


Lampiran A

LISTING PROGRAM

1. Menu Utama

Private Sub Abt_Click()
    frmAbout.Show
End Sub

Private Sub AnalDisp_Click()
    frmDisparity1.Show
End Sub

Private Sub Hasil_Click()
    frmHasil.Show
End Sub

Private Sub Hlp_Click()
    frmHelp.Show
End Sub

Private Sub keluar_Click()
End
End Sub

2. KoreksiDisparity

Private Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long,
    ByVal X As Long, ByVal Y As Long) As Long
Private Declare Function SetPixel Lib "gdi32" (ByVal hdc As Long,
    ByVal X As Long, ByVal Y As Long, ByVal crColor As Long) As Long
Dim R As Integer, G As Integer, B As Integer
Dim Itensity As Long, GradX As Long, GradY As Long, Grad As Long
Dim PixelValue As Long
Dim NilGray1 As Single
Dim NilGray2 As Single
Dim NilGray3 As Single
Dim NilGray4 As Single
Dim NilTGray3 As Single
Dim NilTGray4 As Single
Dim Disparity_0 As Single
Dim Disparity_Neg As Single
Dim Disparity_Pos As Single
Dim PerDisp_0 As Single
Dim PerDisp_Neg As Single
Dim PerDisp_Pos As Single
Dim Dimensi As Single
Dim bolCitra1 As Boolean
Dim bolCitra2 As Boolean
Dim NmFile1, NmFile2, NmFile3 As String * 50
Dim Path1, Path2, Path3 As String * 100
Dim rsNo As Recordset
Dim noPiksel As Single
Sub Bersih()
R = 0
G = 0
B = 0
Itensity = 0
GradX = 0
GradY = 0
Grad = 0
PixelValue = 0
NilGray1 = 0
NilGray2 = 0
NilGray3 = 0
NilGray4 = 0
NilTGray3 = 0
NilTGray4 = 0
Disparity_0 = 0
Disparity_Neg = 0
Disparity_Pos = 0
PerDisp_0 = 0
PerDisp_Neg = 0
PerDisp_Pos = 0
Dimensi = 0
bolCitra1 = False
bolCitra2 = False
NmFile1 = ""
NmFile2 = ""
NmFile3 = ""
Path1 = ""
Path2 = ""
Path3 = ""
noPiksel = 0
lblDispNeg = ""
lblDisp0 = ""
lblDispPos = ""
lblRotasi = ""
PicHasil.Picture = Nothing
lblDimensi = ""
End Sub

Sub DecTORGB(ByVal Col As Long, R As Integer, G As Integer, B As Integer)
R = Col Mod 256
G = ((Col - R) Mod 65536) / 256
B = (Col - R - G) / 65536
If R < 0 Then R = 0: If R >= 255 Then R = 255
If G < 0 Then G = 0: If G >= 255 Then G = 255
If B < 0 Then B = 0: If B >= 255 Then B = 255
End Sub

Private Sub cmdBatal_Click()
Call Bersih
Pic1.Picture = Nothing
Pic1a.Picture = Nothing
PicBin1.Picture = Nothing
PicBin2.Picture = Nothing
Pic2.Picture = Nothing
Pic2a.Picture = Nothing
Call Bersih
lvHorKiri.ListItems.Clear
lvHorKanan.ListItems.Clear
lblNmFile1 = ""
lblNmFile2 = ""
lblDispVer = ""
bolCitra1 = False
bolCitra2 = False
cmdProses.Enabled = False
cmdSave.Enabled = False
Set rsNo = New ADODB.Recordset
rsNo.Open "Select * from tDisparity", CN, 1, 2
If rsNo.EOF Then
    cmdReset.Enabled = False
Else
    cmdReset.Enabled = True
End If
Set rsNo = Nothing
End Sub

Private Sub cmdExit_Click()
Unload Me
End Sub

Private Sub cmdProses_Click()
    cmdProses.Enabled = False
    noPiksel = 0
    Call HapusBiner
    Call BuatBiner1
    Call BuatBiner2
    Call DeteksiTepi
    Call Disparity
    MsgBox "Proses selesai ..."
    cmdSave.Enabled = True
End Sub

