accept commands."
Case SCARD_E_NOT_TRANSACTED
GetScardErrMsg = "An attempt was made to end a non-existent transaction."
Case SCARD_E_NO_MEMORY
GetScardErrMsg = "Not enough memory available to complete this command."
Case SCARD_E_NO_SERVICE
GetScardErrMsg = "The smart card resource manager is not running."
Case SCARD_E_NO_SMARTCARD
GetScardErrMsg = "The operation requires a smart card, but no smart card is currently in the device."
Case SCARD_E_PCI_TOO_SMALL
GetScardErrMsg = "The PCI receive buffer was too small."
Case SCARD_E_PROTO_MISMATCH
GetScardErrMsg = "The requested protocols are incompatible with the protocol currently in use with the card."
Case SCARD_E_READER_UNAVAILABLE
GetScardErrMsg = "The specified reader is not currently available for use."
Case SCARD_E_READER_UNSUPPORTED
GetScardErrMsg = "The reader driver does not meet minimal requirements for support."
Case SCARD_E_SERVICE_STOPPED
GetScardErrMsg = "The smart card resource manager has shut down."
Case SCARD_E_SHARING_VIOLATION
GetScardErrMsg = "The smart card cannot be accessed because of other outstanding connections."
Case SCARD_E_SYSTEM_CANCELLED
GetScardErrMsg = "The action was canceled by the system, presumably to log off or shut down."
Case SCARD_E_TIMEOUT
GetScardErrMsg = "The user-specified timeout value has expired."
Case SCARD_E_UNKNOWN_CARD
GetScardErrMsg = "The specified smart card name is not recognized."
Case SCARD_E_UNKNOWN_READER
GetScardErrMsg = "The specified reader name is not recognized."
Case SCARD_F_COMM_ERROR
GetScardErrMsg = "An internal communications error has been detected."
Case SCARD_F_INTERNAL_ERROR
GetScardErrMsg = "An internal consistency check failed."
Case SCARD_F_UNKNOWN_ERROR
GetScardErrMsg = "An internal error has been detected, but the source is unknown."
Case SCARD_F_WAITED_TOO_LONG
GetScardErrMsg = "An internal consistency timer has expired."
Case SCARD_S_SUCCESS
GetScardErrMsg = "No error was encountered."
Case SCARD_W_REMOVED_CARD
GetScardErrMsg = "The smart card has been removed, so that further communication is not possible."
```

```
Case SCARD_W_RESET_CARD
GetScardErrMsg = "The smart card has been reset, so any shared state information is invalid."
Case SCARD_W_UNPOWERED_CARD
GetScardErrMsg = "Power has been removed from the smart card, so that further communication is not possible."
Case SCARD_W_UNRESPONSIVE_CARD
GetScardErrMsg = "The smart card is not responding to a reset."
Case SCARD_W_UNSUPPORTED_CARD
GetScardErrMsg = "The reader cannot communicate with the card, due to ATR string configuration conflicts."
Case Else
GetScardErrMsg = "?"
End Select
End Function
```

LAMPIRAN B
LISTING PROGRAM MENU ATM

Form AET60Main

```
Option Explicit
Dim scInit, rdrLocked As Boolean
Dim retCode, numDev, numRdr, hContext, hCard, Protocol, RetLength As Long
Dim dwActProtocol, RecvLen, SendLen As Long
Dim ioRequest As SCARD_IO_REQUEST
Dim FpTemplate(0 To 767) As Byte
Dim SendBuff(0 To 262) As Byte
Dim RecvBuff(0 To 262) As Byte
Dim MatchResult As Integer
Dim apdu As APDURec
Dim Transaction As SCARD_IO_REQUEST
Dim rName, scannerName As String
Dim fpTemplateLength, procTimeOut As Long
Dim rList As String * 256
Dim desFar, desFrr, achFar, achFrr As Long
Const MAX_BUFFER_LEN = 256
Const INVALID_SW1SW2 = -450
```

```
Private Sub Form_Activate()
Call ResetInitConnect
End Sub
```

```
Private Sub clearImage()
Dim dib As BITMAPINFO_256
Dim tmpArray(0 To 91259) As Byte
Dim indx As Long

For indx = 0 To 91259
tmpArray(indx) = &H0
Next indx

dib.bmiHeader.biSize = Len(dib.bmiHeader)
dib.bmiHeader.biWidth = 256
dib.bmiHeader.biHeight = 360
dib.bmiHeader.biPlanes = 1
dib.bmiHeader.biBitCount = 8
dib.bmiHeader.biCompression = BI_RGB
dib.bmiHeader.biSizeImage = 0
dib.bmiHeader.biXPelsPerMeter = 0
dib.bmiHeader.biYPelsPerMeter = 0
dib.bmiHeader.biClrUsed = 0
dib.bmiHeader.biClrImportant = 0
dib.bmiColors(0).rgbBlue = 0
dib.bmiColors(0).rgbGreen = 64
dib.bmiColors(0).rgbRed = 128
dib.bmiColors(0).rgbReserved = 0

Call StretchDIBits(AET60Main.fpImage.hDC, _
0, 0, _
AET60Main.fpImage.Width / 16, AET60Main.fpImage.Height / 16, _
0, 0, _
256, 360, _
tmpArray(0), _
dib, _
DIB_RGB_COLORS, _
SRCCOPY)
AET60Main.fpImage.Refresh
End Sub
```

```
Private Sub ResetInitConnect()
'Menon-aktifkan FP scanner
If rdrLocked Then
retCode = AET60_UnlockReader()
```



```

End If
retCode = AET60_Close()

'Menon-aktifkan SC reader
If scInit Then
    retCode = SCardDisconnect(hCard, SCARD_UNPOWER_CARD)
End If
retCode = SCardReleaseContext(hContext)
scInit = False
rdrLocked = False

Dim indx As Long
rList = String(255, vbNullChar)
numRdr = 255

'Inisialisasi SC Reader
retCode = SCardEstablishContext(SCARD_SCOPE_USER, 0, 0, hContext)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If

retCode = SCardListReaders(hContext, rName, rList, numRdr)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If
Call LoadListToControl(cbReader, rList)
cbReader.ListIndex = 0

'Inisialisasi FP Scanner
retCode = AET60_Open()
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If
numDev = AET60_GetNumDevices()
If numDev > 0 Then
    cbScanner.Clear
    scannerName = String(255, vbNullChar)
    RetLength = Len(scannerName)
    For indx = 0 To (numDev - 1)
        retCode = AET60_GetDeviceName(indx, _
            scannerName, _
            RetLength)
        cbScanner.AddItem scannerName
    Next indx
    cbScanner.ListIndex = 0
Else
    Call displayOut(1, retCode, "")
    Exit Sub
End If

Call displayOut(2, 0, "")
Call displayOut(3, 0, "")

' 1. Connect to smartcard
retCode = SCardConnect(hContext, _
    cbReader.Text, _
    SCARD_SHARE_SHARED, _
    SCARD_PROTOCOL_T0 Or SCARD_PROTOCOL_T1, _
    hCard, _
    Protocol)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If

```

```

scInit = True

' 2. Lock scanner and set callback
retCode = AET60_LockReader(cbScanner.ListIndex)
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If
rdrLocked = True

retCode = AET60_SetCallback(AddressOf CALLBACK_FUNCTION)
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If

'Call displayOut(3, 0, "Silahkan simpan Sidik Jari Nasabah")
'Call displayOut(2, 0, "Smart card telah terhubung.")
cbReader.Enabled = False
cbScanner.Enabled = False
End Sub

```

```

Private Sub verify()
Dim tmpStr As String
Dim ctr, indx, fileNo, byteNo As Long
Dim tmpTemplate(0 To 767) As Byte

Call clearMessages
Call clearImage

'Melakukan re-connect untuk menghilangkan error pada kasus kartu belum dimasukkan
retCode = SCARDDisconnect(hCard, SCARD_UNPOWER_CARD)
scInit = False
retCode = SCardConnect(hContext, _
    cbReader.Text, _
    SCARD_SHARE_SHARED, _
    SCARD_PROTOCOL_T0 Or SCARD_PROTOCOL_T1, _
    hCard, _
    Protocol)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Call displayOut(3, 0, "Insert your ID card into the reader.")
    Exit Sub
End If
scInit = True

'Membaca data dari kartu
'Pilih file AA AA
retCode = selectFile(&HAA, &HAA)
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "91 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    Exit Sub
End If

'Membaca data sidik jari dari kartu
Call displayOut(3, 0, "Please wait while fingerprint data is read from card.")
userAction.Refresh
ctr = 0
For fileNo = 0 To 23 ' Loop through record no
    retCode = readRecord(fileNo, &H20)

```

```

If retCode <> SCARD_S_SUCCESS Then
  Exit Sub
End If
For byteNo = 0 To 31      ' Loop through 32 bytes per record
  tmpTemplate(ctr) = RecvBuff(byteNo)
  ctr = ctr + 1
Next byteNo
Next fileNo

```

'Mengambil data sidik jari (real) dan mengatur nilai FAR dan FRR
 Select Case secLevel.Value

```

Case 0
  desFar = 4083440
  desFrr = 346819
Case 1
  desFar = 2515777
  desFrr = 358630
Case 2
  desFar = 500364
  desFrr = 464931
Case 3
  desFar = 310419
  desFrr = 500364
Case 4
  desFar = 2148
  desFrr = 654983
End Select

```

'Membaca data dari fp scanner

```

procTimeOut = 15000
If procTimeOut = 0 Then
  Call ResetInitConnect
End If
fpTemplateLength = 768
MatchResult = 0
retCode = AET60_Verify(tmpTemplate(0), _
  fpTemplateLength, _
  procTimeOut, _
  MatchResult, _
  desFar, _
  desFrr, _
  achFar, _
  achFrr)

```

'Membandingkan data sidik jari dari kartu dengan data sidik jari dari fp scanner

```

If MatchResult = 1 Then
  Call displayOut(2, 0, "Sidik Jari Anda Cocok.")
  Load Menu: Menu.Show: Unload Me
Else
  Call displayOut(2, 0, "Sidik Jari Anda Tidak Cocok.")
  Call displayOut(3, 0, "Silahkan Masukkan Sidik jari Anda Lagi.")
  'Call ResetInitConnect
  Exit Sub
End If

```

'Menampilkan informasi user yang terdapat pada kartu

```

'Pilih file BB BB
retCode = selectFile(&HBB, &HBB)
If retCode <> SCARD_S_SUCCESS Then
  Exit Sub
End If
tmpStr = ""
For indx = 0 To 1
  tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "91 01 " Then
  Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)

```

```

Exit Sub
End If

'Membaca dan Membandingkan Id Kartu di db dengan di kartu
Dim Id_Kartu, IdKartu, IdNasabah, No_Rekening, NoRekening As String
retCode = readRecord(&H0, &H20)
If retCode <> SCARD_S_SUCCESS Then
Exit Sub
End If
tmpStr = ""
indx = 0
While RecvBuff(indx) <> &H0
tmpStr = tmpStr + Chr(RecvBuff(indx))
indx = indx + 1
Wend
Id_Kartu = tmpStr

Query "SELECT * From Nasabah where Id_Kartu='" & Id_Kartu & "'"
If Not rec.EOF Then
IdNasabah = rec!Id_Nasabah
End If

Query "SELECT * From Tabungan where Id_Nasabah='" & IdNasabah & "'"
If Not rec.EOF Then
NoRekening = rec!No_Rekening
End If

retCode = readRecord(&H1, &H20)
If retCode <> SCARD_S_SUCCESS Then
Exit Sub
End If
tmpStr = ""
indx = 0
While RecvBuff(indx) <> &H0
tmpStr = tmpStr + Chr(RecvBuff(indx))
indx = indx + 1
Wend
No_Rekening = tmpStr

If No_Rekening = NoRekening Then
Load Menu: Menu.Show: Unload Me
Else
Call displayOut(2, 0, "Silahkan Masukkan Sidik Jari Anda Lagi.")
End If
End Sub

```

```

Private Sub clearMessages()
Call displayOut(2, 0, "")
Call displayOut(3, 0, "")
userAction.Refresh
MsgBox.Refresh
End Sub

```

```

Private Sub ClearBuffers()
Dim indx As Long
For indx = 0 To 262
SendBuff(indx) = &H0
RecvBuff(indx) = &H0
Next indx
End Sub

```

```

Private Sub displayOut(ByVal errType As Integer, ByVal retVal As Long, ByVal msgText As String)
Select Case (errType)
Case 0 ' Smartcard error
MsgBox.Text = GetScardErrMsg(retVal)
Case 1 ' BioAPI error

```

```

    Dim StatusMsg As String
    StatusMsg = String(255, vbNullChar)
    Call AET60_GetStatus(retVal, StatusMsg)
    MsgBox.Text = StatusMsg
    Case 2 ' Message only
        MsgBox.Text = msgText
    Case 3 ' Required user action
        userAction.Text = msgText
    End Select
End Sub

```

Private Function validIDCard() As Boolean

```

Dim tmpStr As String
Dim indx As Integer

```

```

validIDCard = False

```

```

' 1. Select CC CC

```

```

retCode = selectFile(&HCC, &HCC)

```

```

If retCode <> SCARD_S_SUCCESS Then

```

```

    Exit Function

```

```

End If

```

```

tmpStr = ""

```

```

For indx = 0 To 1

```

```

    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "

```

```

Next indx

```

```

If tmpStr <> "91 02 " Then

```

```

    Call displayOut(2, 0, "Inserted ID card may be invalid. Return value: " + tmpStr)

```

```

    Exit Function

```

```

End If

```

```

' 2. Read ID Flag

```

```

retCode = readRecord(&H0, &H6)

```

```

If retCode <> SCARD_S_SUCCESS Then

```

```

    Exit Function

```

```

End If

```

```

tmpStr = ""

```

```

For indx = 0 To 3

```

```

    tmpStr = tmpStr + Chr(RecvBuff(indx + 2))

```

```

Next indx

```

```

If tmpStr <> "KNEF" Then

```

```

    Call displayOut(2, 0, "The user is not using a valid ID card.")

```

```

    Exit Function

```

```

End If

```

```

validIDCard = True

```

```

End Function

```

Private Function SubmitIC() As Long

```

Dim indx As Integer

```

```

Dim tmpStr As String

```

```

Call ClearBuffers

```

```

SendBuff(0) = &H80 ' CLA

```

```

SendBuff(1) = &H20 ' INS

```

```

SendBuff(2) = &H7 ' P1

```

```

SendBuff(3) = &H0 ' P2

```

```

SendBuff(4) = &H8 ' P3

```

```

SendBuff(5) = &H41 ' A

```

```

SendBuff(6) = &H43 ' C

```

```

SendBuff(7) = &H4F ' O

```

```

SendBuff(8) = &H53 ' S

```

```

SendBuff(9) = &H54 ' T

```

```

SendBuff(10) = &H45 ' E

```

```

SendBuff(11) = &H53 ' S

```

```

SendBuff(12) = &H54 ' T

```

```

SendLen = &HD

```

```

RecvLen = &H2

```

```

retCode = SendAPDU()

```

```

If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    SubmitIC = retCode
    Exit Function
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    retCode = INVALID_SW1SW2
End If
SubmitIC = retCode
End Function

```

```

Private Function selectFile(ByVal HiAddr As Byte, ByVal LoAddr As Byte) As Long
    Call ClearBuffers
    SendBuff(0) = &H80 ' CLA
    SendBuff(1) = &HA4 ' INS
    SendBuff(2) = &H0 ' P1
    SendBuff(3) = &H0 ' P2
    SendBuff(4) = &H2 ' P3
    SendBuff(5) = HiAddr ' Value of High Byte
    SendBuff(6) = LoAddr ' Value of Low Byte
    SendLen = &H7
    RecvLen = &H2
    retCode = SendAPDU()
    If retCode <> SCARD_S_SUCCESS Then
        Call displayOut(0, retCode, "")
    End If
    selectFile = retCode
End Function

```

