



LAMPIRAN 1



**CONTOH
PENCETAKKAN
KARTU**

Cindy
Jl.Salt 125
Phone : 012-1231234





12341234

Bobby
Jl.Cheese 30
Phone : 012-1231235





12341235

Alice
Jl.Sugar 45
Phone : 012-1231236





12341236

Ellie
Jl.Pepper 75
Phone : 012-1231237



12341237

Doe
Jl.Milk 89
Phone : 012-1231238



12341238

LAMPIRAN 2

LISTING PROGRAM

Form Utama

```
Private WithEvents Cat As CuteCat
```

```
Private Sub cmdAdmin_Click()  
    frmTabel.Show vbModal, Me  
End Sub
```

```
Private Sub cmdExit_Click()  
    Unload frmTabel  
    Unload frmReport  
    Unload Me  
End Sub
```

```
Private Sub cmdInput_Click()  
    If (txtInput = "") Or (Len(txtInput.Text) < 8) Then  
        'no blank input  
        txtInput.SetFocus  
    Else  
        With frmTabel.dtaFlex.Recordset  
            .MoveFirst  
            Do Until .EOF  
                If (txtInput.Text = .Fields(0)) Then  
                    Call login(txtInput.Text, Format(Time, "hh:mm:ss"))  
                    lblStatus.Caption = frmReport.dtaStatus.Recordset.RecordCount  
                    Call clearText  
                    Exit Sub  
                Else  
                    .MoveNext  
                End If  
            Loop  
            Call MsgBox("ID Denied" & vbCrLf & _  
                "Illegal ID", vbCritical, "Confirm")  
            Call clearText  
        End With  
    End If  
End Sub
```

```
Private Sub cmdInput_KeyPress(KeyAscii As Integer)  
    'set function to backspace and plus key when focus at button  
    If (KeyAscii = 27) Or (KeyAscii = 8) Or (KeyAscii = 43) _  
        Then txtInput.SetFocus  
End Sub
```

```
Private Sub Form_Activate()  
    lblStatus.Caption = frmReport.dtaStatus.Recordset.RecordCount  
End Sub
```

```
Private Sub Form_Initialize()  
    Set Cat = New CuteCat  
    Me.KeyPreview = True  
End Sub  
  
Private Sub Form_Load()  
    Load frmTabel  
    Load frmReport  
    Call DisableClose(Me, False)  
End Sub  
  
Private Sub Cat_BarcodeScanned(BCData As String, BCType As String,  
BCValid As Boolean)  
    If Not BCValid Then Exit Sub  
    txtInput.Text = BCData  
    Call cmdInput_Click  
End Sub  
  
Private Sub Cat_ScanStarted()  
    Beep  
End Sub  
  
Private Sub txtInput_Change()  
    If (Len(txtInput.Text) > 0) Then  
        lblInput.Visible = True  
        lblSign.Visible = False  
    End If  
    lblInput.Caption = "*" & txtInput.Text & "*"  
    'set focus when input 8 number id  
    If (Len(txtInput.Text) >= 8) Then cmdInput.SetFocus  
End Sub  
  
Private Sub txtInput_KeyPress(KeyAscii As Integer)  
    'input number only  
    If Not ((KeyAscii >= 48) And (KeyAscii <= 57) Or (KeyAscii = 8)) Then  
KeyAscii = 0  
End Sub  
  
Private Sub login(ByVal loginId As String, ByVal loginTime As String)  
    Dim regID As Boolean  
    regID = False  
    With frmReport.dtaStatus.Recordset  
        'check if database empty or not  
        If .RecordCount = 0 Then  
            .AddNew  
        Else  
            .MoveFirst  
            'check if ID already registered or not
```

```
Do Until .EOF
    If (.Fields(0) = loginId) Then
        regID = True
        Exit Do
    Else
        .MoveNext
    End If
Loop
'if registered
If regID Then
    lblInput.Visible = False
    lblSign.Visible = True
    lblSign.Caption = "ID      : " & .Fields(0) & vbCrLf _
        & "Name : " & .Fields(1) & vbCrLf & "Already Registered"
    Exit Sub
Else
    .AddNew
End If
End If
.Fields(0) = loginId
.Fields(1) = frmTabel.dtaFlex.Recordset.Fields(1)
.Fields(2) = loginTime
.Fields(3) = .RecordCount + 1
.Update
lblInput.Visible = False
lblSign.Visible = True
lblSign.Caption = "Registering" & vbCrLf & "ID      : " & _
    loginId & vbCrLf & "Name : " & frmTabel.dtaFlex.Recordset.Fields(1)
End With
End Sub

Private Sub clearText()
    txtInput.Text = ""
    txtInput.SetFocus
End Sub
```