Sub BuatBiner1()
    Dim Gray As Single
    Dim Gray1 As Single
    Dim Gray2 As Single
    Dim R1 As Single, G1 As Single, B1 As Single
    PicBin1.Cls
    For X = 0 To Pic1.ScaleWidth
        DoEvents
        For Y = 0 To Pic1.ScaleHeight
            GET COLORS Pic1.Point(X, Y), R1, G1, B1
            Gray1 = (R1 + G1 + B1) / 3
If Gray1 < 128 Then
    PicBin1.PSet (X, Y), RGB(0, 0, 0)
Else
    PicBin1.PSet (X, Y), RGB(255, 255, 255)
End If
Next Y
Next X
End Sub

Sub BuatBiner2()
Dim Graya As Single
Dim Gray1a As Single
Dim Gray2a As Single
Dim R1a As Single, G1a As Single, B1a As Single
PicBin2.Cls
    For X = 0 To Pic1a.ScaleWidth
        DoEvents
        For Y = 0 To Pic1a.ScaleHeight
            GET_COLORS Pic1a.Point(X, Y), R1a, G1a, B1a
            Gray1a = (R1a + G1a + B1a) / 3
            If Gray1a < 128 Then
                PicBin2.PSet (X, Y), RGB(0, 0, 0)
            Else
                PicBin2.PSet (X, Y), RGB(255, 255, 255)
            End If
        Next Y
    Next X
End Sub

Sub Disparity()
Dim s As Single
Dim X As Long
Dim X1 As Long
Dim X2 As Long
Dim Y As Long
Dim Y1 As Long
Dim Y2 As Long
Dim R1 As Single
Dim G1 As Single
Dim B1 As Single
Dim R2 As Single
Dim G2 As Single
Dim B2 As Single
Dim R3 As Single
Dim G3 As Single
Dim B3 As Single
Dim R4 As Single
Dim G4 As Single
Dim B4 As Single
Dim rsCari1 As Recordset
Dim rsCari2 As Recordset
Dim NilPixel1 As Single
Dim NilPixel2 As Single
Dim IntBenar As Single
Dim NoPixel1 As Single
Dim NoPixel2 As Single
Dim Rolling As Single
Dim No As Single
Dim Dms As Single
Dim Lebar As Single
Dim Panjang As Single
Dim Gray As Single
Dim Gray1 As Single
Dim Gray2 As Single
Dms = 0
Lebar = 0
tinggi = 0
Disparity_0 = 0
Disparity_Neg = 0
Disparity_Pos = 0
NilGray1 = 0
NilGray2 = 0
NilGray3 = 0
NilGray4 = 0
NilTGray3 = 0
NilTGray4 = 0
Dimensi = 0
PerDisp_0 = 0
PerDisp_Neg = 0
PerDisp_Pos = 0
Pic2a.Height = Pic2.Height
Pic2a.Width = Pic2.Width
Dimensi = Val(Pic2.ScaleHeight) * Val(Pic2.ScaleWidth)
lblDimensi.Caption = "Dimensi Citra: " & Dimensi & " Pixel"
PicHasil.Cls
For X = 0 To Pic2.ScaleWidth
    DoEvents
    Lebar = Lebar + 1
    For Y = 0 To Pic2.ScaleHeight
        GET_Gray Pic2.Point(X, Y), Gray1
tinggi = tinggi + 1
        Dms = Dms + 1
        GET_Gray Pic2a.Point(X, Y), Gray2
        If (Gray1 - Gray2) < 0 Then
            Disparity_Pos = Disparity_Pos + 1
        ElseIf (Gray1 - Gray2) = 0 Then
            Disparity_0 = Disparity_0 + 1
        ElseIf (Gray1 - Gray2) > 0 Then
            Disparity_Neg = Disparity_Neg + 1
        End If
        If (Gray1 - Gray2) > 0 Then
            PicHasil.PSet (X, Y), RGB(256, 0, 0)
        ElseIf (Gray1 - Gray2) = 0 Then
            PicHasil.PSet (X, Y), RGB(0, 0, 0)
        ElseIf (Gray1 - Gray2) < 0 Then
            PicHasil.PSet (X, Y), RGB(255, 255, 255)
        End If
        G3 = (G1 - G2)
        B3 = (B1 - B2)
    Next Y
    PerDisp_0 = (Disparity_0 / Dms) * 100
    PerDisp_Neg = (Disparity_Neg / Dms) * 100
    PerDisp_Pos = (Disparity_Pos / Dms) * 100
Next X
lblDisp0.Caption = PerDisp_0 & " %"
lblDispPos.Caption = PerDisp_Pos & " %"
lblDispNeg.Caption = PerDisp_Neg & " %"
X = 0
Y = 0
Dms = 0
Y1 = 1