```

Private Function readRecord(ByVal RecNo As Byte, ByVal DataLen As Byte) As Long
    Dim indx As Integer
    Dim tmpStr As String

    Call ClearBuffers
    SendBuff(0) = &H80 ' CLA
    SendBuff(1) = &HB2 ' INS
    SendBuff(2) = RecNo ' P1 Record No
    SendBuff(3) = &H0 ' P2
    SendBuff(4) = DataLen ' P3 Length of data to be read
    SendLen = &H5
    RecvLen = SendBuff(4) + 2
    retCode = SendAPDU()
    If retCode <> SCARD_S_SUCCESS Then
        Call displayOut(0, retCode, "")
        readRecord = retCode
        Exit Function
    End If
    tmpStr = ""
    For indx = 0 To 1
        tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx + SendBuff(4))), 2) + " "
    Next indx
    If tmpStr <> "90 00 " Then
        Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
        retCode = INVALID_SW1SW2
    End If
    readRecord = retCode
End Function

```

```

Private Function writeRecord(ByVal caseType As Integer, ByVal RecNo As Byte, _
    ByVal maxDataLen As Byte, ByVal DataLen As Byte, _
    ByRef DataIn() As Byte) As Long
    Dim tmpStr As String

```

Dim indx As Integer

If caseType = 1 Then

' 1. Re-initialize card values to \$00

Call ClearBuffers

SendBuff(0) = &H80 ' CLA

SendBuff(1) = &HD2 ' INS

SendBuff(2) = RecNo ' P1 Record to be written

SendBuff(3) = &H0 ' P2

SendBuff(4) = maxDataLen ' P3 Length

For indx = 0 To maxDataLen - 1

SendBuff(indx + 5) = &H0

Next indx

SendLen = maxDataLen + 5

RecvLen = &H2

retCode = SendAPDU()

If retCode <> SCARD_S_SUCCESS Then

Call displayOut(0, retCode, "")

writeRecord = retCode

Exit Function

End If

tmpStr = ""

For indx = 0 To 1

tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "

Next indx

If tmpStr <> "90 00 " Then

Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)

retCode = INVALID_SW1SW2

writeRecord = retCode

Exit Function

End If

End If

' 2. Write data to card

Call ClearBuffers

SendBuff(0) = &H80 ' CLA

SendBuff(1) = &HD2 ' INS

SendBuff(2) = RecNo ' P1 Record to be written

SendBuff(3) = &H0 ' P2

SendBuff(4) = DataLen ' P3 Length

For indx = 0 To DataLen - 1

SendBuff(indx + 5) = DataIn(indx)

Next indx

SendLen = DataLen + 5

RecvLen = &H2

retCode = SendAPDU()

If retCode <> SCARD_S_SUCCESS Then

Call displayOut(0, retCode, "")

writeRecord = retCode

Exit Function

End If

tmpStr = ""

For indx = 0 To 1

tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "

Next indx

If tmpStr <> "90 00 " Then

Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)

retCode = INVALID_SW1SW2

End If

writeRecord = retCode

End Function

Private Function SendAPDU() As Long

ioRequest.dwProtocol = dwActProtocol

ioRequest.cbPciLength = Len(ioRequest)

retCode = SCardTransmit(hCard, _

```
        ioRequest, _
        SendBuff(0), _
        SendLen, _
        ioRequest, _
        RecvBuff(0), _
        RecvLen)
SendAPDU = retCode
End Function
```

Public Function GetScardErrMsg(ByVal ReturnCode As Long) As String

```
Select Case ReturnCode
'Smartcard Reader Errors
Case SCARD_E_CANCELLED
GetScardErrMsg = "The action was canceled by an SCardCancel request."
Case SCARD_E_CANT_DISPOSE
GetScardErrMsg = "The system could not dispose of the media in the requested manner."
Case SCARD_E_CARD_UNSUPPORTED
GetScardErrMsg = "The smart card does not meet minimal requirements for support."
Case SCARD_E_DUPLICATE_READER
GetScardErrMsg = "The reader driver didn't produce a unique reader name."
Case SCARD_E_INSUFFICIENT_BUFFER
GetScardErrMsg = "The data buffer for returned data is too small for the returned data."
Case SCARD_E_INVALID_ATR
GetScardErrMsg = "An ATR string obtained from the registry is not a valid ATR string."
Case SCARD_E_INVALID_HANDLE
GetScardErrMsg = "The supplied handle was invalid."
Case SCARD_E_INVALID_PARAMETER
GetScardErrMsg = "One or more of the supplied parameters could not be properly interpreted."
Case SCARD_E_INVALID_TARGET
GetScardErrMsg = "Registry startup information is missing or invalid."
Case SCARD_E_INVALID_VALUE
GetScardErrMsg = "One or more of the supplied parameter values could not be properly interpreted."
Case SCARD_E_NOT_READY
GetScardErrMsg = "The reader or card is not ready to accept commands."
Case SCARD_E_NOT_TRANSACTED
GetScardErrMsg = "An attempt was made to end a non-existent transaction."
Case SCARD_E_NO_MEMORY
GetScardErrMsg = "Not enough memory available to complete this command."
Case SCARD_E_NO_SERVICE
GetScardErrMsg = "The smart card resource manager is not running."
Case SCARD_E_NO_SMARTCARD
GetScardErrMsg = "The operation requires a smart card, but no smart card is currently in the device."
Case SCARD_E_PCI_TOO_SMALL
GetScardErrMsg = "The PCI receive buffer was too small."
Case SCARD_E_PROTO_MISMATCH
GetScardErrMsg = "The requested protocols are incompatible with the protocol currently in use with the card."
Case SCARD_E_READER_UNAVAILABLE
GetScardErrMsg = "The specified reader is not currently available for use."
Case SCARD_E_READER_UNSUPPORTED
GetScardErrMsg = "The reader driver does not meet minimal requirements for support."
Case SCARD_E_SERVICE_STOPPED
GetScardErrMsg = "The smart card resource manager has shut down."
Case SCARD_E_SHARING_VIOLATION
GetScardErrMsg = "The smart card cannot be accessed because of other outstanding connections."
Case SCARD_E_SYSTEM_CANCELLED
GetScardErrMsg = "The action was canceled by the system, presumably to log off or shut down."
Case SCARD_E_TIMEOUT
GetScardErrMsg = "The user-specified timeout value has expired."
Case SCARD_E_UNKNOWN_CARD
GetScardErrMsg = "The specified smart card name is not recognized."
Case SCARD_E_UNKNOWN_READER
GetScardErrMsg = "The specified reader name is not recognized."
Case SCARD_F_COMM_ERROR
GetScardErrMsg = "An internal communications error has been detected."
Case SCARD_F_INTERNAL_ERROR
GetScardErrMsg = "An internal consistency check failed."
Case SCARD_F_UNKNOWN_ERROR
```



```

GetScardErrMsg = "An internal error has been detected, but the source is unknown."
Case SCARD_F_WAITED_TOO_LONG
GetScardErrMsg = "An internal consistency timer has expired."
Case SCARD_S_SUCCESS
GetScardErrMsg = "No error was encountered."
Case SCARD_W_REMOVED_CARD
GetScardErrMsg = ""
Case SCARD_W_RESET_CARD
GetScardErrMsg = "The smart card has been reset, so any shared state information is invalid."
Case SCARD_W_UNPOWERED_CARD
GetScardErrMsg = "Power has been removed from the smart card, so that further communication is not possible."
Case SCARD_W_UNRESPONSIVE_CARD
GetScardErrMsg = "The smart card is not responding to a reset."
Case SCARD_W_UNSUPPORTED_CARD
GetScardErrMsg = "The reader cannot communicate with the card, due to ATR string configuration conflicts."
Case Else
GetScardErrMsg = "?"
End Select
End Function

```

```
Private Sub Timer1_Timer()
```

```
'Melakukan re-connect untuk menghilangkan error pada kasus kartu belum dimasukkan
```

```
retCode = SCardDisconnect(hCard, SCARD_UNPOWER_CARD)
```

```
scInit = False
```

```
retCode = SCardConnect(hContext, _
    cbReader.Text, _
    SCARD_SHARE_SHARED, _
    SCARD_PROTOCOL_T0 Or SCARD_PROTOCOL_T1, _
    hCard, _
    Protocol)
```

```
If retCode <> SCARD_S_SUCCESS Then
```

```
    Call displayOut(0, retCode, "")
```

```
    Call displayOut(3, 0, "Insert your ID card into the reader.")
```

```
Exit Sub
```

```
Else
```

```
    Call ResetInitConnect
```

```
    Call verify
```

```
Exit Sub
```

```
End If
```

```
scInit = True
```

```
End Sub
```

```
Form DataIsiPulsa
```

```
Private Type RECT
```

```
    Left As Long
```

```
    Top As Long
```

```
    Right As Long
```

```
    Bottom As Long
```

```
End Type
```

```
Dim DefaultPrinterScaleMode As Integer
```

```
Private Const DT_WORDBREAK = &H10
```

```
Private Declare Function DrawText Lib "user32" Alias "DrawTextA" (ByVal hDC As Long, ByVal lpStr As String, ByVal nCount As Long, lpRect As RECT, ByVal wFormat As Long) As Long
```

```
Private Declare Function GetDesktopWindow Lib "user32" () As Long
```

```
Private Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" _
```

```
    (ByVal hwnd As Long, ByVal lpzOp As String, _
```

```
    ByVal lpzFile As String, ByVal lpzParams As String, _
```

```
    ByVal lpzDir As String, ByVal FsShowCmd As Long) _
```

```
    As Long
```

```
Function PrintBoxedText(strTextToPrint As String, sngX As Single, sngY As Single, sngMaxWith As Single, sngMaxHeight As Single) As Single
```

```
Dim rc As RECT
```

```
Dim lngXpos As Long, lngYpos As Long, lngWdt As Long, heightused As Long, lngMaxHgt As Long
```

```

Dim hPrintDc As Long
Printer.ScaleMode = DefaultPrinterScaleMode

hPrintDc = Printer.hDC
lngWdt = Printer.ScaleX(sngMaxWith, DefaultPrinterScaleMode, vbPixels)
lngXpos = Printer.ScaleX(sngX, DefaultPrinterScaleMode, vbPixels)
lngYpos = Printer.ScaleY(sngY, DefaultPrinterScaleMode, vbPixels)
lngMaxHgt = Printer.ScaleY(sngMaxHeight, DefaultPrinterScaleMode, vbPixels)
rc.Left = lngXpos
rc.Right = lngWdt + rc.Left
rc.Top = lngYpos
rc.Bottom = lngMaxHgt + rc.Top
'Fool the printer to make it believe that this is a standard printjob.
Printer.Print " "

```

```
heightused = DrawText(hPrintDc, strTextToPrint, -1, rc, DT_WORDBREAK)
```

```
PrintBoxedText = Printer.ScaleY(heightused, vbPixels, DefaultPrinterScaleMode) + sngY
```

```
End Function
```

```

Private Sub Benar_Click()
If Val(saldo.Text) <= Val(jumlah.Caption) Then
Load SaldoSalah: SaldoSalah.Show: Unload Me
Else
saldo.Text = Val(saldo.Text) - Val(jumlah.Caption)

con.Execute "UPDATE Tabungan SET Saldo=" & _
    "" & saldo.Text & " WHERE No_Rekening=" & rekening.Text & ""

con.Execute "INSERT INTO Transaksi VALUES (" & tipe.Text & "," & nomor.Text & "," & id.Text & ")"

con.Execute "INSERT INTO Provider VALUES (" & tipe.Text & "," & nomor.Text & "," & Idpro.Text & "," & _
    "" & provider.Caption & "," & tanggal.Text & "," & waktu.Text & "," & nohandphone.Caption & "," & _
jumlah.Caption & ")"

pro.Text = "PROVIDER          : " & provider.Caption
handphone.Text = "NOMOR HANDPHONE : " & nohandphone.Caption
jmlh.Text = "JUMLAH              : " & jumlah.Caption

Dim sngNextYCor, sngNextXCor As Single
sngNextYCor = PrintBoxedText(jdl.Text, CSng(10), CSng(10), CSng(100), CSng(300))

sngNextXCor = PrintBoxedText(tanggal.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(waktu.Text, sngNextXCor + 15, sngNextYCor + 1, CSng(100), CSng(300))

sngNextXCor = PrintBoxedText(id.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(lokasi.Text, sngNextXCor + 5, sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(jenis.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(pro.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(handphone.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(jmlh.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

Printer.EndDoc
Load SilahkanTunggu: SilahkanTunggu.Show: Unload Me
End If
End Sub

```

```
Private Sub Form_Activate()
```

```
jdl.Text = " ** ATM ADRIAN ** "  
tanggal.Text = "" & Date  
waktu.Text = "" & Time  
lokasi.Text = "MARANATHA"  
jenis.Text = "ISI PULSA"
```

```
Query "SELECT * FROM ATM WHERE Lokasi = "" & lokasi.Text & ""  
If rec.EOF = False Then  
id.Text = rec!Id_ATM  
End If
```

Dim a As Integer

```
Query "SELECT * FROM Transaksi WHERE Id_Tipe = "" & tipe.Text & ""  
If Not rec.EOF Then  
rec.MoveLast: a = rec!Id_Nomor_Urut  
nomor.Text = a + 1  
End If
```

```
Query "SELECT * FROM Tabungan WHERE No_Rekening = "" & rekening.Text & ""  
If Not rec.EOF Then  
saldo.Text = rec!saldo  
End If
```

```
Query "SELECT * FROM Provider_Esia WHERE No_handphone = "" & nohandphone.Caption & ""  
If Not rec.EOF Then  
provider.Caption = rec!provider  
Idpro.Text = rec!Id_Provider  
End If
```

```
Query "SELECT * FROM Provider_Flexi WHERE No_handphone = "" & nohandphone.Caption & ""  
If Not rec.EOF Then  
provider.Caption = rec!provider  
Idpro.Text = rec!Id_Provider  
End If
```

```
Query "SELECT * FROM Provider_Indosat WHERE No_handphone = "" & nohandphone.Caption & ""  
If Not rec.EOF Then  
provider.Caption = rec!provider  
Idpro.Text = rec!Id_Provider  
End If
```

```
Query "SELECT * FROM Provider_Telkomsel WHERE No_handphone = "" & nohandphone.Caption & ""  
If Not rec.EOF Then  
provider.Caption = rec!provider  
Idpro.Text = rec!Id_Provider  
End If
```

```
Query "SELECT * FROM Provider_XI WHERE No_handphone = "" & nohandphone.Caption & ""  
If Not rec.EOF Then  
provider.Caption = rec!provider  
Idpro.Text = rec!Id_Provider  
End If
```

```
If provider.Caption = "" Then  
Load NomorSalah: NomorSalah.Show: Unload Me  
End If
```

```
DefaultPrinterScaleMode = vbMillimeters ' Set default scalemode on printer to Millimeters
```