Form Administrator

```
Private Sub cmdAdd_Click()
    'add new data
    frmInput.Show vbModal, Me
    Call refreshFlex
    Call hlCol
End Sub

Private Sub cmdDelete_Click()
    'delete selected data
    If flxView.Row = 0 Then Exit Sub
```

```
    Call DeletingData(flxFlexView.Text, dtaFlex)
    Call refreshFlex
End Sub

Private Sub cmdEdit_Click()
    'edit selected data
    If flxFlexView.Row = 0 Then Exit Sub
    For i = 0 To 3
        flxFlexView.Col = i + 1
        frmInput.txtInput(i).Text = flxFlexView.Text
    Next i
    Call hlCol
    frmInput.txtInput(0).Enabled = False
    'open input form
    frmInput.Show vbModal, Me
    Call refreshFlex
End Sub

Private Sub cmdExit_Click()
    'frmCreateDB.Show vbModal, Me
    Me.Hide
End Sub

Private Sub cmdPrintBarcode_Click()
    'opening report
    With envReport.rscomReport
        If optSort(0).Value Then
            .Sort = "ID"
        ElseIf optSort(1).Value Then
            .Sort = "Nama"
        End If
    End With
    Load rptPrintBarcode
    rptPrintBarcode.Show vbModal, Me
End Sub

Private Sub cmdReport_Click()
    'opening report
    With envReport.rscomReport
        If optSort(0).Value Then
            .Sort = "ID"
        ElseIf optSort(1).Value Then
            .Sort = "Nama"
        End If
    End With
    Load rptReport
    rptReport.Show vbModal, Me
End Sub
```

```
Private Sub cmdReportFlex_Click()
    frmReport.Show vbModal, Me
End Sub

Private Sub flxView_Click()
    Call hlCol
End Sub

Private Sub flxView_DblClick()
    Call cmdEdit_Click
End Sub

Private Sub Form_Load()
    Call DisableClose(Me, True)
    'Prepare database
    dtaFlex.DatabaseName = App.path & "\Account.mdb"
    dtaFlex.RecordSource = "select * from account"
    Call refreshFlex
    'open connection for report
    envReport.conReport.ConnectionString =
"Provider=Microsoft.Jet.OLEDB.3.51;Persist Security Info=False;Data Source= "
& App.path & "\Account.mdb"

    'grid size
    With flxView
        .ColWidth(0) = 100
        .ColWidth(1) = 1000
        .ColWidth(2) = 2000
        .ColWidth(3) = 1500
        .ColWidth(4) = 4700
        .ColWidth(5) = 0
    End With
End Sub

Public Function DeletingData(toDel As String, dataName As Data) As Boolean
    'set location
    Call searchId(toDel, dataName)
    With dataName.Recordset
        If MsgBox("Are you sure want to delete" & vbCrLf & _
            "ID      : " & .Fields(0) & vbCrLf & "Name : " & _
            .Fields(1), vbYesNo, "Delete Data") = vbYes Then
            .Delete
        End If
    End With
End Function

Public Function EditingData(toEdit As String, dataName As Data) As Boolean
    'set location
```



```
Call searchId(toEdit, dataName)
With dataName.Recordset
.Edit
For i = 0 To 3
.Fields(i) = frmInput.txtInput(i).Text
Next i
.Update
End With
End Function

Public Function EditingDataName(IDtoEdit As String, dataName As Data) As
Boolean
'set location
If Not (dataName.Recordset.RecordCount = 0) Then
If searchId(IDtoEdit, dataName) Then
With dataName.Recordset
.Edit
.Fields(1) = frmInput.txtInput(1).Text
.Update
End With
End If
End If
End Function

Public Function AddingData(dataName As Data) As Boolean
'set location
With dataName.Recordset
.AddNew
For i = 0 To 3
.Fields(i) = frmInput.txtInput(i).Text
Next i
.Fields(4) = "*" & .Fields(0) & "*"
.Update
End With
End Function

Private Sub hlCol()
flxView.Col = 1
flxView.ColSel = 4
lblCurrent.Caption = flxView.Row & " / "
End Sub

Private Sub refreshFlex()
i = flxView.Row
dtaFlex.Refresh
If i = flxView.Rows Then
i = i - 1
End If
flxView.Row = i
```

```
    lblTotal.Caption = flxView.Rows - 1
    Call hlCol
End Sub

Private Sub optSort_Click(Index As Integer)
    'sort by what
    Dim sortType As String
    With dtaFlex
        Select Case (Index)
            Case 0
                sortType = "ID"
            Case 1
                sortType = "Nama"
        End Select
        .RecordSource = "select * from Account ORDER BY " & sortType
        .Refresh
    End With
End Sub

Private Sub tmrTime_Timer()
    lblTime.Caption = Format(Time, "hh:mm:ss ampm")
    lblDate.Caption = WeekdayName(Weekday(Date), False, vbSunday) & ", " &
VBA.MonthName(VBA.Month(Date)) & " " & Day(Date) & ", " &
VBA.Year(Date)
    frmMain.lblTime.Caption = lblTime.Caption
    frmMain.lblDate.Caption = lblDate.Caption
    frmReport.lblTime.Caption = lblTime.Caption
    frmReport.lblDate.Caption = lblDate.Caption
End Sub

Form Laporan

Private Sub cmdAdmin_Click()
    Me.Hide
End Sub

Private Sub cmdExit_Click()
    Me.Hide
    frmTabel.Hide
End Sub

Private Sub cmdPrintPrize_Click()
'generate temporary data
    With dtaTemp.Recordset
        If Not (.RecordCount = 0) Then .MoveFirst
        If dtaStatus.Recordset.RecordCount = 0 Then
            Call MsgBox("Can't Print Anything" & vbCrLf & _
                "Database is empty")
        End If
    End With
End Sub
```

```
        Exit Sub
    End If
    Do Until .EOF
        .Delete