For X1 = 1 To Pic1.ScaleWidth
    DoEvents
    GET COLORS Pic1.Point(X1, Y1), R3, G3, B3
    NilGray3 = (R3 + G3 + B3) / 3
    NilTGray3 = NilTGray3 + NilGray3
    If NilGray3 < 128 Then
        NilGray3 = 0
    Else
        NilGray3 = 1
    End If

    Set rsNo = New ADODB.Recordset
    rsNo.Open "tBinerKiri", CN, 1, 2
    rsNo.AddNew
    rsNo!NoPixel = X1
    rsNo!Nilai = NilGray3
    rsNo.Update
    Set rsNo = Nothing
Next X1

NilTGray3 = (NilTGray3 / Pic2.ScaleWidth) - 128
If NilTGray3 < 128 Then
    NilTGray3 = 0
Else
    NilTGray3 = 1
End If

Y2 = 1
For X2 = 1 To Pic1a.ScaleWidth
    DoEvents
    GET COLORS Pic1a.Point(X2, Y2), R4, G4, B4 ' , INVERTED1
    NilGray4 = (R4 + G4 + B4) / 3
    NilTGray4 = NilTGray4 + NilGray4
    'NilGray4 = NilGray4 - 128
    If NilGray4 < 128 Then
        NilGray4 = 0
    Else
        NilGray4 = 1
    End If

    Set rsNo = New ADODB.Recordset
    rsNo.Open "tBinerKanan", CN, 1, 2
    rsNo.AddNew
    rsNo!NoPixel = X2
    rsNo!Nilai = NilGray4
    rsNo.Update
    Set rsNo = Nothing
Next X2

NilTGray4 = (NilTGray4 / Pic2a.ScaleWidth) - 128
If NilTGray4 < 128 Then
    NilTGray4 = 0
Else
    NilTGray4 = 1
End If
Set rsCari1 = New ADODB.Recordset
rsCari1.Open "Select * from tBinerKiri order by noPixel asc",
CN, 1, 2
If Not rsCari1.EOF Then
    Do While Not rsCari1.EOF
        NoPixel1 = rsCari1!NoPixel
        NilPixel1 = rsCari1!Nilai
        Set J = lvHorKiri.ListItems.Add(, , NoPixel1)
        J.SubItems(1) = NilPixel1
        No = No + 1
        Set rsCari2 = New ADODB.Recordset
        rsCari2.Open "Select * from tBinerKanan where
noPixel=" & NoPixel1 & ", CN, 1, 2
        If Not rsCari2.EOF Then
            NoPixel2 = rsCari2!NoPixel
            NilPixel2 = rsCari2!Nilai
            Set J = lvHorKanan.ListItems.Add(, , NoPixel2)
            J.SubItems(1) = NilPixel2
            If NilPixel2 = NilPixel1 Then
                IntBenar = IntBenar + 1
            End If
        End If
        Set rsCari2 = Nothing
    End If
    rsCari1.MoveNext
End If
Set rsCari1 = Nothing
Rolling = (IntBenar / No) * 100
If Rolling = 100 Then
    lblRotasi.Caption = "Tidak"
Else
    lblRotasi.Caption = "Ada"
End If
Beep
Exit Sub
End Sub
Private Sub GET_COLORS(COLOR As Long, ByRef R As Single, ByRef G As Single, ByRef B As Single)
    R = COLOR Mod 255
    G = (COLOR And &HFF00) / 256
    B = (COLOR And &HFF0000) / 65536
    If R > 256 Then
        R = 256
    End If
    If G > 256 Then
        G = 256
    End If
    If B > 256 Then
        B = 256
    End If
    noPiksel = noPiksel + 1
End Sub
Private Sub GET_Gray(COLOR As Long, ByRef Gray As Single)
Gray = COLOR 'Mod 255
If Gray > 255 Then
Gray = 255
End If
End Sub