```
End Sub
```

```
Private Sub Salah_Click()  
Load AET60Main: AET60Main.Show: Unload Me  
End Sub
```

Form DataPembayaran

Private Type RECT

Left As Long
Top As Long
Right As Long
Bottom As Long

End Type

Dim DefaultPrinterScaleMode As Integer

Private Const DT_WORDBREAK = &H10

Private Declare Function DrawText Lib "user32" Alias "DrawTextA" (ByVal hDC As Long, ByVal lpStr As String, ByVal nCount As Long, lpRect As RECT, ByVal wFormat As Long) As Long

Private Declare Function GetDesktopWindow Lib "user32" () As Long

Private Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" _
(ByVal hwnd As Long, ByVal lpzOp As String, _
ByVal lpzFile As String, ByVal lpzParams As String, _
ByVal lpzDir As String, ByVal FsShowCmd As Long) _
As Long

Function PrintBoxedText(strTextToPrint As String, sngX As Single, sngY As Single, sngMaxWith As Single, sngMaxHeight As Single) As Single

Dim rc As RECT

Dim lngXpos As Long, lngYpos As Long, lngWdt As Long, heightused As Long, lngMaxHgt As Long

Dim hPrintDc As Long

Printer.ScaleMode = DefaultPrinterScaleMode

hPrintDc = Printer.hDC

lngWdt = Printer.ScaleX(sngMaxWith, DefaultPrinterScaleMode, vbPixels)

lngXpos = Printer.ScaleX(sngX, DefaultPrinterScaleMode, vbPixels)

lngYpos = Printer.ScaleY(sngY, DefaultPrinterScaleMode, vbPixels)

lngMaxHgt = Printer.ScaleY(sngMaxHeight, DefaultPrinterScaleMode, vbPixels)

rc.Left = lngXpos

rc.Right = lngWdt + rc.Left

rc.Top = lngYpos

rc.Bottom = lngMaxHgt + rc.Top

'Fool the printer to make it believe that this is a standard printjob.

Printer.Print " "

heightused = DrawText(hPrintDc, strTextToPrint, -1, rc, DT_WORDBREAK)

PrintBoxedText = Printer.ScaleY(heightused, vbPixels, DefaultPrinterScaleMode) + sngY

End Function

Private Sub Benar_Click()

If Val(saldo.Text) <= Val(jmlh.Caption) Then

Load SaldoSalah: SaldoSalah.Show: Unload Me

Else

saldo.Text = Val(saldo.Text) - Val(jmlh.Caption)

con.Execute "UPDATE Tabungan SET Saldo=" & _
"" & saldo.Text & " WHERE No_Rekening=" & rekening.Text & ""

con.Execute "INSERT INTO Transaksi VALUES (" & tipe.Text & "," & nomor.Text & "," & id.Text & ")"

con.Execute "INSERT INTO Pembayaran VALUES (" & tipe.Text & "," & nomor.Text & "," & idpmb.Text & "," & _
"" & jenis.Caption & "," & tanggal.Text & "," & waktu.Text & "," & jmlh.Caption & ")"

Dim sngNextYCor, sngNextXCor As Single

sngNextYCor = PrintBoxedText(jdl.Text, CSng(10), CSng(10), CSng(100), CSng(300))

sngNextXCor = PrintBoxedText(tanggal.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

```

sngNextYCor = PrintBoxedText(waktu.Text, sngNextXCor + 15, sngNextYCor + 1, CSng(100), CSng(300))

sngNextXCor = PrintBoxedText(id.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(lokasi.Text, sngNextXCor + 5, sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(jns.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(nm.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(no.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(jml.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

Printer.EndDoc
Load SilahkanTunggu: SilahkanTunggu.Show: Unload Me
End If
End Sub

```

```

Private Sub Form_Activate()
jdl.Text = "    ** ATM ADRIAN **    "
tanggal.Text = "" & Date
waktu.Text = "" & Time
lokasi.Text = "MARANATHA"
nm.Text = "NAMA CUSTOMER : " & nama.Caption
no.Text = "NOMOR CUSTOMER : " & nocustomer.Caption
jns.Text = "" & jenis.Caption
jml.Text = "JUMLAH            : " & jmlh.Caption

Query "SELECT * FROM ATM WHERE Lokasi = " & lokasi.Text & ""
If rec.EOF = False Then
id.Text = rec!Id_ATM
End If

Dim a As Integer
Query "SELECT * FROM Transaksi WHERE Id_Tipe = " & tipe.Text & ""
If Not rec.EOF Then
rec.MoveLast: a = rec!Id_Nomor_Urut
nomor.Text = a + 1
End If

Query "SELECT * FROM Tabungan WHERE No_Rekening = " & rekening.Text & ""
If Not rec.EOF Then
saldo.Text = rec!saldo
End If

DefaultPrinterScaleMode = vbMillimeters ' Set default scalemode on printer to Millimeters

End Sub

```

```

Private Sub Salah_Click()
Load AET60Main: AET60Main.Show: Unload Me
End Sub

```

Form DataTransfer

```

Option Explicit
Dim scInit, rdrLocked As Boolean
Dim retCode, numDev, numRdr, hContext, hCard, Protocol, RetLength As Long
Dim dwActProtocol, RecvLen, SendLen As Long
Dim ioRequest As SCARD_IO_REQUEST

```

```

Dim FpTemplate(0 To 767) As Byte
Dim SendBuff(0 To 262) As Byte
Dim RecvBuff(0 To 262) As Byte
Dim MatchResult As Integer
Dim apdu As APDUREc
Dim Transaction As SCARD_IO_REQUEST
Dim rName, scannerName As String
Dim fpTemplateLength, procTimeOut As Long
Dim rList As String * 256
Dim desFar, desFrr, achFar, achFrr As Long
Const MAX_BUFFER_LEN = 256
Const INVALID_SW1SW2 = -450

Private Type RECT
    Left As Long
    Top As Long
    Right As Long
    Bottom As Long
End Type
Dim DefaultPrinterScaleMode As Integer
Private Const DT_WORDBREAK = &H10
Private Declare Function DrawText Lib "user32" Alias "DrawTextA" (ByVal hDC As Long, ByVal lpStr As String, ByVal nCount As Long, lpRect As RECT, ByVal wFormat As Long) As Long
Private Declare Function GetDesktopWindow Lib "user32" () As Long
Private Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" _
    (ByVal hwnd As Long, ByVal lpzOp As String, _
    ByVal lpzFile As String, ByVal lpzParams As String, _
    ByVal lpzDir As String, ByVal FsShowCmd As Long) _
    As Long

```

```
Private Sub Benar_Click()
```

```

If Val(Text5.Text) <= Val(Label6.Caption) Then
Load SaldoSalah: SaldoSalah.Show: Unload Me
Else
con.Execute "UPDATE Tabungan SET Saldo=" & _
    "" & Text1.Text & " WHERE No_Rekening=" & Text2.Text & ""

con.Execute "UPDATE Tabungan SET Saldo=" & _
    "" & Text4.Text & " WHERE No_Rekening=" & Text3.Text & ""

con.Execute "INSERT INTO Transaksi VALUES (" & transfer.Text & "," & nomor.Text & "," & id.Text & ")"

con.Execute "INSERT INTO Transfer VALUES (" & transfer.Text & "," & nomor.Text & "," & Label5.Caption & "," &
-
    "" & tgl.Text & "," & jam.Text & "," & Label6.Caption & ")"

```

```

Dim sngNextYCor, sngNextXCor As Single
sngNextYCor = PrintBoxedText(jdl.Text, CSng(10), CSng(10), CSng(100), CSng(300))

sngNextXCor = PrintBoxedText(tgl.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(jam.Text, sngNextXCor + 15, sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(id.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(lokasi.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(jenis.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(kerek.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(nama.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(jmlh.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

Printer.EndDoc

```

```
Load SilahkanTunggu: SilahkanTunggu.Show: Unload Me
End If
End Sub
```

```
Private Sub Form_Activate()
Dim tmpStr As String
  Dim ctr, indx, fileNo, byteNo As Long
  Dim tmpTemplate(0 To 767) As Byte

Call ResetInitConnect
'Membaca dan Membandingkan Id Kartu di db dengan di kartu
  Dim No_Rekening As String
  retCode = readRecord(&H1, &H20)
  If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
  End If
  tmpStr = ""
  indx = 0
  While RecvBuff(indx) <> &H0
    tmpStr = tmpStr + Chr(RecvBuff(indx))
    indx = indx + 1
  Wend
  Text2.Text = tmpStr

  jdl.Text = "  ** ATM ADRIAN **  "
  tgl.Text = "" & Date
  jam.Text = "" & Time
  lokasi.Text = "MARANATHA"
  jenis.Text = "TRANSFER"
  kerek.Text = "KE REK    : " & Label5.Caption
  nama.Text = "NAMA      : " & Label4.Caption
  jmlh.Text = "JUMLAH    : " & Label6.Caption
  DefaultPrinterScaleMode = vbMillimeters ' Set default scalemode on printer to Millimeters

Query "SELECT * FROM ATM WHERE Lokasi = " & lokasi.Text & ""
  If rec.EOF = False Then
    id.Text = rec!Id_ATM
  End If

Dim a As Integer
  Query "SELECT * FROM Transaksi WHERE Id_Tipe = " & transfer.Text & ""
  If Not rec.EOF Then
    rec.MoveLast: a = rec!Id_Nomor_Urut
    nomor.Text = a + 1
  End If

End Sub
```

```
Private Sub Salah_Click()
Load AET60Main: AET60Main.Show: Unload Me
End Sub
```

```
Function PrintBoxedText(strTextToPrint As String, sngX As Single, sngY As Single, sngMaxWith As Single,
sngMaxHeight As Single) As Single
```

```
Dim rc As RECT
Dim lngXpos As Long, lngYpos As Long, lngWdt As Long, heightused As Long, lngMaxHgt As Long
Dim hPrintDc As Long
Printer.ScaleMode = DefaultPrinterScaleMode
```

```
hPrintDc = Printer.hDC
lngWdt = Printer.ScaleX(sngMaxWith, DefaultPrinterScaleMode, vbPixels)
lngXpos = Printer.ScaleX(sngX, DefaultPrinterScaleMode, vbPixels)
lngYpos = Printer.ScaleY(sngY, DefaultPrinterScaleMode, vbPixels)
lngMaxHgt = Printer.ScaleY(sngMaxHeight, DefaultPrinterScaleMode, vbPixels)
rc.Left = lngXpos
rc.Right = lngWdt + rc.Left
```

```
rc.Top = lngYpos
rc.Bottom = lngMaxHgt + rc.Top
Printer.Print " "
```

```
heightused = DrawText(hPrintDc, strTextToPrint, -1, rc, DT_WORDBREAK)
```

```
PrintBoxedText = Printer.ScaleY(heightused, vbPixels, DefaultPrinterScaleMode) + sngY
```

End Function

```
Private Sub ResetInitConnect()
```

```
'Menon-aktifkan FP scanner
```

```
If rdrLocked Then
```

```
    retCode = AET60_UnlockReader()
```

```
End If
```

```
retCode = AET60_Close()
```

```
'Menon-aktifkan SC reader
```

```
If scInit Then
```

```
    retCode = SCardDisconnect(hCard, SCARD_UNPOWER_CARD)
```

```
End If
```

```
retCode = SCardReleaseContext(hContext)
```

```
scInit = False
```

```
rdrLocked = False
```

```
Dim indx As Long
```

```
rList = String(255, vbNullChar)
```

```
numRdr = 255
```

```
'Inisialisasi SC Reader
```

```
retCode = SCardEstablishContext(SCARD_SCOPE_USER, 0, 0, hContext)
```

```
If retCode <> SCARD_S_SUCCESS Then
```

```
    Call displayOut(0, retCode, "")
```

```
    Exit Sub
```

```
End If
```

```
retCode = SCardListReaders(hContext, rName, rList, numRdr)
```

```
If retCode <> SCARD_S_SUCCESS Then
```

```
    Call displayOut(0, retCode, "")
```

```
    Exit Sub
```

```
End If
```

```
Call LoadListToControl(cbReader, rList)
```

```
cbReader.ListIndex = 0
```

```
'Inisialisasi FP Scanner
```

```
retCode = AET60_Open()
```

```
If retCode <> 0 Then
```

```
    Call displayOut(1, retCode, "")
```

```
    Exit Sub
```

```
End If
```

```
numDev = AET60_GetNumDevices()
```

```
If numDev > 0 Then
```

```
    cbScanner.Clear
```

```
    scannerName = String(255, vbNullChar)
```

```
    RetLength = Len(scannerName)
```

```
    For indx = 0 To (numDev - 1)
```

```
        retCode = AET60_GetDeviceName(indx, _  
            scannerName, _  
            RetLength)
```

```
        cbScanner.AddItem scannerName
```

```
        cbScanner.ListIndex = 0
```

```
    Next indx
```

```
Else
```

```
    Call displayOut(1, retCode, "")
```

```
    Exit Sub
```

```
End If
```



```

    Call displayOut(2, 0, "")
    Call displayOut(3, 0, "")

' 1. Connect to smartcard
retCode = SCardConnect(hContext, _
    cbReader.Text, _
    SCARD_SHARE_SHARED, _
    SCARD_PROTOCOL_T0 Or SCARD_PROTOCOL_T1, _
    hCard, _
    Protocol)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If
scInit = True

' 2. Lock scanner and set callback
retCode = AET60_LockReader(cbScanner.ListIndex)
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If
rdrLocked = True

retCode = AET60_SetCallback(AddressOf CALLBACK_FUNCTION)
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If

Dim tmpStr As String
Dim ctr, fileNo, byteNo As Long
Dim tmpTemplate(0 To 767) As Byte

Call clearMessages

'Menampilkan informasi user yang terdapat pada kartu
'Pilih file BB BB
retCode = selectFile(&HBB, &HBB)
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "91 01 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    Exit Sub
End If

End Sub

```

```

Private Sub clearMessages()
    Call displayOut(2, 0, "")
    Call displayOut(3, 0, "")
    userAction.Refresh
    MsgBox.Refresh
End Sub

```

```

Private Sub ClearBuffers()
    Dim indx As Long
    For indx = 0 To 262
        SendBuff(indx) = &H0
        RecvBuff(indx) = &H0
    Next indx

```

End Sub

Private Sub displayOut(ByVal errType As Integer, ByVal retVal As Long, ByVal msgText As String)

```
Select Case (errType)
    Case 0 ' Smartcard error
        userAction.Text = "Error encountered. Please click on Reset button."
        MsgBox.Text = GetScardErrMsg(retVal)
    Case 1 ' BioAPI error
        Dim StatusMsg As String
        StatusMsg = String(255, vbNullChar)
        userAction.Text = "Error encountered. Please click on Reset button."
        Call AET60_GetStatus(retVal, StatusMsg)
        MsgBox.Text = StatusMsg
    Case 2 ' Message only
        MsgBox.Text = msgText
    Case 3 ' Required user action
        userAction.Text = msgText
End Select
End Sub
```

Private Function validIDCard() As Boolean

```
Dim tmpStr As String
Dim indx As Integer

validIDCard = False
' 1. Select CC CC
retCode = selectFile(&HCC, &HCC)
If retCode <> SCARD_S_SUCCESS Then
    Exit Function
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "91 02 " Then
    Call displayOut(2, 0, "Inserted ID card may be invalid. Return value: " + tmpStr)
    Exit Function
End If

' 2. Read ID Flag
retCode = readRecord(&H0, &H6)
If retCode <> SCARD_S_SUCCESS Then
    Exit Function
End If
tmpStr = ""
For indx = 0 To 3
    tmpStr = tmpStr + Chr(RecvBuff(indx + 2))
Next indx
If tmpStr <> "KNEF" Then
    Call displayOut(2, 0, "The user is not using a valid ID card.")
    Exit Function
End If
validIDCard = True
End Function
```

Private Function SubmitIC() As Long

```
Dim indx As Integer
Dim tmpStr As String

Call ClearBuffers
SendBuff(0) = &H80 ' CLA
SendBuff(1) = &H20 ' INS
SendBuff(2) = &H7 ' P1
SendBuff(3) = &H0 ' P2
SendBuff(4) = &H8 ' P3
SendBuff(5) = &H41 ' A
```