        .MoveNext
    Loop
    dtaPrize.Recordset.MoveFirst
    Do Until dtaPrize.Recordset.EOF
        If searchId(dtaPrize.Recordset.Fields(0), frmTabel.dtaFlex) Then
            .AddNew
            .Fields(0) = frmTabel.dtaFlex.Recordset.Fields(0)
            .Fields(1) = frmTabel.dtaFlex.Recordset.Fields(1)
            .Fields(2) = frmTabel.dtaFlex.Recordset.Fields(2)
            .Fields(3) = "Empty"
            .Fields(4) = frmTabel.dtaFlex.Recordset.Fields(3)
            .Fields(5) = dtaPrize.Recordset.Fields(2)
            .Update
        End If
        dtaPrize.Recordset.MoveNext
    Loop
    End With
    dtaTemp.Refresh
    'opening report
    With envReport.rscomStatus
        If optSortPrize(0).Value Then
            .Sort = "ID"
        ElseIf optSortPrize(1).Value Then
            .Sort = "Nama"
        ElseIf optSortPrize(2).Value Then
            .Sort = "Time"
        End If
    End With
    Load rptPrize
    rptPrize.Show vbModal, Me
End Sub

Private Sub cmdPrintStatus_Click()
    'generate temporary data
    With dtaTemp.Recordset
        If Not (.RecordCount = 0) Then .MoveFirst
        If dtaStatus.Recordset.RecordCount = 0 Then
            Call MsgBox("Can't Print Anything" & vbCrLf & _
                "Database is empty")
        End If
    End With
    Exit Sub
End If
Do Until .EOF
    .Delete
    .MoveNext
```

```
Loop
dtaStatus.Recordset.MoveFirst
Do Until dtaStatus.Recordset.EOF
    If searchId(dtaStatus.Recordset.Fields(0), frmTabel.dtaFlex) Then
        .AddNew
        .Fields(0) = frmTabel.dtaFlex.Recordset.Fields(0)
        .Fields(1) = frmTabel.dtaFlex.Recordset.Fields(1)
        .Fields(2) = frmTabel.dtaFlex.Recordset.Fields(2)
        .Fields(4) = frmTabel.dtaFlex.Recordset.Fields(3)
        .Fields(3) = dtaStatus.Recordset.Fields(2)
        .Fields(5) = dtaStatus.Recordset.Fields(3)
        .Update
    End If
    dtaStatus.Recordset.MoveNext
Loop
End With
dtaTemp.Refresh
'opening report
With envReport.rscomStatus
    If optSortStatus(0).Value Then
        .Sort = "ID"
    ElseIf optSortStatus(1).Value Then
        .Sort = "Nama"
    ElseIf optSortStatus(2).Value Then
        .Sort = "Time"
    End If
End With
Load rptStatus
rptStatus.Show vbModal, Me
End Sub

Private Sub cmdShufflePrize_Click()
    Dim accIn As Integer
    Dim i As Integer
    Dim pass As Boolean
    Dim max As Integer

    If txtMax.Text = "" Then
        max = 1
    Else
        max = txtMax.Text
    End If
    If MsgBox("Are you sure to clear current winner and shuffle new one?", _
vbYesNo, "Warning") = vbYes Then
        'clear database for prize
        With frmReport.dtaPrize.Recordset
            If Not .RecordCount = 0 Then
                'Clear Database
```

```
.MoveFirst
Do Until .EOF
    .Delete
    .MoveNext
Loop
End If
'shuffle prize
Call Randomize
accIn = dtaStatus.Recordset.RecordCount
If max > accIn Then max = accIn
txtMax.Text = max
i = 1
Do Until (i > max)
    dtaStatus.Recordset.AbsolutePosition = Rnd() * accIn
    pass = False
    If .RecordCount = 0 Then
        pass = True
    ElseIf Not searchId(dtaStatus.Recordset.Fields(0), frmReport.dtaPrize)
Then
        pass = True
    End If
    If pass Then
        .AddNew
        .Fields(0) = dtaStatus.Recordset.Fields(0)
        .Fields(1) = dtaStatus.Recordset.Fields(1)
        .Fields(2) = i
        .Update
        i = i + 1
    End If
Loop
End With

    Call refreshFlexPrize
End If
End Sub

Private Sub flxViewPrize_Click()
    Call hlColPrize
End Sub

Private Sub flxViewStatus_Click()
    Call hlColStatus
End Sub

Private Sub Form_Activate()
    Call refreshFlexStatus
    Call refreshFlexPrize
End Sub
```

```
Private Sub Form_Load()
    Call DisableClose(Me, True)
    'prepare database
    dtaStatus.DatabaseName = App.path & "\\Account.mdb"
    dtaStatus.RecordSource = "select * from status ORDER BY Time"
    dtaPrize.DatabaseName = App.path & "\\Account.mdb"
    dtaPrize.RecordSource = "select * from prize ORDER BY NoList"
    dtaTemp.DatabaseName = App.path & "\\Account.mdb"
    dtaTemp.RecordSource = "select * from temp"

    'grid size
    With flxViewStatus
        .ColWidth(0) = 100
        .ColWidth(1) = 1000
        .ColWidth(2) = 2000
        .ColWidth(3) = 1350
        .ColWidth(4) = 0
    End With
    With flxViewPrize
        .ColWidth(0) = 100
        .ColWidth(1) = 1000
        .ColWidth(2) = 2000
        .ColWidth(3) = 1000
    End With
End Sub

Private Sub cmdCleardtaStatus_Click()
    With frmReport.dtaStatus.Recordset
        If .RecordCount = 0 Then
            'if database already empty
            Call MsgBox("Empty")
        Else
            'clear database
            .MoveFirst
            Do Until .EOF
                .Delete
                .MoveNext
            Loop
        End If
    End With
    Call refreshFlexStatus
End Sub

Private Sub hlColStatus()
    flxViewStatus.Col = 1
    flxViewStatus.ColSel = 4
    lblCurrentStatus.Caption = flxViewStatus.Row & " / "
```