Private Sub cmdReset_Click()
CmdReset.Enabled = False
Call Bersih
Set rsNo = New ADODB.Recordset
rsNo.Open "Delete from tDisparity", CN, 1, 2
Set rsNo = Nothing
Pic1.Picture = Nothing
Pic1a.Picture = Nothing
Pic2.Picture = Nothing
Pic2a.Picture = Nothing
PicHasil.Picture = Nothing
LblDispNeg = ""
LblDisp0 = ""
LblDispPos = ""
LblRotasi = ""
LvHorKiri.ListItems.Clear
LvHorKanan.ListItems.Clear
LblNmFile1 = ""
LblNmFile2 = ""
BolCitra1 = False
BolCitra2 = False
cmdProses.Enabled = False
cmdSave.Enabled = False
MsgBox "Data sudah di Reset"
End Sub

Sub HapusBiner()
Set rsNo = New ADODB.Recordset
rsNo.Open "Delete from tBinerKiri", CN, 1, 2
Set rsNo = Nothing
Set rsNo = New ADODB.Recordset
rsNo.Open "Delete from tBinerKanan", CN, 1, 2
Set rsNo = Nothing
End Sub

Private Sub cmdSave_Click()
CmdSave.Enabled = False
Set rsNo = New ADODB.Recordset
rsNo.Open "Select * from tDisparity order by ID desc", CN, 1, 2
If Not rsNo.EOF Then
Id = rsNo!Id + 1
Else
Id = 1
End If
Set rsNo = Nothing
Set rsNo = New ADODB.Recordset
rsNo.Open "tDisparity", CN, 1, 2
rsNo.AddNew
rsNo!Id = Id
rsNo!NmFile1 = NmFile1
rsNo!NmFile2 = NmFile2
rsNo!NmFile3 = "Hasil " & Id 'NmFile3
rsNo!Path1 = Path1
rsNo!Path2 = Path2
rsNo!Path3 = App.Path
rsNo!Disp_Neg = PerDisp_Neg
rsNo!Disp_0 = PerDisp_0
rsNo!Disp_Pos = PerDisp_Pos
rsNo!Rotasi = lblRotasi.Caption
rsNo.Update
Set rsNo = Nothing
Set rsNo = New ADODB.Recordset
rsNo.Open "Select * from tDisparity", CN, 1, 2
If rsNo.EOF Then
    cmdReset.Enabled = False
Else
    cmdReset.Enabled = True
End If
Set rsNo = Nothing
NmFile1 = ""
NmFile2 = ""
NmFile3 = ""
Path1 = ""
Path2 = ""
Path3 = ""
Id = 0
PerDisp_Neg = 0
PerDisp_0 = 0
PerDisp_Pos = 0
Call Bersih
Pic1.Picture = Nothing
Pic1a.Picture = Nothing
Pic2.Picture = Nothing
Pic2a.Picture = Nothing
PicHasil.Picture = Nothing
lblDimensi = ""
Call Bersih
lvHorKiri.ListItems.Clear
lvHorKanan.ListItems.Clear
lblNmFile1 = ""
lblNmFile2 = ""
lblDispVer = ""
bolCitra1 = False
bolCitra2 = False
cmdProses.Enabled = False
cmdSave.Enabled = False
MsgBox "Citra sudah disimpan .."
End Sub

Private Sub Command1_Click()
Dim Token As Long
CM.Filter = "Image|*.bmp;*.jpg"
CM.ShowOpen
If CM.FileName <> "" Then
    Token = InitGDIPlus
End Sub
Pic1.Picture = LoadPictureGDIPlus(CM.FileName, Pic1.Width, Pic1.Height, , False)
FreeGDIPlus Token
bolCitra1 = True
NmFile1 = CM.FileTitle
lblNmFile1 = NmFile1
Path1 = CM.FileName
End If
End Sub

Private Sub Command2_Click()
Dim Token As Long
CM.Filter = "Image|*.bmp;*.jpg"
CM.ShowOpen
If CM.FileName <> "" Then
    Token = InitGDIPlus
    Pic1a.Picture = LoadPictureGDIPlus(CM.FileName, Pic1a.Width, Pic1a.Height, , False)
    FreeGDIPlus Token
    bolCitra2 = True
    NmFile2 = CM.FileTitle
    lblNmFile2 = NmFile2
    Path2 = CM.FileName
End If
If bolCitra1 = True And bolCitra2 = True Then
    cmdProses.Enabled = True
Else
    cmdProses.Enabled = False
End If
End Sub