```

SendBuff(6) = &H43 ' C
SendBuff(7) = &H4F ' O
SendBuff(8) = &H53 ' S
SendBuff(9) = &H54 ' T
SendBuff(10) = &H45 ' E
SendBuff(11) = &H53 ' S
SendBuff(12) = &H54 ' T
SendLen = &HD
RecvLen = &H2
retCode = SendAPDU()
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    SubmitIC = retCode
    Exit Function
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    retCode = INVALID_SW1SW2
End If
SubmitIC = retCode
End Function

```

Private Function selectFile(ByVal HiAddr As Byte, ByVal LoAddr As Byte) As Long

```

Call ClearBuffers
SendBuff(0) = &H80 ' CLA
SendBuff(1) = &HA4 ' INS
SendBuff(2) = &H0 ' P1
SendBuff(3) = &H0 ' P2
SendBuff(4) = &H2 ' P3
SendBuff(5) = HiAddr ' Value of High Byte
SendBuff(6) = LoAddr ' Value of Low Byte
SendLen = &H7
RecvLen = &H2
retCode = SendAPDU()
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
End If
selectFile = retCode
End Function

```

Private Function readRecord(ByVal RecNo As Byte, ByVal DataLen As Byte) As Long

```

Dim indx As Integer
Dim tmpStr As String

Call ClearBuffers
SendBuff(0) = &H80 ' CLA
SendBuff(1) = &HB2 ' INS
SendBuff(2) = RecNo ' P1 Record No
SendBuff(3) = &H0 ' P2
SendBuff(4) = DataLen ' P3 Length of data to be read
SendLen = &H5
RecvLen = SendBuff(4) + 2
retCode = SendAPDU()
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    readRecord = retCode
    Exit Function
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx + SendBuff(4))), 2) + " "
Next indx
If tmpStr <> "90 00 " Then

```

```

    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    retCode = INVALID_SW1SW2
End If
readRecord = retCode
End Function

```

```

Private Function writeRecord(ByVal caseType As Integer, ByVal RecNo As Byte, _
    ByVal maxDataLen As Byte, ByVal DataLen As Byte, _
    ByVal DataIn() As Byte) As Long
    Dim tmpStr As String
    Dim indx As Integer

    If caseType = 1 Then
        ' 1. Re-initialize card values to $00
        Call ClearBuffers
        SendBuff(0) = &H80      ' CLA
        SendBuff(1) = &HD2      ' INS
        SendBuff(2) = RecNo     ' P1  Record to be written
        SendBuff(3) = &H0       ' P2
        SendBuff(4) = maxDataLen ' P3  Length
        For indx = 0 To maxDataLen - 1
            SendBuff(indx + 5) = &H0
        Next indx
        SendLen = maxDataLen + 5
        RecvLen = &H2
        retCode = SendAPDU()
        If retCode <> SCARD_S_SUCCESS Then
            Call displayOut(0, retCode, "")
            writeRecord = retCode
            Exit Function
        End If
        tmpStr = ""
        For indx = 0 To 1
            tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
        Next indx
        If tmpStr <> "90 00 " Then
            Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
            retCode = INVALID_SW1SW2
            writeRecord = retCode
            Exit Function
        End If
    End If

    ' 2. Write data to card
    Call ClearBuffers
    SendBuff(0) = &H80      ' CLA
    SendBuff(1) = &HD2      ' INS
    SendBuff(2) = RecNo     ' P1  Record to be written
    SendBuff(3) = &H0       ' P2
    SendBuff(4) = DataLen   ' P3  Length
    For indx = 0 To DataLen - 1
        SendBuff(indx + 5) = DataIn(indx)
    Next indx
    SendLen = DataLen + 5
    RecvLen = &H2
    retCode = SendAPDU()
    If retCode <> SCARD_S_SUCCESS Then
        Call displayOut(0, retCode, "")
        writeRecord = retCode
        Exit Function
    End If
    tmpStr = ""
    For indx = 0 To 1
        tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
    Next indx
    If tmpStr <> "90 00 " Then
        Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    End If

```

```
    retCode = INVALID_SW1SW2
End If
writeRecord = retCode
End Function
```

```
Private Function SendAPDU() As Long
ioRequest.dwProtocol = dwActProtocol
ioRequest.cbPciLength = Len(ioRequest)
```

```
retCode = SCardTransmit(hCard, _
    ioRequest, _
    SendBuff(0), _
    SendLen, _
    ioRequest, _
    RecvBuff(0), _
    RecvLen)
```

```
SendAPDU = retCode
End Function
```

```
Public Function GetScardErrMsg(ByVal ReturnCode As Long) As String
```

```
Select Case ReturnCode
'Smartcard Reader Errors
Case SCARD_E_CANCELLED
GetScardErrMsg = "The action was canceled by an SCardCancel request."
Case SCARD_E_CANT_DISPOSE
GetScardErrMsg = "The system could not dispose of the media in the requested manner."
Case SCARD_E_CARD_UNSUPPORTED
GetScardErrMsg = "The smart card does not meet minimal requirements for support."
Case SCARD_E_DUPLICATE_READER
GetScardErrMsg = "The reader driver didn't produce a unique reader name."
Case SCARD_E_INSUFFICIENT_BUFFER
GetScardErrMsg = "The data buffer for returned data is too small for the returned data."
Case SCARD_E_INVALID_ATR
GetScardErrMsg = "An ATR string obtained from the registry is not a valid ATR string."
Case SCARD_E_INVALID_HANDLE
GetScardErrMsg = "The supplied handle was invalid."
Case SCARD_E_INVALID_PARAMETER
GetScardErrMsg = "One or more of the supplied parameters could not be properly interpreted."
Case SCARD_E_INVALID_TARGET
GetScardErrMsg = "Registry startup information is missing or invalid."
Case SCARD_E_INVALID_VALUE
GetScardErrMsg = "One or more of the supplied parameter values could not be properly interpreted."
Case SCARD_E_NOT_READY
GetScardErrMsg = "The reader or card is not ready to accept commands."
Case SCARD_E_NOT_TRANSACTED
GetScardErrMsg = "An attempt was made to end a non-existent transaction."
Case SCARD_E_NO_MEMORY
GetScardErrMsg = "Not enough memory available to complete this command."
Case SCARD_E_NO_SERVICE
GetScardErrMsg = "The smart card resource manager is not running."
Case SCARD_E_NO_SMARTCARD
GetScardErrMsg = "The operation requires a smart card, but no smart card is currently in the device."
Case SCARD_E_PCI_TOO_SMALL
GetScardErrMsg = "The PCI receive buffer was too small."
Case SCARD_E_PROTO_MISMATCH
GetScardErrMsg = "The requested protocols are incompatible with the protocol currently in use with the card."
Case SCARD_E_READER_UNAVAILABLE
GetScardErrMsg = "The specified reader is not currently available for use."
Case SCARD_E_READER_UNSUPPORTED
GetScardErrMsg = "The reader driver does not meet minimal requirements for support."
Case SCARD_E_SERVICE_STOPPED
GetScardErrMsg = "The smart card resource manager has shut down."
Case SCARD_E_SHARING_VIOLATION
GetScardErrMsg = "The smart card cannot be accessed because of other outstanding connections."
Case SCARD_E_SYSTEM_CANCELLED
GetScardErrMsg = "The action was canceled by the system, presumably to log off or shut down."
Case SCARD_E_TIMEOUT
```

```

GetScardErrMsg = "The user-specified timeout value has expired."
Case SCARD_E_UNKNOWN_CARD
GetScardErrMsg = "The specified smart card name is not recognized."
Case SCARD_E_UNKNOWN_READER
GetScardErrMsg = "The specified reader name is not recognized."
Case SCARD_F_COMM_ERROR
GetScardErrMsg = "An internal communications error has been detected."
Case SCARD_F_INTERNAL_ERROR
GetScardErrMsg = "An internal consistency check failed."
Case SCARD_F_UNKNOWN_ERROR
GetScardErrMsg = "An internal error has been detected, but the source is unknown."
Case SCARD_F_WAITED_TOO_LONG
GetScardErrMsg = "An internal consistency timer has expired."
Case SCARD_S_SUCCESS
GetScardErrMsg = "No error was encountered."
Case SCARD_W_REMOVED_CARD
GetScardErrMsg = "The smart card has been removed, so that further communication is not possible."
Case SCARD_W_RESET_CARD
GetScardErrMsg = "The smart card has been reset, so any shared state information is invalid."
Case SCARD_W_UNPOWERED_CARD
GetScardErrMsg = "Power has been removed from the smart card, so that further communication is not possible."
Case SCARD_W_UNRESPONSIVE_CARD
GetScardErrMsg = "The smart card is not responding to a reset."
Case SCARD_W_UNSUPPORTED_CARD
GetScardErrMsg = "The reader cannot communicate with the card, due to ATR string configuration conflicts."
Case Else
GetScardErrMsg = "?"
End Select
End Function

```

Form InformasiSaldo

```

Private Sub Form_Activate()
Query "SELECT * FROM Tabungan WHERE No_Rekening = " & Text1.Text & ""
If Not rec.EOF Then
saldo.Caption = rec!saldo
End If
End Sub

```

```

Private Sub Tidak_Click()
Load TerimaKasih: TerimaKasih.Show: Unload Me
End Sub

```

```

Private Sub Ya_Click()
Load AET60Main: AET60Main.Show: Unload Me
End Sub

```

Form JenisPembayaran

```

Private Sub air_Click()
NoPembayaran.jenis.Text = "AIR"
NomorCustomerSalah.jenis.Text = "AIR"
DataPembayaran.jenis.Caption = "PEMBAYARAN AIR"
DataPembayaran.idpmb.Text = "7"
Load NoPembayaran: NoPembayaran.Show: Unload Me
End Sub

```

```

Private Sub krtukrdt_Click()
Load KartuKredit: KartuKredit.Show: Unload Me
End Sub

```

```

Private Sub lstrk_Click()
NoPembayaran.jenis.Text = "LISTRIK"
NomorCustomerSalah.jenis.Text = "LISTRIK"
DataPembayaran.jenis.Caption = "PEMBAYARAN LISTRIK"

```

```
DataPembayaran.idpmb.Text = "8"  
Load NoPembayaran: NoPembayaran.Show: Unload Me  
End Sub
```

```
Private Sub t1prmh_Click()  
NoPembayaran.jenis.Text = "TELEPHONE RUMAH"  
NomorCustomerSalah.jenis.Text = "TELEPHONE RUMAH"  
DataPembayaran.jenis.Caption = "PEMBAYARAN TELEPHONE"  
DataPembayaran.idpmb.Text = "9"  
Load NoPembayaran: NoPembayaran.Show: Unload Me  
End Sub
```

Form JumlahPembayaran

```
Private Sub Benar_Click()  
DataPembayaran.jmlh.Caption = jmlhpmyran.Text  
Load DataPembayaran: DataPembayaran.Show: Unload Me  
End Sub
```

```
Private Sub Form_Activate()  
jmlhpmyran.SetFocus  
End Sub
```

```
Private Sub Salah_Click()  
jmlhpmyran.Text = ""  
jmlhpmyran.SetFocus  
End Sub
```

Form KartuKredit

```
Private Sub bca_Click()  
NoPembayaran.jenis.Text = "BCA"  
NomorCustomerSalah.jenis.Text = "BCA"  
DataPembayaran.jenis.Caption = "KARTU KREDIT BCA"  
DataPembayaran.idpmb.Text = "1"  
Load NoPembayaran: NoPembayaran.Show: Unload Me  
End Sub
```

```
Private Sub bii_Click()  
NoPembayaran.jenis.Text = "BII"  
NomorCustomerSalah.jenis.Text = "BII"  
DataPembayaran.jenis.Caption = "KARTU KREDIT BII"  
DataPembayaran.idpmb.Text = "2"  
Load NoPembayaran: NoPembayaran.Show: Unload Me  
End Sub
```

```
Private Sub city_Click()  
NoPembayaran.jenis.Text = "CITYBANK"  
NomorCustomerSalah.jenis.Text = "CITYBANK"  
DataPembayaran.jenis.Caption = "KARTU KREDIT CITYBANK"  
DataPembayaran.idpmb.Text = "3"  
Load NoPembayaran: NoPembayaran.Show: Unload Me  
End Sub
```

```
Private Sub mega_Click()  
NoPembayaran.jenis.Text = "MEGA"  
NomorCustomerSalah.jenis.Text = "MEGA"  
DataPembayaran.jenis.Caption = "KARTU KREDIT MEGA"  
DataPembayaran.idpmb.Text = "4"  
Load NoPembayaran: NoPembayaran.Show: Unload Me  
End Sub
```

```
Private Sub mndri_Click()  
NoPembayaran.jenis.Text = "MANDIRI"
```

```
NomorCustomerSalah.jenis.Text = "MANDIRI"
DataPembayaran.jenis.Caption = "KARTU KREDIT MANDIRI"
DataPembayaran.idpmb.Text = "5"
Load NoPembayaran: NoPembayaran.Show: Unload Me
End Sub
```

```
Private Sub nisp_Click()
NoPembayaran.jenis.Text = "NISP"
NomorCustomerSalah.jenis.Text = "NISP"
DataPembayaran.jenis.Caption = "KARTU KREDIT NISP"
DataPembayaran.idpmb.Text = "6"
Load NoPembayaran: NoPembayaran.Show: Unload Me
End Sub
```

Form MauTransaksiLain

```
Private Sub Tidak_Click()
Load TerimaKasih: TerimaKasih.Show: Unload Me
End Sub
```

```
Private Sub Ya_Click()
Load AET60Main: AET60Main.Show: Unload Me
End Sub
```

Form Menu

```
Option Explicit
Dim scInit, rdrLocked As Boolean
Dim retCode, numDev, numRdr, hContext, hCard, Protocol, RetLength As Long
Dim dwActProtocol, RecvLen, SendLen As Long
Dim ioRequest As SCARD_IO_REQUEST
Dim FpTemplate(0 To 767) As Byte
Dim SendBuff(0 To 262) As Byte
Dim RecvBuff(0 To 262) As Byte
Dim MatchResult As Integer
Dim apdu As APDURec
Dim Transaction As SCARD_IO_REQUEST
Dim rName, scannerName As String
Dim fpTemplateLength, procTimeOut As Long
Dim rList As String * 256
Dim desFar, desFrr, achFar, achFrr As Long
Const MAX_BUFFER_LEN = 256
Const INVALID_SW1SW2 = -450
Dim cek As Integer
```

```
Private Sub DuaRts_Click()
Dim tmpStr As String
Dim ctr, indx, fileNo, byteNo As Long
Dim tmpTemplate(0 To 767) As Byte

Call ResetInitConnect
'Membaca dan Membandingkan Id Kartu di db dengan di kartu
Dim No_Rekening As String
retCode = readRecord(&H1, &H20)
If retCode <> SCARD_S_SUCCESS Then
Exit Sub
End If
tmpStr = ""
indx = 0
While RecvBuff(indx) <> &H0
tmpStr = tmpStr + Chr(RecvBuff(indx))
indx = indx + 1
Wend
No_Rekening = tmpStr
```



```

Query "SELECT * FROM Tabungan WHERE No_Rekening = " & No_Rekening & ""
If Not rec.EOF Then
    Text2.Text = rec!saldo
End If
Text1.Text = 200000