End Sub

```
Private Sub refreshFlexStatus()
```

```
    i = flxViewStatus.Row
```

```
    dtaStatus.Refresh
```

```
    If i = flxViewStatus.Rows Then
```

```
        i = i - 1
```

```
    End If
```

```
    flxViewStatus.Row = i
```

```
    lblTotalStatus.Caption = flxViewStatus.Rows - 1
```

```
    Call hlColStatus
```

```
End Sub
```

```
Private Sub hlColPrize()
```

```
    flxViewPrize.Col = 1
```

```
    flxViewPrize.ColSel = 3
```

```
    lblCurrentPrize.Caption = flxViewPrize.Row & " / "
```

```
End Sub
```

```
Private Sub refreshFlexPrize()
```

```
    i = flxViewPrize.Row
```

```
    dtaPrize.Refresh
```

```
    If i = flxViewPrize.Rows Then
```

```
        i = i - 1
```

```
    End If
```

```
    flxViewPrize.Row = i
```

```
    lblTotalPrize.Caption = flxViewPrize.Rows - 1
```

```
    Call hlColPrize
```

```
End Sub
```

```
Private Sub optSortPrize_Click(Index As Integer)
```

```
    'sort by what
```

```
    Dim sortType As String
```

```
    With dtaPrize
```

```
        Select Case (Index)
```

```
            Case 0
```

```
                sortType = "ID"
```

```
            Case 1
```

```
                sortType = "Nama"
```

```
            Case 2
```

```
                sortType = "NoList"
```

```
        End Select
```

```
        .RecordSource = "select * from Prize ORDER BY " & sortType
```

```
        .Refresh
```

```
    End With
```

```
End Sub
```

```
Private Sub optSortStatus_Click(Index As Integer)
```

```
'sort by what
Dim sortType As String
With dtaStatus
    Select Case (Index)
        Case 0
            sortType = "ID"
        Case 1
            sortType = "Nama"
        Case 2
            sortType = "Time"
    End Select
    .RecordSource = "select * from Status ORDER BY " & sortType
    .Refresh
End With
End Sub

Private Sub txtShuffle_KeyPress(KeyAscii As Integer)
    'input number only
    If Not ((KeyAscii >= 48) And (KeyAscii <= 57) Or (KeyAscii = 8)) Then
        KeyAscii = 0
    End Sub
End Sub
```