Private Sub Form_Activate()
bolCitra1 = False
bolCitra2 = False
Call Bersih
Set rsNo = New ADODB.Recordset
rsNo.Open "Select * from tDisparity", CN, 1, 2
If rsNo.EOF Then
    cmdReset.Enabled = False
Else
    cmdReset.Enabled = True
End If
Set rsNo = Nothing
cmdProses.Enabled = False
cmdSave.Enabled = False
End Sub

Private Sub Form_Load()
cmdProses.Enabled = False
cmdSave.Enabled = False
Call Koneksi
End Sub

Sub DeteksiTepi()
Dim Op_X(-1 To 1, -1 To 1) As Integer, Op_Y(-1 To 1, -1 To 1) As Integer
Dim X As Integer, Y As Integer, i As Integer, J As Integer
Pic2.Cls
Grad = 0
Op_X(-1, -1) = -1: Op_X(0, -1) = -2: Op_X(1, -1) = -1
Op_X(-1, 0) = 0: Op_X(0, 0) = 0: Op_X(1, 0) = 0
Op_X(-1, 1) = 1: Op_X(0, 1) = 2: Op_X(1, 1) = 1

Op_Y(-1, -1) = -1: Op_Y(0, -1) = 0: Op_Y(1, -1) = 1
Op_Y(-1, 0) = -2: Op_Y(0, 0) = 0: Op_Y(1, 0) = 2
Op_Y(-1, 1) = -1: Op_Y(0, 1) = 0: Op_Y(1, 1) = 1

Dim tmps As String
tmps = "Kernel Sobel Citra Kiri" & vbCrLf & "Horisontal >>" & vbCrLf
For X = -1 To 1
For Y = -1 To 1
    tmps = tmps & Format(Op_X(X, Y), "0") & " 
Next Y
tmps = tmps & vbCrLf
Next X
tmps = tmps & "Vertikal >>" & vbCrLf
For X = -1 To 1
For Y = -1 To 1
    tmps = tmps & Format(Op_Y(X, Y), "0") & " 
Next Y
tmps = tmps & vbCrLf
Next X
DoEvents
For Y = 0 To PicBin1.Height - 1
    For X = 0 To PicBin1.Width - 1
        GradX = 0: GradY = 0: Grad = 0
        If X = 0 Or Y = 0 Or X = PicBin1.Width - 1 Or Y = _
            PicBin1.Height - 1 Then
            Grad = 0
        Else
            For i = -1 To 1
                For J = -1 To 1
                    PixelValue = GetPixel(PicBin1.hdc, X + i, Y + J)
                    DecTORGB PixelValue, R, G, B
                    Itensity = (R + G + B) / 3 'Itensitas / B & W
                    GradX = GradX + (Itensity * Op_X(i, J))
                    GradY = GradY + (Itensity * Op_Y(i, J))
                Next J
            Next i
            Grad = Round(Sqr(Abs(GradX * GradX) + Abs(GradY * 
                            GradY)))
        End If
        If Grad <= 0 Then Grad = 0: If Grad >= 255 Then Grad = 255
        SetPixel Pic2.hdc, X, Y, RGB(Grad, Grad, Grad)
    Pic2.Refresh
Next X
Pic2.Refresh
Next Y
Pic2a.Cls
Grad = 0
Op_X(-1, -1) = -1: Op_X(0, -1) = -2: Op_X(1, -1) = -1
Op_X(-1, 0) = 0: Op_X(0, 0) = 0: Op_X(1, 0) = 0
Op_X(-1, 1) = 1: Op_X(0, 1) = 2: Op_X(1, 1) = 1
Op_Y(-1, -1) = -1: Op_Y(0, -1) = 0: Op_Y(1, -1) = 1  
Op_Y(-1, 0) = -2: Op_Y(0, 0) = 0: Op_Y(1, 0) = 2  
Op_Y(-1, 1) = -1: Op_Y(0, 1) = 0: Op_Y(1, 1) = 1

tmps = ""  
tmps = "Kernel Sobel Citra Kanan" & vbCrLf & "Horisontal >>" & vbCrLf  
For X = -1 To 1  
    For Y = -1 To 1  
        tmps = tmps & Format(Op_X(X, Y), "0") & " "  
    Next Y  
    tmps = tmps & vbCrLf  
Next X  