If Text2.Text <= 200000 Then
Load SaldoSalah: SaldoSalah.Show: Unload Me
Else
If saldoatm.Text <= 200000 Then
Load SaldoAtmHabis: SaldoAtmHabis.Show: Unload Me
Else
Text3.Text = Val(Text2.Text) - Val(Text1.Text)
con.Execute "UPDATE Tabungan SET Saldo=" & _
    "" & Text3.Text & " WHERE No_Rekening=" & No_Rekening & ""

jumlah.Text = Val(saldoatm.Text) - Val(Text1.Text)
con.Execute "UPDATE ATM SET Saldo_ATM=" & _
    "" & jumlah.Text & " WHERE Id_ATM=" & idatm.Text & ""

con.Execute "INSERT INTO Transaksi VALUES (" & penarikan.Text & "," & nomor.Text & "," & idatm.Text & ")"

con.Execute "INSERT INTO Penarikan VALUES (" & penarikan.Text & "," & nomor.Text & "," & _
    "" & tanggal.Text & "," & waktu.Text & "," & 200000 & ")"

SilahkanAmbilUang.Timer5.Interval = 4000
Load SilahkanAmbilUang: SilahkanAmbilUang.Show: Unload Me

Text5.Text = "2"
Timer2.Enabled = True

End If
End If
End Sub

```

```

Private Sub Form_Activate()
Call ResetInitConnect
Query "SELECT * FROM ATM WHERE Lokasi = " & lokasi.Text & ""
If rec.EOF = False Then
    idatm.Text = rec!Id_ATM
    saldoatm.Text = rec!Saldo_ATM
End If

penarikan.Text = "PNRKAN"
tanggal.Text = Date
waktu.Text = Time

Dim a As Integer
Query "SELECT Id_Nomor_Urut FROM Transaksi"
If Not rec.EOF Then
    rec.MoveLast: a = rec!Id_Nomor_Urut
    nomor.Text = a + 1
End If

Call PortOut(888, 0)
cek = 1

End Sub

```

```

Private Sub ResetInitConnect()
'Menon-aktifkan FP scanner
If rdrLocked Then
    retCode = AET60_UnlockReader()
End If
retCode = AET60_Close()

'Menon-aktifkan SC reader

```

```

If scInit Then
    retCode = SCardDisconnect(hCard, SCARD_UNPOWER_CARD)
End If
retCode = SCardReleaseContext(hContext)
scInit = False
rdrLocked = False

Dim indx As Long
rList = String(255, vbNullChar)
numRdr = 255

'Inisialisasi SC Reader
retCode = SCardEstablishContext(SCARD_SCOPE_USER, 0, 0, hContext)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If

retCode = SCardListReaders(hContext, rName, rList, numRdr)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If
Call LoadListToControl(cbReader, rList)
cbReader.ListIndex = 0

'Inisialisasi FP Scanner
retCode = AET60_Open()
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If
numDev = AET60_GetNumDevices()
If numDev > 0 Then
    cbScanner.Clear
    scannerName = String(255, vbNullChar)
    RetLength = Len(scannerName)
    For indx = 0 To (numDev - 1)
        retCode = AET60_GetDeviceName(indx, _
            scannerName, _
            RetLength)
        cbScanner.AddItem scannerName
        cbScanner.ListIndex = 0
    Next indx
Else
    Call displayOut(1, retCode, "")
    Exit Sub
End If

    Call displayOut(2, 0, "")
Call displayOut(3, 0, "")

' 1. Connect to smartcard
retCode = SCardConnect(hContext, _
    cbReader.Text, _
    SCARD_SHARE_SHARED, _
    SCARD_PROTOCOL_T0 Or SCARD_PROTOCOL_T1, _
    hCard, _
    Protocol)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If
scInit = True

' 2. Lock scanner and set callback
retCode = AET60_LockReader(cbScanner.ListIndex)

```

```

If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If
rdrLocked = True

retCode = AET60_SetCallback(AddressOf CALLBACK_FUNCTION)
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If

```

```

Dim tmpStr As String
Dim ctr, fileNo, byteNo As Long
Dim tmpTemplate(0 To 767) As Byte

```

```

Call clearMessages

```

```

'Menampilkan informasi user yang terdapat pada kartu
'Pilih file BB BB
retCode = selectFile(&HBB, &HBB)
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "91 01 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    Exit Sub
End If

```

```

End Sub

```

```

Private Sub clearMessages()
    Call displayOut(2, 0, "")
    Call displayOut(3, 0, "")
    userAction.Refresh
    MsgBox.Refresh
End Sub

```

```

Private Sub ClearBuffers()
    Dim indx As Long
    For indx = 0 To 262
        SendBuff(indx) = &H0
        RecvBuff(indx) = &H0
    Next indx
End Sub

```

```

Private Sub displayOut(ByVal errType As Integer, ByVal retVal As Long, ByVal msgText As String)
    Select Case (errType)
        Case 0 ' Smartcard error
            userAction.Text = "Error encountered. Please click on Reset button."
            MsgBox.Text = GetScardErrMsg(retVal)
        Case 1 ' BioAPI error
            Dim StatusMsg As String
            StatusMsg = String(255, vbNullChar)
            userAction.Text = "Error encountered. Please click on Reset button."
            Call AET60_GetStatus(retVal, StatusMsg)
            MsgBox.Text = StatusMsg
        Case 2 ' Message only
            MsgBox.Text = msgText
        Case 3 ' Required user action
            userAction.Text = msgText
    End Select

```

End Sub

Private Function validIDCard() As Boolean

Dim tmpStr As String

Dim indx As Integer

validIDCard = False

' 1. Select CC CC

retCode = selectFile(&HCC, &HCC)

If retCode <> SCARD_S_SUCCESS Then

Exit Function

End If

tmpStr = ""

For indx = 0 To 1

tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "

Next indx

If tmpStr <> "91 02 " Then

Call displayOut(2, 0, "Inserted ID card may be invalid. Return value: " + tmpStr)

Exit Function

End If

' 2. Read ID Flag

retCode = readRecord(&H0, &H6)

If retCode <> SCARD_S_SUCCESS Then

Exit Function

End If

tmpStr = ""

For indx = 0 To 3

tmpStr = tmpStr + Chr(RecvBuff(indx + 2))

Next indx

If tmpStr <> "KNEF" Then

Call displayOut(2, 0, "The user is not using a valid ID card.")

Exit Function

End If

validIDCard = True

End Function

Private Function SubmitIC() As Long

Dim indx As Integer

Dim tmpStr As String

Call ClearBuffers

SendBuff(0) = &H80 ' CLA

SendBuff(1) = &H20 ' INS

SendBuff(2) = &H7 ' P1

SendBuff(3) = &H0 ' P2

SendBuff(4) = &H8 ' P3

SendBuff(5) = &H41 ' A

SendBuff(6) = &H43 ' C

SendBuff(7) = &H4F ' O

SendBuff(8) = &H53 ' S

SendBuff(9) = &H54 ' T

SendBuff(10) = &H45 ' E

SendBuff(11) = &H53 ' S

SendBuff(12) = &H54 ' T

SendLen = &HD

RecvLen = &H2

retCode = SendAPDU()

If retCode <> SCARD_S_SUCCESS Then

Call displayOut(0, retCode, "")

SubmitIC = retCode

Exit Function

End If

tmpStr = ""

For indx = 0 To 1

tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "

Next indx

```

If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    retCode = INVALID_SW1SW2
End If
SubmitIC = retCode
End Function

```

Private Function selectFile(ByVal HiAddr As Byte, ByVal LoAddr As Byte) As Long

```

Call ClearBuffers
SendBuff(0) = &H80 ' CLA
SendBuff(1) = &HA4 ' INS
SendBuff(2) = &H0 ' P1
SendBuff(3) = &H0 ' P2
SendBuff(4) = &H2 ' P3
SendBuff(5) = HiAddr ' Value of High Byte
SendBuff(6) = LoAddr ' Value of Low Byte
SendLen = &H7
RecvLen = &H2
retCode = SendAPDU()
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
End If
selectFile = retCode
End Function

```

Private Function readRecord(ByVal RecNo As Byte, ByVal DataLen As Byte) As Long

```

Dim indx As Integer
Dim tmpStr As String

Call ClearBuffers
SendBuff(0) = &H80 ' CLA
SendBuff(1) = &HB2 ' INS
SendBuff(2) = RecNo ' P1 Record No
SendBuff(3) = &H0 ' P2
SendBuff(4) = DataLen ' P3 Length of data to be read
SendLen = &H5
RecvLen = SendBuff(4) + 2
retCode = SendAPDU()
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    readRecord = retCode
    Exit Function
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx + SendBuff(4))), 2) + " "
Next indx
If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    retCode = INVALID_SW1SW2
End If
readRecord = retCode
End Function

```

Private Function writeRecord(ByVal caseType As Integer, ByVal RecNo As Byte, _
ByVal maxDataLen As Byte, ByVal DataLen As Byte, _
ByRef DataIn() As Byte) As Long

```

Dim tmpStr As String
Dim indx As Integer

If caseType = 1 Then
    ' 1. Re-initialize card values to $00
    Call ClearBuffers
    SendBuff(0) = &H80 ' CLA
    SendBuff(1) = &HD2 ' INS
    SendBuff(2) = RecNo ' P1 Record to be written

```

```

SendBuff(3) = &H0      ' P2
SendBuff(4) = maxDataLen ' P3 Length
For indx = 0 To maxDataLen - 1
    SendBuff(indx + 5) = &H0
Next indx
SendLen = maxDataLen + 5
RecvLen = &H2
retCode = SendAPDU()
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    writeRecord = retCode
    Exit Function
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    retCode = INVALID_SW1SW2
    writeRecord = retCode
    Exit Function
End If
End If

```

```

' 2. Write data to card
    Call ClearBuffers
SendBuff(0) = &H80      ' CLA
SendBuff(1) = &HD2      ' INS
SendBuff(2) = RecNo     ' P1 Record to be written
SendBuff(3) = &H0      ' P2
SendBuff(4) = DataLen   ' P3 Length
For indx = 0 To DataLen - 1
    SendBuff(indx + 5) = DataIn(indx)
Next indx
SendLen = DataLen + 5
RecvLen = &H2
retCode = SendAPDU()
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    writeRecord = retCode
    Exit Function
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    retCode = INVALID_SW1SW2
End If
writeRecord = retCode
End Function

```

```

Private Function SendAPDU() As Long
    ioRequest.dwProtocol = dwActProtocol
    ioRequest.cbPciLength = Len(ioRequest)

```

```

    retCode = SCardTransmit(hCard, _
        ioRequest, _
        SendBuff(0), _
        SendLen, _
        ioRequest, _
        RecvBuff(0), _
        RecvLen)

```

```

SendAPDU = retCode
End Function

```

```

Public Function GetScardErrMsg(ByVal ReturnCode As Long) As String
Select Case ReturnCode
'Smartcard Reader Errors
Case SCARD_E_CANCELLED
GetScardErrMsg = "The action was canceled by an SCardCancel request."
Case SCARD_E_CANT_DISPOSE
GetScardErrMsg = "The system could not dispose of the media in the requested manner."
Case SCARD_E_CARD_UNSUPPORTED
GetScardErrMsg = "The smart card does not meet minimal requirements for support."
Case SCARD_E_DUPLICATE_READER
GetScardErrMsg = "The reader driver didn't produce a unique reader name."
Case SCARD_E_INSUFFICIENT_BUFFER
GetScardErrMsg = "The data buffer for returned data is too small for the returned data."
Case SCARD_E_INVALID_ATR
GetScardErrMsg = "An ATR string obtained from the registry is not a valid ATR string."
Case SCARD_E_INVALID_HANDLE
GetScardErrMsg = "The supplied handle was invalid."
Case SCARD_E_INVALID_PARAMETER
GetScardErrMsg = "One or more of the supplied parameters could not be properly interpreted."
Case SCARD_E_INVALID_TARGET
GetScardErrMsg = "Registry startup information is missing or invalid."
Case SCARD_E_INVALID_VALUE
GetScardErrMsg = "One or more of the supplied parameter values could not be properly interpreted."
Case SCARD_E_NOT_READY
GetScardErrMsg = "The reader or card is not ready to accept commands."
Case SCARD_E_NOT_TRANSACTED
GetScardErrMsg = "An attempt was made to end a non-existent transaction."
Case SCARD_E_NO_MEMORY
GetScardErrMsg = "Not enough memory available to complete this command."
Case SCARD_E_NO_SERVICE
GetScardErrMsg = "The smart card resource manager is not running."
Case SCARD_E_NO_SMARTCARD
GetScardErrMsg = "The operation requires a smart card, but no smart card is currently in the device."
Case SCARD_E_PCI_TOO_SMALL
GetScardErrMsg = "The PCI receive buffer was too small."
Case SCARD_E_PROTO_MISMATCH
GetScardErrMsg = "The requested protocols are incompatible with the protocol currently in use with the card."
Case SCARD_E_READER_UNAVAILABLE
GetScardErrMsg = "The specified reader is not currently available for use."
Case SCARD_E_READER_UNSUPPORTED
GetScardErrMsg = "The reader driver does not meet minimal requirements for support."
Case SCARD_E_SERVICE_STOPPED
GetScardErrMsg = "The smart card resource manager has shut down."
Case SCARD_E_SHARING_VIOLATION
GetScardErrMsg = "The smart card cannot be accessed because of other outstanding connections."
Case SCARD_E_SYSTEM_CANCELLED
GetScardErrMsg = "The action was canceled by the system, presumably to log off or shut down."
Case SCARD_E_TIMEOUT
GetScardErrMsg = "The user-specified timeout value has expired."
Case SCARD_E_UNKNOWN_CARD
GetScardErrMsg = "The specified smart card name is not recognized."
Case SCARD_E_UNKNOWN_READER
GetScardErrMsg = "The specified reader name is not recognized."
Case SCARD_F_COMM_ERROR
GetScardErrMsg = "An internal communications error has been detected."
Case SCARD_F_INTERNAL_ERROR
GetScardErrMsg = "An internal consistency check failed."
Case SCARD_F_UNKNOWN_ERROR
GetScardErrMsg = "An internal error has been detected, but the source is unknown."
Case SCARD_F_WAITED_TOO_LONG
GetScardErrMsg = "An internal consistency timer has expired."
Case SCARD_S_SUCCESS
GetScardErrMsg = "No error was encountered."
Case SCARD_W_REMOVED_CARD
GetScardErrMsg = "The smart card has been removed, so that further communication is not possible."
Case SCARD_W_RESET_CARD

```

```

GetScardErrMsg = "The smart card has been reset, so any shared state information is invalid."
Case SCARD_W_UNPOWERED_CARD
GetScardErrMsg = "Power has been removed from the smart card, so that further communication is not possible."
Case SCARD_W_UNRESPONSIVE_CARD
GetScardErrMsg = "The smart card is not responding to a reset."
Case SCARD_W_UNSUPPORTED_CARD
GetScardErrMsg = "The reader cannot communicate with the card, due to ATR string configuration conflicts."
Case Else
GetScardErrMsg = "?"
End Select
End Function

```

```

Private Sub LmRts_Click()
Dim tmpStr As String
Dim ctr, indx, fileNo, byteNo As Long
Dim tmpTemplate(0 To 767) As Byte

Call ResetInitConnect
Membaca dan Membandingkan Id Kartu di db dengan di kartu
Dim No_Rekening As String
retCode = readRecord(&H1, &H20)
If retCode <> SCARD_S_SUCCESS Then
Exit Sub
End If
tmpStr = ""
indx = 0
While RecvBuff(indx) <> &H0
tmpStr = tmpStr + Chr(RecvBuff(indx))
indx = indx + 1
Wend
No_Rekening = tmpStr

Query "SELECT * FROM Tabungan WHERE No_Rekening = " & No_Rekening & ""
If Not rec.EOF Then
Text2.Text = rec!saldo
End If
Text1.Text = 500000