Form Input

```
Private Sub cmdCancel_Click()
    txtInput(0).Enabled = True
    Unload Me
End Sub

Private Sub cmdOk_Click()
    'check edit mode or add mode
    If txtInput(0).Enabled Then
        'add mode
        If frmTabel.flxView.Row = 0 Then
            If txtInput(1).Text = "" Then
                MsgBox "Fill your Name"
                Exit Sub
            ElseIf Len(txtInput(0).Text) = 8 Then
                Call frmTabel.AddingData(frmTabel.dtaFlex)
            Else
                MsgBox "Fill your ID with 8 number"
                Exit Sub
            End If
        ElseIf searchId(txtInput(0).Text, frmTabel.dtaFlex) Then
            MsgBox "Id Used"
            Exit Sub
        ElseIf txtInput(1).Text = "" Then
            MsgBox "Fill your Name"
        End If
    End Sub
```



```
Exit Sub
ElseIf Len(txtInput(0).Text) = 8 Then
    Call frmTabel.AddingData(frmTabel.dtaFlex)
Else
    MsgBox "Fill your ID with 8 number"
    Exit Sub
End If
Else
'edit mode
Call frmTabel.EditingData(frmTabel.flxView.Text, frmTabel.dtaFlex)
Call frmTabel.EditingDataName(txtInput(0).Text, frmReport.dtaStatus)
Call frmTabel.EditingDataName(txtInput(0).Text, frmReport.dtaPrize)
End If
'exit input form
Call cmdCancel_Click
End Sub

Private Sub Form_Activate()
    If txtInput(0).Enabled Then
        lblTitle.Caption = "Add New Data"
        frmInput.Caption = "Administrator - Add New Data"
    Else
        lblTitle.Caption = "Edit Data"
        frmInput.Caption = "Administrator - Edit Data"
    End If
End Sub

Private Sub txtInput_Change(Index As Integer)
    'set focus when input 8 number id
    If Index = 0 And lblTitle.Caption = "Add New Data" Then
        If (Len(txtInput(0).Text) >= 8) Then txtInput(1).SetFocus
    End If
End Sub

Private Sub txtInput_KeyPress(Index As Integer, KeyAscii As Integer)
    'input number only
    If Index = 0 Then
        If Not ((KeyAscii >= 48) And (KeyAscii <= 57) Or (KeyAscii = 8)) Then
            KeyAscii = 0
        End If
    End Sub
```