For X = -1 To 1  
    For Y = 0 To PicBin2.Height - 1  
        For X = 0 To PicBin2.Width - 1  
            GradX = 0: GradY = 0: Grad = 0  
            If X = 0 Or Y = 0 Or X = PicBin2.Width - 1 Or Y = PicBin2.Height - 1 Then  
                Grad = 0  
            Else  
                For i = -1 To 1  
                    For J = -1 To 1  
                        PixelValue = GetPixel(PicBin2.hdc, X + i, Y + J)  
                        DecTORGB PixelValue, R, G, B  
                        Itensity = (R + G + B) / 3  
                        GradX = GradX + (Itensity * Op_X(i, J))  
                        GradY = GradY + (Itensity * Op_Y(i, J))  
                    Next J  
                Next i  
                Grad = Round(Sqr(Abs(GradX * GradX) + Abs(GradY * GradY)))  
            End If  
        Next X  
    Next Y  
Next X  
DoEvents  
End Sub

3. Modul-1

Option Explicit  
Public CN As Connection  
Private Type GUID  
    Data1 As Long  
    Data2 As Integer  
    Data3 As Integer  
Universitas Sumatera Utara
Data4(7) As Byte
End Type

Private Type PICTDESC
    size     As Long
    Type     As Long
    hBmp     As Long
    hPal     As Long
    Reserved As Long
End Type

Private Type GdiplusStartupInput
    GdiplusVersion           As Long
    DebugEventCallback       As Long
    SuppressBackgroundThread As Long
    SuppressExternalCodecs   As Long
End Type

Private Type PWMFRect16
    Left   As Integer
    Top    As Integer
    Right  As Integer
    Bottom As Integer
End Type

Private Type wmfPlaceableFileHeader
    Key         As Long
    hMf         As Integer
    BoundingBox As PWMFRect16
    Inch        As Integer
    Reserved    As Long
    CheckSum    As Integer
End Type

Private Declare Function CreateCompatibleDC Lib "gdi32" _
    (ByVal hdc As Long) As Long

Private Declare Function OleCreatePictureIndirect Lib "olepro32.dll"_
    (PicDesc As PICTDESC, RefIID As GUID, ByVal fPictureOwnsHandle As _
    Long, IPic As IPicture) As Long

Private Declare Function CreateCompatibleBitmap Lib "gdi32" _
    (ByVal hdc As Long, ByVal nWidth As Long, ByVal nHeight As Long) _
    As Long

Private Declare Function GetDeviceCaps Lib "gdi32" (ByVal hdc As _
    Long, ByVal nIndex As Long) As Long

Private Declare Function PatBlt Lib "gdi32" (ByVal hdc As Long, _
    ByVal X As Long, ByVal Y As Long, ByVal nWidth As Long, _
    ByVal nHeight As Long, ByVal dwRop As Long) As Long

Private Declare Function CreateBitmap Lib "gdi32" (ByVal nWidth As _
    Long, ByVal nHeight As Long, ByVal nPlanes As Long, ByVal nBitCount As Long, lpBits As Any) As Long