If Text2.Text <= 500000 Then
Load SaldoSalah: SaldoSalah.Show: Unload Me
Else
If saldoatm.Text <= 500000 Then
Load SaldoAtmHabis: SaldoAtmHabis.Show: Unload Me
Else
Text3.Text = Val(Text2.Text) - Val(Text1.Text)
con.Execute "UPDATE Tabungan SET Saldo=" & _
"" & Text3.Text & " WHERE No_Rekening=" & No_Rekening & ""

jumlah.Text = Val(saldoatm.Text) - Val(Text1.Text)
con.Execute "UPDATE ATM SET Saldo_ATM=" & _
"" & jumlah.Text & " WHERE Id_ATM=" & idatm.Text & ""

con.Execute "INSERT INTO Transaksi VALUES (" & penarikan.Text & "," & nomor.Text & "," & idatm.Text & ")"

con.Execute "INSERT INTO Penarikan VALUES (" & penarikan.Text & "," & nomor.Text & "," & _
"" & tanggal.Text & "," & waktu.Text & "," & 500000 & ")"

SilahkanAmbilUang.Timer5.Interval = 12000
Load SilahkanAmbilUang: SilahkanAmbilUang.Show: Unload Me

Text5.Text = "5"
Timer2.Enabled = True

End If
End If
End Sub

```

```

Private Sub Srts_Click()
Dim tmpStr As String
Dim ctr, indx, fileNo, byteNo, proc As Long
Dim tmpTemplate(0 To 767) As Byte

Call ResetInitConnect

Membaca dan Membandingkan Id Kartu di db dengan di kartu
Dim No_Rekening As String
retCode = readRecord(&H1, &H20)
If retCode <> SCARD_S_SUCCESS Then
Exit Sub
End If
tmpStr = ""
indx = 0
While RecvBuff(indx) <> &H0
tmpStr = tmpStr + Chr(RecvBuff(indx))
indx = indx + 1
Wend
No_Rekening = tmpStr

Query "SELECT * FROM Tabungan WHERE No_Rekening = " & No_Rekening & ""
If Not rec.EOF Then
Text2.Text = rec!saldo
End If
Text1.Text = 100000

If Text2.Text <= 100000 Then
Load SaldoSalah: SaldoSalah.Show: Unload Me
Else
If saldoatm.Text <= 100000 Then
Load SaldoAtmHabis: SaldoAtmHabis.Show: Unload Me
Else
Text3.Text = Val(Text2.Text) - Val(Text1.Text)
con.Execute "UPDATE Tabungan SET Saldo=" & _
"" & Text3.Text & " WHERE No_Rekening=" & No_Rekening & ""

jumlah.Text = Val(saldoatm.Text) - Val(Text1.Text)
con.Execute "UPDATE ATM SET Saldo_ATM=" & _
"" & jumlah.Text & " WHERE Id_ATM=" & idatm.Text & ""

con.Execute "INSERT INTO Transaksi VALUES (" & penarikan.Text & "," & nomor.Text & "," & idatm.Text & ")"

con.Execute "INSERT INTO Penarikan VALUES (" & penarikan.Text & "," & nomor.Text & "," & _
"" & tanggal.Text & "," & waktu.Text & "," & 100000 & ")"

SilahkanAmbilUang.Timer5.Interval = 2000
Load SilahkanAmbilUang: SilahkanAmbilUang.Show: Unload Me

Text5.Text = "1"
Timer2.Enabled = True

End If
End If

```

End Sub

```

Private Sub StJt_Click()
Dim tmpStr As String
Dim ctr, indx, fileNo, byteNo As Long
Dim tmpTemplate(0 To 767) As Byte

Call ResetInitConnect
Membaca dan Membandingkan Id Kartu di db dengan di kartu
Dim No_Rekening As String

```

```

retCode = readRecord(&H1, &H20)
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
End If
tmpStr = ""
indx = 0
While RecvBuff(indx) <> &H0
    tmpStr = tmpStr + Chr(RecvBuff(indx))
    indx = indx + 1
Wend
No_Rekening = tmpStr

Query "SELECT * FROM Tabungan WHERE No_Rekening = " & No_Rekening & ""
If Not rec.EOF Then
    Text2.Text = rec!saldo
End If
Text1.Text = 1000000

If Text2.Text <= 1000000 Then
    Load SaldoSalah: SaldoSalah.Show: Unload Me
Else
    If saldoatm.Text <= 1000000 Then
        Load SaldoAtmHabis: SaldoAtmHabis.Show: Unload Me
    Else
        Text3.Text = Val(Text2.Text) - Val(Text1.Text)
        con.Execute "UPDATE Tabungan SET Saldo=" & _
            "" & Text3.Text & " WHERE No_Rekening=" & No_Rekening & ""

        jumlah.Text = Val(saldoatm.Text) - Val(Text1.Text)
        con.Execute "UPDATE ATM SET Saldo_ATM=" & _
            "" & jumlah.Text & " WHERE Id_ATM=" & idatm.Text & ""

        con.Execute "INSERT INTO Transaksi VALUES (" & penarikan.Text & "," & nomor.Text & "," & idatm.Text & ")"

        con.Execute "INSERT INTO Penarikan VALUES (" & penarikan.Text & "," & nomor.Text & "," & _
            "" & tanggal.Text & "," & waktu.Text & "," & 1000000 & ")"

        SilahkanAmbilUang.Timer5.Interval = 24000
        Load SilahkanAmbilUang: SilahkanAmbilUang.Show: Unload Me

        Text5.Text = "10"
        Timer2.Enabled = True

    End If
End If
End Sub

```

```

Private Sub StJtDuaRts_Click()
Dim tmpStr As String
Dim ctr, indx, fileNo, byteNo As Long
Dim tmpTemplate(0 To 767) As Byte

Call ResetInitConnect
Membaca dan Membandingkan Id Kartu di db dengan di kartu
Dim No_Rekening As String
retCode = readRecord(&H1, &H20)
If retCode <> SCARD_S_SUCCESS Then
    Exit Sub
End If
tmpStr = ""
indx = 0
While RecvBuff(indx) <> &H0
    tmpStr = tmpStr + Chr(RecvBuff(indx))
    indx = indx + 1
Wend
No_Rekening = tmpStr

```

```

Query "SELECT * FROM Tabungan WHERE No_Rekening = " & No_Rekening & ""
If Not rec.EOF Then
    Text2.Text = rec!saldo
End If
Text1.Text = 1500000

If Text2.Text <= 1500000 Then
Load SaldoSalah: SaldoSalah.Show: Unload Me
Else
If saldoatm.Text <= 1500000 Then
Load SaldoAtmHabis: SaldoAtmHabis.Show: Unload Me
Else
Text3.Text = Val(Text2.Text) - Val(Text1.Text)
con.Execute "UPDATE Tabungan SET Saldo=" & _
    "" & Text3.Text & " WHERE No_Rekening=" & No_Rekening & ""

jumlah.Text = Val(saldoatm.Text) - Val(Text1.Text)
con.Execute "UPDATE ATM SET Saldo_ATM=" & _
    "" & jumlah.Text & " WHERE Id_ATM=" & idatm.Text & ""

con.Execute "INSERT INTO Transaksi VALUES (" & penarikan.Text & "," & nomor.Text & "," & idatm.Text & ")"

con.Execute "INSERT INTO Penarikan VALUES (" & penarikan.Text & "," & nomor.Text & "," & _
    "" & tanggal.Text & "," & waktu.Text & "," & 1500000 & ")"

SilahkanAmbilUang.Timer5.Interval = 36000
Load SilahkanAmbilUang: SilahkanAmbilUang.Show: Unload Me

Text5.Text = "15"
Timer2.Enabled = True

End If
End If
End Sub

```

```

Private Sub TgRts_Click()
Dim tmpStr As String
Dim ctr, indx, fileNo, byteNo As Long
Dim tmpTemplate(0 To 767) As Byte

Call ResetInitConnect
'Membaca dan Membandingkan Id Kartu di db dengan di kartu
Dim No_Rekening As String
retCode = readRecord(&H1, &H20)
If retCode <> SCARD_S_SUCCESS Then
Exit Sub
End If
tmpStr = ""
indx = 0
While RecvBuff(indx) <> &H0
tmpStr = tmpStr + Chr(RecvBuff(indx))
indx = indx + 1
Wend
No_Rekening = tmpStr
Text4.Text = tmpStr

Query "SELECT * FROM Tabungan WHERE No_Rekening = " & No_Rekening & ""
If Not rec.EOF Then
    Text2.Text = rec!saldo
End If
Text1.Text = 300000

If Text2.Text <= 300000 Then
Load SaldoSalah: SaldoSalah.Show: Unload Me
Else
If saldoatm.Text <= 300000 Then
Load SaldoAtmHabis: SaldoAtmHabis.Show: Unload Me

```

```

Else
Text3.Text = Val(Text2.Text) - Val(Text1.Text)
con.Execute "UPDATE Tabungan SET Saldo=" & _
    "" & Text3.Text & " WHERE No_Rekening=" & No_Rekening & ""

jumlah.Text = Val(saldoatm.Text) - Val(Text1.Text)
con.Execute "UPDATE ATM SET Saldo_ATM=" & _
    "" & jumlah.Text & " WHERE Id_ATM=" & idatm.Text & ""

con.Execute "INSERT INTO Transaksi VALUES (" & penarikan.Text & "," & nomor.Text & "," & idatm.Text & ")"

con.Execute "INSERT INTO Penarikan VALUES (" & penarikan.Text & "," & nomor.Text & "," & _
    "" & tanggal.Text & "," & waktu.Text & "," & 300000 & ")"

SilahkanAmbilUang.Timer5.Interval = 6000
Load SilahkanAmbilUang: SilahkanAmbilUang.Show: Unload Me

Text5.Text = "3"
Timer2.Enabled = True

End If
End If
End Sub

```

```

Private Sub Timer1_Timer()
Timer1.Enabled = False
Call PortOut(888, 0)
Timer3.Enabled = True
End Sub

```

```

Private Sub Timer2_Timer()
If Text5.Text > 0 And cek = 1 Then
Text5.Text = Text5.Text - 1
Timer1.Enabled = True
cek = 0
Call PortOut(888, 255)
ElseIf Text5.Text = 0 Then
Timer2.Enabled = False
End If
End Sub

```

```

Private Sub Timer3_Timer()
cek = 1
Timer3.Enabled = False
End Sub

```

```

Private Sub TrnsksiLain_Click()
Dim tmpStr As String
Dim ctr, indx, fileNo, byteNo As Long
Dim tmpTemplate(0 To 767) As Byte

```

```

Call ResetInitConnect
'Membaca dan Membandingkan Id Kartu di db dengan di kartu
Dim No_Rekening As String
retCode = readRecord(&H1, &H20)
If retCode <> SCARD_S_SUCCESS Then
Exit Sub
End If
tmpStr = ""
indx = 0
While RecvBuff(indx) <> &H0
tmpStr = tmpStr + Chr(RecvBuff(indx))
indx = indx + 1
Wend
Text4.Text = tmpStr

```

```
InformasiSaldo.Text1.Text = Text4.Text
PenarikanTunai.Text2.Text = Text4.Text
SilahkanMasukkanNominal.Text5.Text = Text4.Text
DataIsiPulsa.rekening.Text = Text4.Text
DataPembayaran.rekening.Text = Text4.Text
Load TransaksiLain: TransaksiLain.Show: Unload Me
End Sub
```

Form NoHandphone

```
Private Sub Benar_Click()
DataIsiPulsa.nohandphone.Caption = handphone.Text
Load DataIsiPulsa: DataIsiPulsa.Show: Unload Me
End Sub
```

```
Private Sub Form_Activate()
handphone.SetFocus
End Sub
```

```
Private Sub Salah_Click()
handphone.Text = ""
handphone.SetFocus
End Sub
```

Form NomorCustomerSalah

```
Private Sub Form_Deactivate()
Timer1.Enabled = False
End Sub
```

```
Private Sub Timer1_Timer()
NoPembayaran.jenis.Text = jenis.Text
Load NoPembayaran: NoPembayaran.Show: Unload Me
End Sub
```

Form NomorSalah

```
Private Sub Timer1_Timer()
Load nohandphone: nohandphone.Show: Unload Me
End Sub
```

Form NoPembayaran

```
Private Sub Benar_Click()
Query "SELECT * FROM Pembayaran_Air WHERE No_Customer = " & cstmer.Text & ""
If Not rec.EOF Then
nama.Text = rec!Nama_Customer
jenis2.Text = rec!Pembayaran
End If
```

```
Query "SELECT * FROM Pembayaran_Listrik WHERE No_Customer = " & cstmer.Text & ""
If Not rec.EOF Then
nama.Text = rec!Nama_Customer
jenis2.Text = rec!Pembayaran
End If
```

```
Query "SELECT * FROM Pembayaran_Telephone WHERE No_Customer = " & cstmer.Text & ""
If Not rec.EOF Then
nama.Text = rec!Nama_Customer
jenis2.Text = rec!Pembayaran
End If
```

```
Query "SELECT * FROM Pembayaran_Kartu_Kredit WHERE No_Customer = " & cstmer.Text & ""
If Not rec.EOF Then
nama.Text = rec!Nama_Customer
```

```
jenis2.Text = rec!Bank
End If
```

```
If nama.Text = "" Then
NomorCustomerSalah.jenis.Text = jenis.Text
Load NomorCustomerSalah: NomorCustomerSalah.Show: Unload Me
Else
If jenis.Text <> jenis2.Text Then
NomorCustomerSalah.jenis.Text = jenis.Text
Load NomorCustomerSalah: NomorCustomerSalah.Show: Unload Me
Else
DataPembayaran.nocustomer.Caption = cstmer.Text
DataPembayaran.nama.Caption = nama.Text
Load JumlahPembayaran: JumlahPembayaran.Show: Unload Me
End If
End If
```

```
End Sub
```

```
Private Sub Form_Activate()
cstmer.SetFocus
End Sub
```

```
Private Sub Salah_Click()
cstmer.Text = ""
cstmer.SetFocus
End Sub
```

```
Form PenarikanTunai
```

```
Private Type RECT
Left As Long
Top As Long
Right As Long
Bottom As Long
End Type
Dim DefaultPrinterScaleMode As Integer
Private Const DT_WORDBREAK = &H10
Private Declare Function DrawText Lib "user32" Alias "DrawTextA" (ByVal hDC As Long, ByVal lpStr As String, ByVal nCount As Long, lpRect As RECT, ByVal wFormat As Long) As Long
Private Declare Function GetDesktopWindow Lib "user32" () As Long
Private Declare Function ShellExecute Lib "shell32.dll" Alias "ShellExecuteA" _
(ByVal hwnd As Long, ByVal lpzOp As String, _
ByVal lpzFile As String, ByVal lpzParams As String, _
ByVal lpzDir As String, ByVal FsShowCmd As Long) _
As Long
Dim cek As Integer
```

```
Private Sub Benar_Click()
Query "SELECT * FROM Penarikan_Tunai WHERE Id = " & Text1.Text & ""
If rec.EOF = False Then
tunai.Text = rec!Saldo_Penarikan
SilahkanAmbilUang.Text1.Text = rec!Saldo_Penarikan
'SilahkanTunggu.Text2.Text = rec!led
End If

If Text1.Text = tunai.Text Then
Query "SELECT * FROM Tabungan WHERE No_Rekening = " & Text2.Text & ""
If Not rec.EOF Then
Text3.Text = rec!saldo
End If

If Val(Text3.Text) <= Val(Text1.Text) Then
Load SaldoSalah: SaldoSalah.Show: Unload Me
Else
If Val(saldoatm.Text) <= Val(Text1.Text) Then
```