Modul Database

```
Public Sub MoveData(ByVal Index As Integer, dataName As Data)
    'Database navigator
    With dataName.Recordset
        Select Case Index
```

```
Case 0:
    .MoveFirst
Case 1:
    .MovePrevious
    If .BOF Then .MoveFirst
Case 2:
    .MoveNext
    If .EOF Then .MoveLast
Case 3:
    .MoveLast
End Select
End With
End Sub

Public Sub ViewData(controlsName As Variant, dataName As Data)
    'View data
    Dim i As Integer
    For i = controlsName.LBound To controlsName.UBound
        controlsName(i).Text = dataName.Recordset.Fields(i)
    Next i
End Sub

Public Sub ModeView(modeData As Boolean, controlsName As Variant,
Optional fromDate As Integer = 0, Optional todata As Integer = 0)
    'View data
    Dim i As Integer
    For i = controlsName.LBound + fromDate To controlsName.UBound - todata
        controlsName(i).Enabled = modeData
    Next i
End Sub

Public Function searchId(toFind As String, dataName As Data) As Boolean
    'Searching ID
    searchId = False
    With dataName.Recordset
        .MoveFirst
        Do Until .EOF
            If (.Fields(0) = toFind) Then
                searchId = True
                Exit Function
            Else
                .MoveNext
            End If
        Loop
    End With
End Function
```