Private Declare Function SelectObject Lib "gdi32" _
    (ByVal hdc As Long, ByVal hObject As Long) As Long
Private Declare Function CreateSolidBrush Lib "gdi32" (ByVal crColor As Long) As Long
Private Declare Function DeleteObject Lib "gdi32" (ByVal hObject As Long) As Long
Private Declare Function DeleteDC Lib "gdi32" (ByVal hdc As Long) As Long
Private Declare Function GdipLoadImageFromFile Lib "gdiplus.dll" (ByVal FileName As Long, GpImage As Long) As Long
Private Declare Function GdiplusStartup Lib "gdiplus.dll" (Token As Long, gdipInput As GdiplusStartupInput, GdiplusStartupOutput As Long) As Long
Private Declare Function GdipCreateFromHDC Lib "gdiplus.dll" (ByVal hdc As Long, GpGraphics As Long) As Long
Private Declare Function GdipSetInterpolationMode Lib "gdiplus.dll" (ByVal Graphics As Long, ByVal InterMode As Long) As Long
Private Declare Function GdipDrawImageRectI Lib "gdiplus.dll" (ByVal Graphics As Long, ByVal Img As Long, ByVal X As Long, ByVal Y As Long, ByVal Width As Long, ByVal Height As Long, ByVal X As Long, ByVal Y As Long, ByVal Width As Long, ByVal Height As Long)
Private Declare Function GdipDeleteGraphics Lib "gdiplus.dll" (ByVal Graphics As Long) As Long
Private Declare Function GdipDisposeImage Lib "gdiplus.dll" (ByVal Image As Long) As Long
Private Declare Function GdipCreateBitmapFromHBITMAP Lib "gdiplus.dll" (ByVal hBmp As Long, ByVal hPal As Long, GpBitmap As Long) As Long
Private Declare Function GdipGetImageWidth Lib "gdiplus.dll" (ByVal Image As Long, Width As Long) As Long
Private Declare Function GdipGetImageHeight Lib "gdiplus.dll" (ByVal Image As Long, Height As Long) As Long
Private Declare Function GdipCreateMetafileFromWmf Lib "gdiplus.dll" (ByVal hWmf As Long, ByVal deleteWmf As Long, WmfHeader As wmfPlaceableFileHeader, Metafile As Long) As Long
Private Declare Function GdipCreateMetafileFromEmf Lib "gdiplus.dll" (ByVal hEmf As Long, ByVal deleteEmf As Long, Metafile As Long) As Long
Private Declare Function GdipCreateBitmapFromHICON Lib "gdiplus.dll" (ByVal hIcon As Long, GpBitmap As Long) As Long
Private Declare Function GdipDrawImageRectRectI Lib "gdiplus.dll" (ByVal Graphics As Long, ByVal GpImage As Long, ByVal dstx As Long, ByVal dsty As Long, ByVal dstwidth As Long, ByVal dstheight As Long, ByVal srcx As Long, ByVal srcy As Long, _
ByVal srcwidth As Long, ByVal srcheight As Long, ByVal srcUnit As Long, ByVal imageAttributes As Long, ByVal callback As Long, ByVal callbackData As Long) As Long

Private Declare Sub GdiplusShutdown Lib "gdiplus.dll" (ByVal Token As Long)

Private Const PLANES = 14
Private Const BITSPIXEL = 12
Private Const PATCOPY = &HF00021
Private Const PICTYPE_BITMAP = 1
Private Const InterpolationModeHighQualityBicubic = 7
Private Const GDIP_WMF_PLACEABLEKEY = &H9AC6CDD7
Private Const UnitPixel = 2

Public Function InitGDIPlus() As Long
    Dim Token As Long
    Dim gdipInit As GdiplusStartupInput
gdipInit.GdiplusVersion = 1
    GdiplusStartup Token, gdipInit, ByVal 0&
    InitGDIPlus = Token
End Function

Public Sub FreeGDIPlus(Token As Long)
    GdiplusShutdown Token
End Sub

Public Function LoadPictureGDIPlus(PicFile As String, Optional Width As Long = -1, Optional Height As Long = -1, Optional ByVal BackColor As Long = vbWhite, Optional RetainRatio As Boolean = False) As IPicture
    Dim hdc As Long
    Dim hBitmap As Long
    Dim Img As Long
    If GdipLoadImageFromFile(StrPtr(PicFile), Img) <> 0 Then
        Err.Raise 999, "GDI+ Module", "Error loading picture " & PicFile
        Exit Function
    End If
    If Width = -1 Or Height = -1 Then
        GdipGetImageWidth Img, Width
        GdipGetImageHeight Img, Height
    End If
    InitDC hdc, hBitmap, BackColor, Width, Height
    gdipResize Img, hdc, Width, Height, RetainRatio
    GdipDisposeImage Img
    GetBitmap hdc, hBitmap
    Set LoadPictureGDIPlus = CreatePicture(hBitmap)
End Function

Private Sub InitDC(hdc As Long, hBitmap As Long, BackColor As Long, _
Width As Long, Height As Long)
Dim hBrush As Long
hdc = CreateCompatibleDC(ByVal 0&)
HBitmap = CreateBitmap(Width, Height, GetDeviceCaps(hdc, PLANES), GetDeviceCaps(hdc, BITSPIXEL), ByVal 0&)
hBitmap = SelectObject(hdc, hBitmap)
hBrush = CreateSolidBrush(BackColor)
hBrush = SelectObject(hdc, hBrush)
PatBlt hdc, 0, 0, Width, Height, PATCOPY
DeleteObject SelectObject(hdc, hBrush)
End Sub