```

Load SaldoAtmHabis: SaldoAtmHabis.Show: Unload Me
Else
Text4.Text = Val(Text3.Text) - Val(Text1.Text)
saldoatm.Text = Val(saldoatm.Text) - Val(Text1.Text)
jumlah.Text = "JUMLAH    : " & Text1.Text
saldo.Text = "SALDO      : " & Text4.Text

Dim sngNextYCor, sngNextXCor As Single
sngNextYCor = PrintBoxedText(jdl.Text, CSng(10), CSng(10), CSng(100), CSng(300))

sngNextXCor = PrintBoxedText(tgl.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(jam.Text, sngNextXCor + 15, sngNextYCor + 1, CSng(100), CSng(300))

sngNextXCor = PrintBoxedText(id.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(lokasi.Text, sngNextXCor + 5, sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(jenis.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(jumlah.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

sngNextYCor = PrintBoxedText(saldo.Text, CSng(10), sngNextYCor + 1, CSng(100), CSng(300))

Printer.EndDoc

con.Execute "UPDATE Tabungan SET Saldo=" & _
    "" & Text4.Text & " WHERE No_Rekening=" & Text2.Text & ""

con.Execute "UPDATE ATM SET Saldo_ATM=" & _
    "" & saldoatm.Text & " WHERE Id_ATM=" & id.Text & ""

con.Execute "INSERT INTO Transaksi VALUES (" & penarikan.Text & "," & nomor.Text & "," & id.Text & ")"

con.Execute "INSERT INTO Penarikan VALUES (" & penarikan.Text & "," & nomor.Text & "," & _
    "" & tgl.Text & "," & jam.Text & "," & Text1.Text & ")"

SilahkanAmbilUang.Timer2.Enabled = True
Load SilahkanAmbilUang: SilahkanAmbilUang.Show: Unload Me

End If
End If
Else
SalahSaldo.Text1.Text = Val(Text2.Text)
Load SalahSaldo: SalahSaldo.Show: Unload Me
End If

End Sub

```

```

Private Sub Form_Activate()
Text1.SetFocus
jdl.Text = "    ** ATM ADRIAN **    "
tgl.Text = "" & Date
jam.Text = "" & Time
lokasi.Text = "MARANATHA"
jenis.Text = "PENARIKAN"

Query "SELECT * FROM ATM WHERE Lokasi = " & lokasi.Text & ""
If rec.EOF = False Then
id.Text = rec!Id_ATM
saldoatm.Text = rec!Saldo_ATM
End If

Dim a As Integer
Query "SELECT * FROM Transaksi WHERE Id_Tipe = " & penarikan.Text & ""
If Not rec.EOF Then

```

```
rec.MoveLast: a = rec!Id_Nomor_Urut
nomor.Text = a + 1
End If
```

```
DefaultPrinterScaleMode = vbMillimeters ' Set default scalemode on printer to Millimeters
End Sub
```

```
Private Sub Salah_Click()
Text1.Text = ""
Text1.SetFocus
End Sub
```

```
Function PrintBoxedText(strTextToPrint As String, sngX As Single, sngY As Single, sngMaxWith As Single,
sngMaxHeight As Single) As Single
Dim rc As RECT
Dim lngXpos As Long, lngYpos As Long, lngWdt As Long, heightused As Long, lngMaxHgt As Long
Dim hPrintDc As Long
Printer.ScaleMode = DefaultPrinterScaleMode
```

```
hPrintDc = Printer.hDC
lngWdt = Printer.ScaleX(sngMaxWith, DefaultPrinterScaleMode, vbPixels)
lngXpos = Printer.ScaleX(sngX, DefaultPrinterScaleMode, vbPixels)
lngYpos = Printer.ScaleY(sngY, DefaultPrinterScaleMode, vbPixels)
lngMaxHgt = Printer.ScaleY(sngMaxHeight, DefaultPrinterScaleMode, vbPixels)
rc.Left = lngXpos
rc.Right = lngWdt + rc.Left
rc.Top = lngYpos
rc.Bottom = lngMaxHgt + rc.Top
'Fool the printer to make it believe that this is a standard printjob.
Printer.Print " "
```

```
heightused = DrawText(hPrintDc, strTextToPrint, -1, rc, DT_WORDBREAK)
```

```
PrintBoxedText = Printer.ScaleY(heightused, vbPixels, DefaultPrinterScaleMode) + sngY
```

```
End Function
```

```
Form PulsaIsi
```

```
Private Sub dua_Click()
DataIsiPulsa.jumlah.Caption = "200000"
Load nohandphone: nohandphone.Show: Unload Me
End Sub
```

```
Private Sub dualima_Click()
DataIsiPulsa.jumlah.Caption = "250000"
Load nohandphone: nohandphone.Show: Unload Me
End Sub
```

```
Private Sub lima_Click()
DataIsiPulsa.jumlah.Caption = "500000"
Load nohandphone: nohandphone.Show: Unload Me
End Sub
```

```
Private Sub seratus_Click()
DataIsiPulsa.jumlah.Caption = "1000000"
Load nohandphone: nohandphone.Show: Unload Me
End Sub
```

```
Form Salah
```

```
Private Sub Timer1_Timer()
Load transfer: transfer.Show: Unload Me
End Sub
```

Form SalahSaldo

```
Private Sub Timer1_Timer()  
PenarikanTunai.Text2.Text = Val(Text1.Text)  
Load PenarikanTunai: PenarikanTunai.Show: Unload Me  
End Sub
```

Form SaldoAtmHabis

```
Private Sub Timer1_Timer()  
PenarikanTunai.Text2.Text = Val(Text1.Text)  
Load PenarikanTunai: PenarikanTunai.Show: Unload Me  
End Sub
```

Form SaldoSalah

```
Private Sub Tidak_Click()  
Load TerimaKasih: TerimaKasih.Show: Unload Me  
End Sub
```

```
Private Sub Ya_Click()  
Load AET60Main: AET60Main.Show: Unload Me  
End Sub
```

Form SilahkanAmbilUang

Dim cek As Integer

```
Private Sub Form_Load()  
Query "SELECT * FROM Penarikan_Tunai WHERE Id = '" & PenarikanTunai.Text1.Text & """  
    If rec.EOF = False Then  
        Text2.Text = rec!led  
    End If  
  
    Call PortOut(888, 0)  
    cek = 1
```

End Sub

```
Private Sub Timer1_Timer()  
Timer1.Enabled = False  
Call PortOut(888, 0)  
Timer3.Enabled = True  
End Sub
```

```
Private Sub Timer2_Timer()  
If Text2.Text > 0 And cek = 1 Then  
Text2.Text = Text2.Text - 1  
Timer1.Enabled = True  
cek = 0  
Call PortOut(888, 255)  
ElseIf Text2.Text = 0 Then  
Timer2.Enabled = False  
Timer4.Enabled = True  
End If  
End Sub
```

```
Private Sub Timer3_Timer()  
cek = 1  
Timer3.Enabled = False  
End Sub
```

```
Private Sub Timer4_Timer()
```

```
Load MauTransaksiLain: MauTransaksiLain.Show: Unload Me
End Sub
```

```
Private Sub Timer5_Timer()
Load TerimaKasih: TerimaKasih.Show: Unload Me
End Sub
```

Form SilahkanMasukkanNominal

```
Private Sub Benar_Click()
Dim nama, IdNasabah As String
If Text5.Text = Text2.Text Then
Load Salah: Salah.Show: Unload Me
Else
Query "SELECT * FROM Tabungan WHERE No_Rekening = " & Text5.Text & ""
If Not rec.EOF Then
Text6.Text = rec!saldo
End If
Text7.Text = Val(Text6.Text) - Val(Text1.Text)
```

```
DataTransfer.Text1.Text = Text7.Text
DataTransfer.Text5.Text = Text6.Text
```

```
Query "SELECT * FROM Tabungan WHERE No_Rekening = " & Text2.Text & ""
If Not rec.EOF Then
Text3.Text = rec!saldo
IdNasabah = rec!Id_Nasabah
End If
Text4.Text = Val(Text3.Text) + Val(Text1.Text)
```

```
DataTransfer.Text4.Text = Text4.Text
```

```
Query "SELECT * FROM Nasabah WHERE Id_Nasabah = " & IdNasabah & ""
If Not rec.EOF Then
nama = rec!nama
End If
```

```
DataTransfer.Label4.Caption = nama
DataTransfer.Label5.Caption = Text2.Text
DataTransfer.Label6.Caption = Text1.Text
Load DataTransfer: DataTransfer.Show: Unload Me
End If
```

```
End Sub
```

```
Private Sub Form_Activate()
Text1.SetFocus
End Sub
```

```
Private Sub Slh_Click()
Text1.Text = ""
Text1.SetFocus
End Sub
```

Form SilahkanTunggu

```
Private Sub Timer1_Timer()
Load MauTransaksiLain: MauTransaksiLain.Show: Unload Me
End Sub
```

Form TerimaKasih

```
Option Explicit
Dim scInit, rdrLocked As Boolean
```

```
Dim retCode, numDev, numRdr, hContext, hCard, Protocol, RetLength As Long
Dim dwActProtocol, RecvLen, SendLen As Long
Dim ioRequest As SCARD_IO_REQUEST
Dim FpTemplate(0 To 767) As Byte
Dim SendBuff(0 To 262) As Byte
Dim RecvBuff(0 To 262) As Byte
Dim MatchResult As Integer
Dim apdu As APDURec
Dim Transaction As SCARD_IO_REQUEST
Dim rName, scannerName As String
Dim fpTemplateLength, procTimeOut As Long
Dim rList As String * 256
Dim desFar, desFrr, achFar, achFrr As Long
Const MAX_BUFFER_LEN = 256
Const INVALID_SW1SW2 = -450
```

```
Private Sub Form_Activate()
    Call ResetInitConnect
```

```
End Sub
```

```
Private Sub ResetInitConnect()
'Menon-aktifkan FP scanner
    If rdrLocked Then
        retCode = AET60_UnlockReader()
    End If
    retCode = AET60_Close()

'Menon-aktifkan SC reader
    If scInit Then
        retCode = SCardDisconnect(hCard, SCARD_UNPOWER_CARD)
    End If
    retCode = SCardReleaseContext(hContext)
    scInit = False
    rdrLocked = False

Dim indx As Long
    rList = String(255, vbNullChar)
    numRdr = 255

'Inisialisasi SC Reader
    retCode = SCardEstablishContext(SCARD_SCOPE_USER, 0, 0, hContext)
    If retCode <> SCARD_S_SUCCESS Then
        Call displayOut(0, retCode, "")
        Exit Sub
    End If

    retCode = SCardListReaders(hContext, rName, rList, numRdr)
    If retCode <> SCARD_S_SUCCESS Then
        Call displayOut(0, retCode, "")
        Exit Sub
    End If
    Call LoadListToControl(cbReader, rList)
    cbReader.ListIndex = 0

'Inisialisasi FP Scanner
    retCode = AET60_Open()
    If retCode <> 0 Then
        Call displayOut(1, retCode, "")
        Exit Sub
    End If
    numDev = AET60_GetNumDevices()
    If numDev > 0 Then
        cbScanner.Clear
        scannerName = String(255, vbNullChar)
        RetLength = Len(scannerName)
        For indx = 0 To (numDev - 1)
```

```

        retCode = AET60_GetDeviceName(indx, _
            scannerName, _
            RetLength)
        cbScanner.AddItem scannerName
        cbScanner.ListIndex = 0
    Next indx
Else
    Call displayOut(1, retCode, "")
    Exit Sub
End If

    Call displayOut(2, 0, "")
    Call displayOut(3, 0, "")

' 1. Connect to smartcard
retCode = SCardConnect(hContext, _
    cbReader.Text, _
    SCARD_SHARE_SHARED, _
    SCARD_PROTOCOL_T0 Or SCARD_PROTOCOL_T1, _
    hCard, _
    Protocol)
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    Exit Sub
End If
scInit = True

' 2. Lock scanner and set callback
retCode = AET60_LockReader(cbScanner.ListIndex)
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If
rdrLocked = True

retCode = AET60_SetCallback(AddressOf CALLBACK_FUNCTION)
If retCode <> 0 Then
    Call displayOut(1, retCode, "")
    Exit Sub
End If

    Call displayOut(3, 0, "Silahkan simpan Sidik Jari Nasabah")
    Call displayOut(2, 0, "Smart card telah terhubung.")
    cbReader.Enabled = False
    cbScanner.Enabled = False
End Sub

```

```

Private Sub clearMessages()
    Call displayOut(2, 0, "")
    Call displayOut(3, 0, "")
    userAction.Refresh
    MsgBox.Refresh
End Sub

```

```

Private Sub ClearBuffers()
    Dim indx As Long
    For indx = 0 To 262
        SendBuff(indx) = &H0
        RecvBuff(indx) = &H0
    Next indx
End Sub

```

```

Private Sub displayOut(ByVal errType As Integer, ByVal retVal As Long, ByVal msgText As String)
    Select Case (errType)
        Case 0 ' Smartcard error
            userAction.Text = "Error encountered. Please click on Reset button."
    
```

```

    MsgBox.Text = GetScardErrMsg(retVal)
Case 1 ' BioAPI error
    Dim StatusMsg As String
    StatusMsg = String(255, vbNullChar)
    userAction.Text = "Error encountered. Please click on Reset button."
    Call AET60_GetStatus(retVal, StatusMsg)
    MsgBox.Text = StatusMsg
Case 2 ' Message only
    MsgBox.Text = msgText
Case 3 ' Required user action
    userAction.Text = msgText
End Select
End Sub

```

Private Function validIDCard() As Boolean

```

Dim tmpStr As String
Dim indx As Integer

```

```

validIDCard = False

```

```

' 1. Select CC CC

```

```

retCode = selectFile(&HCC, &HCC)

```

```

If retCode <> SCARD_S_SUCCESS Then

```

```

    Exit Function

```

```

End If

```

```

tmpStr = ""

```

```

For indx = 0 To 1

```

```

    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "

```

```

Next indx

```

```

If tmpStr <> "91 02 " Then

```

```

    Call displayOut(2, 0, "Inserted ID card may be invalid. Return value: " + tmpStr)

```

```

    Exit Function

```

```

End If

```

```

' 2. Read ID Flag

```

```

retCode = readRecord(&H0, &H6)

```

```

If retCode <> SCARD_S_SUCCESS Then

```

```

    Exit Function

```

```

End If

```

```

tmpStr = ""

```

```

For indx = 0 To 3

```

```

    tmpStr = tmpStr + Chr(RecvBuff(indx + 2))

```

```

Next indx

```

```

If tmpStr <> "KNEF" Then

```

```

    Call displayOut(2, 0, "The user is not using a valid ID card.")

```

```

    Exit Function

```

```

End If

```

```

validIDCard = True

```

```

End Function

```

Private Function SubmitIC() As Long

```

Dim indx As Integer

```

```

Dim tmpStr As String

```

```

Call ClearBuffers

```

```

SendBuff(0) = &H80 ' CLA

```

```

SendBuff(1) = &H20 ' INS

```

```

SendBuff(2) = &H7 ' P1

```

```

SendBuff(3) = &H0 ' P2

```

```

SendBuff(4) = &H8 ' P3

```

```

SendBuff(5) = &H41 ' A

```

```

SendBuff(6) = &H43 ' C

```

```

SendBuff(7) = &H4F ' O

```

```

SendBuff(8) = &H53 ' S

```

```

SendBuff(9) = &H54 ' T

```

```

SendBuff(10) = &H45 ' E

```

```

SendBuff(11) = &H53 ' S

```

```

SendBuff(12) = &H54 ' T

```

```

SendLen = &HD
RecvLen = &H2
retCode = SendAPDU()
If retCode <> SCARD_S_SUCCESS Then
    Call displayOut(0, retCode, "")
    SubmitIC = retCode
    Exit Function
End If
tmpStr = ""
For indx = 0 To 1
    tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
Next indx
If tmpStr <> "90 00 " Then
    Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
    retCode = INVALID_SW1SW2
End If
SubmitIC = retCode
End Function

```