Class CuteCat

Option Explicit

Private CatScratching As Boolean, CatData As String

Private BCData As String, BCType As String, BCValid As Boolean, PrevPtr As

Integer

Public Event ScanStarted()

Public Event BarcodeScanned(BCData As String, BCType As String, BCValid As Boolean)

'Base64 decode constants

Private m_bytReverseIndex(0 To 255) As Byte

Private Const k_bytEqualSign As Byte = 61

Private Const k_bytMask1 As Byte = 3 '00000011

Private Const k_bytMask2 As Byte = 15 '00001111

Private Const k_bytMask3 As Byte = 63 '00111111

Private Const k_bytMask4 As Byte = 192 '11000000

Private Const k_bytMask5 As Byte = 240 '11110000

Private Const k_bytMask6 As Byte = 252 '11111100

Private Const k_bytShift2 As Byte = 4

Private Const k_bytShift4 As Byte = 16

Private Const k_bytShift6 As Byte = 64

Private Const k_lMaxBytesPerLine As Long = 152

Private Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory"

(ByVal Destination As Long, ByVal Source As Long, ByVal Length As Long)

Public Property Get BarcodeData() As String

BarcodeData = BCData

End Property

Public Property Get BarcodeType() As String

BarcodeType = BCType

End Property

Public Property Get BarcodeValid() As String

BarcodeValid = BCValid

End Property

Private Sub Class_Initialize()

Dim i As Integer

For i = Asc("a") To Asc("z") 'lowercase letters

m_bytReverseIndex(i) = (i - Asc("a"))

Next i

For i = Asc("A") To Asc("Z") 'uppercase letters

m_bytReverseIndex(i) = 26 + (i - Asc("A"))

Next i

For i = Asc("0") To Asc("9") '0-9

m_bytReverseIndex(i) = 52 + (i - Asc("0"))

```

Next i
m_bytReverseIndex(43) = 62 'Asc("+")
m_bytReverseIndex(47) = 63 'Asc("/")
End Sub

Private Function Decode(sInput As String) As String
    If sInput = "" Then Exit Function
    Decode = StrConv(DecodeArr(sInput), vbUnicode)
End Function

Private Function DecodeArr(sInput As String) As Byte()
    'returns a SBCS byte array
    Dim bytInput() As Byte      'base64 encoded string to work with
    Dim bytWorkspace() As Byte  'byte array to use as workspace
    Dim bytResult() As Byte    'array that result will be copied to
    Dim lInputCounter As Long   'iteration counter for input array
    Dim lWorkspaceCounter As Long 'iteration counter for workspace array
    'get rid of CrLfs, and "="s since they're not required for decoding,
    'and place the input in the byte array
    If sInput = "" Then Exit Function
    bytInput = Replace(Replace(sInput, vbCrLf, ""), "=", "")
    'size the workspace
    ReDim bytWorkspace(LBound(bytInput) To (UBound(bytInput) * 2)) As Byte
    lWorkspaceCounter = LBound(bytWorkspace)
    'pass bytes back through index to get original values
    For lInputCounter = LBound(bytInput) To UBound(bytInput)
        bytInput(lInputCounter) = m_bytReverseIndex(bytInput(lInputCounter))
    Next lInputCounter
    For lInputCounter = LBound(bytInput) To (UBound(bytInput) -
((UBound(bytInput) Mod 8) + 8)) Step 8
        'left shift first input byte by 2 and right shift second input byte by 4
        bytWorkspace(lWorkspaceCounter) = (bytInput(lInputCounter) *
k_bytShift2) + (bytInput(lInputCounter + 2) \ k_bytShift4)

        'mask bits 5-8 of second byte, left shift it by 4
        'right shift third byte by 2, add it to result of second byte
        bytWorkspace(lWorkspaceCounter + 1) = ((bytInput(lInputCounter + 2) And
k_bytMask2) * k_bytShift4) + _
            (bytInput(lInputCounter + 4) \ k_bytShift2)
        'mask bits 3-8 of third byte, left shift it by 6, add it to fourth byte
        bytWorkspace(lWorkspaceCounter + 2) = ((bytInput(lInputCounter + 4) And
k_bytMask1) * k_bytShift6) + _
            bytInput(lInputCounter + 6)
        lWorkspaceCounter = lWorkspaceCounter + 3
    Next lInputCounter
    'decode any remaining bytes that are not part of a full 4 byte block
    Select Case (UBound(bytInput) Mod 8):
        Case 3:

```

```

        'left shift first input byte by 2 and right shift second input byte by 4
        bytWorkspace(IWorkspaceCounter) = (bytInput(IInputCounter) *
k_bytShift2) + (bytInput(IInputCounter + 2) \ k_bytShift4)
    Case 5:
        'left shift first input byte by 2 and right shift second input byte by 4
        bytWorkspace(IWorkspaceCounter) = (bytInput(IInputCounter) *
k_bytShift2) + (bytInput(IInputCounter + 2) \ k_bytShift4)
        'mask bits 5-8 of second byte, left shift it by 4
        'right shift third byte by 2, add it to result of second byte
        bytWorkspace(IWorkspaceCounter + 1) = ((bytInput(IInputCounter + 2)
And k_bytMask2) * k_bytShift4) + _
            (bytInput(IInputCounter + 4) \ k_bytShift2)
        IWorkspaceCounter = IWorkspaceCounter + 1
    Case 7:
        'left shift first input byte by 2 and right shift second input byte by 4
        bytWorkspace(IWorkspaceCounter) = (bytInput(IInputCounter) *
k_bytShift2) + (bytInput(IInputCounter + 2) \ k_bytShift4)
        'mask bits 5-8 of second byte, left shift it by 4
        'right shift third byte by 2, add it to result of second byte
        bytWorkspace(IWorkspaceCounter + 1) = ((bytInput(IInputCounter + 2)
And k_bytMask2) * k_bytShift4) + _
            (bytInput(IInputCounter + 4) \ k_bytShift2)
        'mask bits 3-8 of third byte, left shift it by 6, add it to fourth byte
        bytWorkspace(IWorkspaceCounter + 2) = ((bytInput(IInputCounter + 4)
And k_bytMask1) * k_bytShift6) + _
            bytInput(IInputCounter + 6)
        IWorkspaceCounter = IWorkspaceCounter + 2
    End Select

    'size the result array
    ReDim bytResult(LBound(bytWorkspace) To IWorkspaceCounter) As Byte
    'if option base is set to 1 then don't increment this value
    If LBound(bytWorkspace) = 0 Then
        IWorkspaceCounter = IWorkspaceCounter + 1
    End If
    'move decoded data to a properly sized array
    CopyMemory VarPtr(bytResult(LBound(bytResult))),
VarPtr(bytWorkspace(LBound(bytWorkspace))), IWorkspaceCounter
    'return
    'CuteCat encryption hack
    For IWorkspaceCounter = LBound(bytResult) To UBound(bytResult)
        bytResult(IWorkspaceCounter) = bytResult(IWorkspaceCounter) Xor 67
    Next
    DecodeArr = bytResult
End Function

Public Sub KeyDown(KeyCode As Integer, Shift As Integer)
Static PrevKey As Integer, PrevShift As Integer

```

```
'start processing barcode data from scanner if Alt-F10 is detected
If PrevKey = 18 And PrevShift = 4 And KeyCode = 121 And Shift = 4 Then
    CatData = ""
    CatScratching = True
    PrevPtr = Screen.MousePointer
    RaiseEvent ScanStarted
    Screen.MousePointer = vbHourglass
End If
PrevKey = KeyCode
PrevShift = Shift
End Sub

Public Sub KeyPress(KeyAscii As Integer)
Dim X As Long, Y As Long, Z As Long
If CatScratching Then
    If KeyAscii = 13 Then
        CatScratching = False
        If CatData = "" Then GoTo quit
        'Parse out barcode data from scanner input
        X = InStr(2, CatData, ".") + 1
        If X = 1 Then GoTo quit
        Y = InStr(X, CatData, ".") + 1
        If Y = 1 Then GoTo quit
        Z = InStr(Y, CatData, ".")
        If Z = 0 Then GoTo quit
        BCType = Decode(Mid$(CatData, X, (Y - X)))
        BCData = Decode(Mid$(CatData, Y, (Z - Y)))
        BCValid = True
        RaiseEvent BarcodeScanned(BCData, BCType, BCValid)
quit:
        Screen.MousePointer = PrevPtr
    Else
        If (Len(CatData) = 0) And (KeyAscii <> 46) Then
            'Invalid barcode data; should always start with a period
            CatScratching = False
            BCValid = False
            RaiseEvent BarcodeScanned("", "", BCValid)
            Screen.MousePointer = PrevPtr
        Else
            CatData = CatData & Chr$(KeyAscii)
        End If
    End If
End If
KeyAscii = 0
End If
End Sub
```