Private Sub gdipResize(Img As Long, hdc As Long, Width As Long, Height As Long, Optional RetainRatio As Boolean = False)
Dim Graphics As Long
Dim OrWidth As Long
Dim OrHeight As Long
Dim OrRatio As Double
Dim DesRatio As Double
Dim DestX As Long
Dim DestY As Long
Dim DestWidth As Long
Dim DestHeight As Long
GdipCreateFromHDC hdc, Graphics, GdipSetInterpolationMode Graphics, InterpolationModeHighQualityBicubic
If RetainRatio Then
    GdipGetImageWidth Img, OrWidth
    GdipGetImageHeight Img, OrHeight
    OrRatio = OrWidth / OrHeight
    DesRatio = Width / Height
    DestWidth = IIf(DesRatio < OrRatio, Width, Height * OrRatio)
    DestHeight = IIf(DesRatio < OrRatio, Width / OrRatio, Height)
    DestX = (Width - DestWidth) / 2
    DestY = (Height - DestHeight) / 2
    GdipDrawImageRectRectI Graphics, Img, DestX, DestY, DestWidth, DestHeight, 0, 0, 0, 0, UnitPixel, 0, 0,
Else
    GdipDrawImageRectI Graphics, Img, 0, 0, Width, Height
End If
GdipDeleteGraphics Graphics
End Sub

Private Sub GetBitmap(hdc As Long, hBitmap As Long)
hBitmap = SelectObject(hdc, hBitmap)
DeleteDC hdc
End Sub

Private Function CreatePicture(hBitmap As Long) As IPicture
Dim IID_IDispatch As GUID

Dim Pic As PICTDESC
Dim IPic As IPicture
IID_IDispatch.Data1 = &H20400
IID_IDispatch.Data4(0) = &HC0
IID_IDispatch.Data4(7) = &H46
Pic.size = Len(Pic)
Pic.Type = PICTYPE_BITMAP
Pic.hBmp = hBitmap
OleCreatePictureIndirect Pic, IID_IDispatch, True, IPic
Set CreatePicture = IPic
End Function

Public Function Resize(Handle As Long, PicType As PictureTypeConstants, Width As Long, Height As Long, Optional BackColor As Long = vbWhite, Optional RetainRatio As Boolean = False) As IPicture
Dim Img As Long
Dim hdc As Long
Dim hBitmap As Long
Dim WmfHeader As wmfPlaceableFileHeader
Select Case PicType
Case vbPicTypeBitmap
GdipCreateBitmapFromHBITMAP Handle, ByVal 0&, Img
Case vbPicTypeMetafile
FillInWmfHeader WmfHeader, Width, Height
GdipCreateMetafileFromWmf Handle, False, WmfHeader, Img
Case vbPicTypeEMetafile
GdipCreateMetafileFromEmf Handle, False, Img
Case vbPicTypeIcon
GdipCreateBitmapFromHICON Handle, Img
End Select
If Img Then
InitDC hdc, hBitmap, BackColor, Width, Height
gdipResize Img, hdc, Width, Height, RetainRatio
GdipDisposeImage Img
GetBitmap hdc, hBitmap
Set Resize = CreatePicture(hBitmap)
End If
End Function

Private Sub FillInWmfHeader(WmfHeader As wmfPlaceableFileHeader, Width As Long, Height As Long)
WmfHeader.BoundingBox.Right = Width
WmfHeader.BoundingBox.Bottom = Height
WmfHeader.Inch = 1440
WmfHeader.Key = GDIP_WMF_PLACEABLEKEY
End Sub

Public Sub Koneksi()
Set CN = New ADODB.Connection
CN.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source= & App.Path & ":dbDisparity.mdb;Persist Security Info=False"
CN.Open
End Sub
Lampiran B

HASIL KUESIONER PENELITIAN
PENGARUH DISPARITY IMAGE TERHADAP EFEK VIRTUAL YANG DIHASILKAN PADA CITRA STEREO

Dari satu kelompok gambar, responden diminta untuk memilih satu gambar yang menghasilkan efek 3D paling nyaman dilihat tanpa menyebabkan pusing dan sakit mata (efek virtual yang optimum).

Objek 1 Guci Buah

Objek 2 Bunga
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

Tabel 1 Hasil Kuesioner