```

Private Function selectFile(ByVal HiAddr As Byte, ByVal LoAddr As Byte) As Long
    Call ClearBuffers
    SendBuff(0) = &H80 ' CLA
    SendBuff(1) = &HA4 ' INS
    SendBuff(2) = &H0 ' P1
    SendBuff(3) = &H0 ' P2
    SendBuff(4) = &H2 ' P3
    SendBuff(5) = HiAddr ' Value of High Byte
    SendBuff(6) = LoAddr ' Value of Low Byte
    SendLen = &H7
    RecvLen = &H2
    retCode = SendAPDU()
    If retCode <> SCARD_S_SUCCESS Then
        Call displayOut(0, retCode, "")
    End If
    selectFile = retCode
End Function

```

```

Private Function readRecord(ByVal RecNo As Byte, ByVal DataLen As Byte) As Long
    Dim indx As Integer
    Dim tmpStr As String

    Call ClearBuffers
    SendBuff(0) = &H80 ' CLA
    SendBuff(1) = &HB2 ' INS
    SendBuff(2) = RecNo ' P1 Record No
    SendBuff(3) = &H0 ' P2
    SendBuff(4) = DataLen ' P3 Length of data to be read
    SendLen = &H5
    RecvLen = SendBuff(4) + 2
    retCode = SendAPDU()
    If retCode <> SCARD_S_SUCCESS Then
        Call displayOut(0, retCode, "")
        readRecord = retCode
        Exit Function
    End If
    tmpStr = ""
    For indx = 0 To 1
        tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx + SendBuff(4))), 2) + " "
    Next indx
    If tmpStr <> "90 00 " Then
        Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
        retCode = INVALID_SW1SW2
    End If
    readRecord = retCode
End Function

```

```

Private Function writeRecord(ByVal caseType As Integer, ByVal RecNo As Byte, _
    ByVal maxDataLen As Byte, ByVal DataLen As Byte, _
    ByVal DataIn() As Byte) As Long
    Dim tmpStr As String
    Dim indx As Integer

    If caseType = 1 Then
        ' 1. Re-initialize card values to $00
        Call ClearBuffers
        SendBuff(0) = &H80      ' CLA
        SendBuff(1) = &HD2      ' INS
        SendBuff(2) = RecNo     ' P1  Record to be written
        SendBuff(3) = &H0       ' P2
        SendBuff(4) = maxDataLen ' P3  Length
        For indx = 0 To maxDataLen - 1
            SendBuff(indx + 5) = &H0
        Next indx
        SendLen = maxDataLen + 5
        RecvLen = &H2
        retCode = SendAPDU()
        If retCode <> SCARD_S_SUCCESS Then
            Call displayOut(0, retCode, "")
            writeRecord = retCode
            Exit Function
        End If
        tmpStr = ""
        For indx = 0 To 1
            tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
        Next indx
        If tmpStr <> "90 00 " Then
            Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
            retCode = INVALID_SW1SW2
            writeRecord = retCode
            Exit Function
        End If
    End If

    ' 2. Write data to card
    Call ClearBuffers
    SendBuff(0) = &H80      ' CLA
    SendBuff(1) = &HD2      ' INS
    SendBuff(2) = RecNo     ' P1  Record to be written
    SendBuff(3) = &H0       ' P2
    SendBuff(4) = DataLen   ' P3  Length
    For indx = 0 To DataLen - 1
        SendBuff(indx + 5) = DataIn(indx)
    Next indx
    SendLen = DataLen + 5
    RecvLen = &H2
    retCode = SendAPDU()
    If retCode <> SCARD_S_SUCCESS Then
        Call displayOut(0, retCode, "")
        writeRecord = retCode
        Exit Function
    End If
    tmpStr = ""
    For indx = 0 To 1
        tmpStr = tmpStr + Right("00" & Hex(RecvBuff(indx)), 2) + " "
    Next indx
    If tmpStr <> "90 00 " Then
        Call displayOut(2, 0, "The return string is invalid. Value: " + tmpStr)
        retCode = INVALID_SW1SW2
    End If
    writeRecord = retCode
End Function

```

```

Private Function SendAPDU() As Long
    ioRequest.dwProtocol = dwActProtocol

```

```
ioRequest.cbPciLength = Len(ioRequest)
```

```
retCode = SCardTransmit(hCard, _  
    ioRequest, _  
    SendBuff(0), _  
    SendLen, _  
    ioRequest, _  
    RecvBuff(0), _  
    RecvLen)
```

```
SendAPDU = retCode
```

```
End Function
```

```
Public Function GetScardErrMsg(ByVal ReturnCode As Long) As String
```

```
    Select Case ReturnCode
```

```
        'Smartcard Reader Errors
```

```
        Case SCARD_E_CANCELLED
```

```
            GetScardErrMsg = "The action was canceled by an SCardCancel request."
```

```
        Case SCARD_E_CANT_DISPOSE
```

```
            GetScardErrMsg = "The system could not dispose of the media in the requested manner."
```

```
        Case SCARD_E_CARD_UNSUPPORTED
```

```
            GetScardErrMsg = "The smart card does not meet minimal requirements for support."
```

```
        Case SCARD_E_DUPLICATE_READER
```

```
            GetScardErrMsg = "The reader driver didn't produce a unique reader name."
```

```
        Case SCARD_E_INSUFFICIENT_BUFFER
```

```
            GetScardErrMsg = "The data buffer for returned data is too small for the returned data."
```

```
        Case SCARD_E_INVALID_ATR
```

```
            GetScardErrMsg = "An ATR string obtained from the registry is not a valid ATR string."
```

```
        Case SCARD_E_INVALID_HANDLE
```

```
            GetScardErrMsg = "The supplied handle was invalid."
```

```
        Case SCARD_E_INVALID_PARAMETER
```

```
            GetScardErrMsg = "One or more of the supplied parameters could not be properly interpreted."
```

```
        Case SCARD_E_INVALID_TARGET
```

```
            GetScardErrMsg = "Registry startup information is missing or invalid."
```

```
        Case SCARD_E_INVALID_VALUE
```

```
            GetScardErrMsg = "One or more of the supplied parameter values could not be properly interpreted."
```

```
        Case SCARD_E_NOT_READY
```

```
            GetScardErrMsg = "The reader or card is not ready to accept commands."
```

```
        Case SCARD_E_NOT_TRANSACTED
```

```
            GetScardErrMsg = "An attempt was made to end a non-existent transaction."
```

```
        Case SCARD_E_NO_MEMORY
```

```
            GetScardErrMsg = "Not enough memory available to complete this command."
```

```
        Case SCARD_E_NO_SERVICE
```

```
            GetScardErrMsg = "The smart card resource manager is not running."
```

```
        Case SCARD_E_NO_SMARTCARD
```

```
            GetScardErrMsg = "The operation requires a smart card, but no smart card is currently in the device."
```

```
        Case SCARD_E_PCI_TOO_SMALL
```

```
            GetScardErrMsg = "The PCI receive buffer was too small."
```

```
        Case SCARD_E_PROTO_MISMATCH
```

```
            GetScardErrMsg = "The requested protocols are incompatible with the protocol currently in use with the card."
```

```
        Case SCARD_E_READER_UNAVAILABLE
```

```
            GetScardErrMsg = "The specified reader is not currently available for use."
```

```
        Case SCARD_E_READER_UNSUPPORTED
```

```
            GetScardErrMsg = "The reader driver does not meet minimal requirements for support."
```

```
        Case SCARD_E_SERVICE_STOPPED
```

```
            GetScardErrMsg = "The smart card resource manager has shut down."
```

```
        Case SCARD_E_SHARING_VIOLATION
```

```
            GetScardErrMsg = "The smart card cannot be accessed because of other outstanding connections."
```

```
        Case SCARD_E_SYSTEM_CANCELLED
```

```
            GetScardErrMsg = "The action was canceled by the system, presumably to log off or shut down."
```

```
        Case SCARD_E_TIMEOUT
```

```
            GetScardErrMsg = "The user-specified timeout value has expired."
```

```
        Case SCARD_E_UNKNOWN_CARD
```

```
            GetScardErrMsg = "The specified smart card name is not recognized."
```

```
        Case SCARD_E_UNKNOWN_READER
```

```
            GetScardErrMsg = "The specified reader name is not recognized."
```

```
        Case SCARD_F_COMM_ERROR
```

```
            GetScardErrMsg = "An internal communications error has been detected."
```



```

Case SCARD_F_INTERNAL_ERROR
GetScardErrMsg = "An internal consistency check failed."
Case SCARD_F_UNKNOWN_ERROR
GetScardErrMsg = "An internal error has been detected, but the source is unknown."
Case SCARD_F_WAITED_TOO_LONG
GetScardErrMsg = "An internal consistency timer has expired."
Case SCARD_S_SUCCESS
GetScardErrMsg = "No error was encountered."
Case SCARD_W_REMOVED_CARD
GetScardErrMsg = "The smart card has been removed, so that further communication is not possible."
Case SCARD_W_RESET_CARD
GetScardErrMsg = "The smart card has been reset, so any shared state information is invalid."
Case SCARD_W_UNPOWERED_CARD
GetScardErrMsg = "Power has been removed from the smart card, so that further communication is not possible."
Case SCARD_W_UNRESPONSIVE_CARD
GetScardErrMsg = "The smart card is not responding to a reset."
Case SCARD_W_UNSUPPORTED_CARD
GetScardErrMsg = "The reader cannot communicate with the card, due to ATR string configuration conflicts."
Case Else
GetScardErrMsg = "?"
End Select
End Function

```

```

Private Sub Timer1_Timer()
'Melakukan re-connect untuk menghilangkan error pada kasus kartu belum dimasukkan
retCode = SCardDisconnect(hCard, SCARD_UNPOWER_CARD)
scInit = False
retCode = SCardConnect(hContext, _
    cbReader.Text, _
    SCARD_SHARE_SHARED, _
    SCARD_PROTOCOL_T0 Or SCARD_PROTOCOL_T1, _
    hCard, _
    Protocol)
If retCode <> SCARD_S_SUCCESS Then
    Load AET60Main: AET60Main.Show: Unload Me
Exit Sub
End If
scInit = True
End Sub

```

Form TransaksiLain

```

Private Sub InfrmsiSaldo_Click()
Load InformasiSaldo: InformasiSaldo.Show: Unload Me
End Sub

```

```

Private Sub Pembayaran_Click()
Load JenisPembayaran: JenisPembayaran.Show: Unload Me
End Sub

```

```

Private Sub PlsaIsiUlng_Click()
Load PulsaIsi: PulsaIsi.Show: Unload Me
End Sub

```

```

Private Sub PnrkanTunai_Click()
Load PenarikanTunai: PenarikanTunai.Show: Unload Me
End Sub

```

```

Private Sub Trnsfr_Click()
Load Transfer: Transfer.Show: Unload Me
End Sub

```

Form Transfer

```

Private Sub Benar_Click()

```

```
If Val(Text1.Text) <= 10000 Then
Load Salah: Salah.Show: Unload Me
Else
If Val(Text1.Text) > Val(Text5.Text) Then
Load Salah: Salah.Show: Unload Me
Else
SilahkanMasukkanNominal.Text2.Text = Text1.Text
DataTransfer.Text3.Text = Text1.Text
Load SilahkanMasukkanNominal: SilahkanMasukkanNominal.Show: Unload Me
End If
End If
```

End Sub

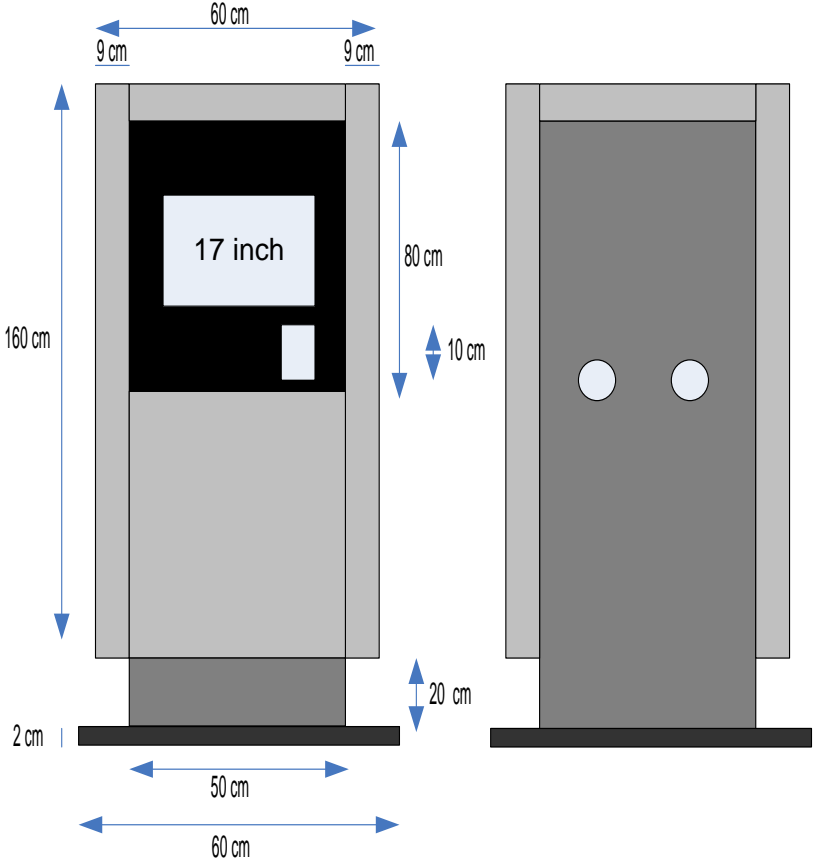
```
Private Sub Form_Activate()
Dim c As Integer
Query "SELECT Tabungan.No_Rekening FROM Tabungan"
If Not rec.EOF Then
rec.MoveLast: c = rec!No_Rekening
Text5.Text = c
End If
Text1.SetFocus
End Sub
```

```
Private Sub Slh_Click()
Text1.Text = ""
Text1.SetFocus
End Sub
```

LAMPIRAN C
GAMBAR PROTOTYPE ATM

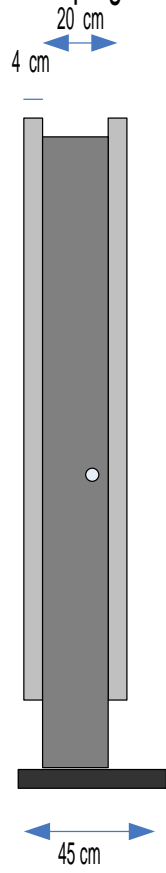
Tampak Depan

Tampak Belakang

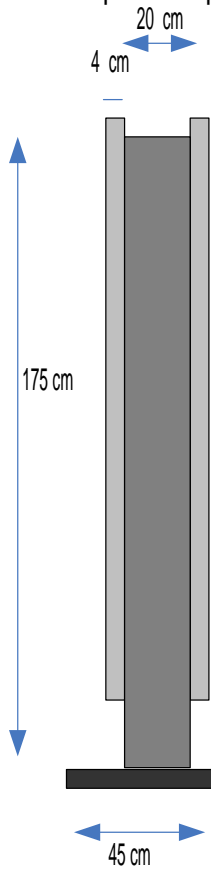


- = Kayu
- = Besi
- = Stainless Steel

Tampak Samping Kiri



Tampak Samping Kanan



■ = Kayu

■ = Besi

■ = Stainless Steel

