PRAKTIKUM II

Pre Processing

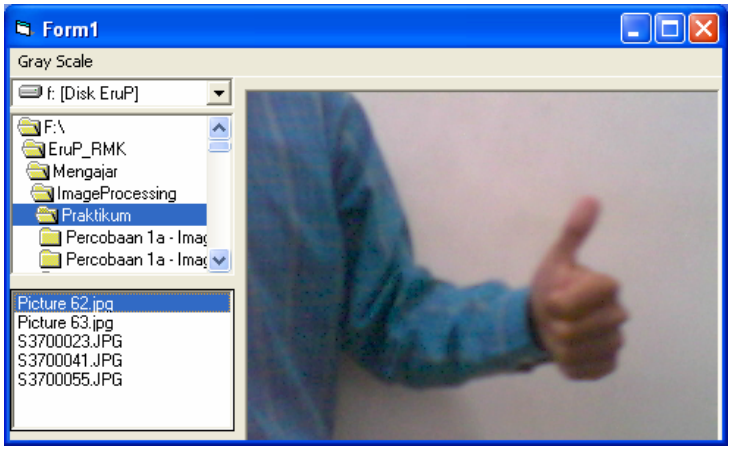
Tujuan

- Memberikan gambaran jenis-jenis pre-processing

Teori Penunjang

Prosedur Percobaan

1. Berikut ini percobaan konversi dari gambar RGB menjadi Gray Scale.
2. Gunakan proyek dari program yang telah dibuat untuk percobaan sebelumnya, yaitu percobaan penampilan gambar menggunakan SetPixel. Salin proyek tersebut ke direktori lain agar dapat dipisahkan dari proyek sebelumnya.



3. Tambahkan menu ”Gray Scale” dengan nama mnuGrayScale.
4. Masukkan program berikut pada file Module1

```
Option Explicit

Public Declare Function SetPixel Lib "gdi32" ( _
    ByVal hdc As Long, ByVal x As Long, ByVal y As Long, _
    ByVal crColor As Long) As Long
Public Declare Function GetPixel Lib "gdi32" ( _
    ByVal hdc As Long, ByVal x As Long, ByVal y As Long) As Long
```

5. Masukkan program berikut pada file Form1

```
Option Explicit

Private Sub Dir1_Change()
    File1.Path = Dir1.Path
End Sub

Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub

Private Sub File1_Click()
    Picture1.Picture = LoadPicture(File1.Path + "\" + File1.FileName)
End Sub

Private Sub Form_Load()
    File1.Pattern = "*.bmp;*.jpg;*.jpeg;*.gif;*.tif"
    Picture1.ScaleMode = 3
    Picture1.AutoSize = True
    Picture1.AutoRedraw = True
End Sub

Private Sub mnuGrayScale_Click()
    Dim x As Integer, y As Integer
    Dim p As Long
    Dim R As Integer, G As Integer, B As Integer
    Dim Gray As Integer
    For y = 0 To Picture1.ScaleHeight - 1
        For x = 0 To Picture1.ScaleWidth - 1
            p = GetPixel(Picture1.hdc, x, y)
            R = p And &HFF
            G = (p \ &H100) And &HFF
            B = (p \ &H10000) And &HFF
            Gray = (R + G + B) / 3
            If Gray > 255 Then Gray = 255
            SetPixel Picture1.hdc, x, y, RGB(Gray, Gray, Gray)
        Next
    Next
    Picture1.Refresh
End Sub
```

6. Jalankan program dan pilih gambar yang akan diproses
7. Tekan menu ”Gray Scale” untuk mulai memproses gambar.
8. Coba jadikan project tersebut menjadi file *.exe dan jalankan kembali. Bandingkanlah kecepatan antara menjalankan melalui IDE VB dengan file *.exe.
9. Ubah formula untuk konversi gray scale dengan formulasi lainnya, yaitu

   $$Gray = 0{,}299R + 0{,}587G + 0{,}114B$$

   $$Gray = \sqrt{\frac{R^2 + G^2 + B^2}{3}}$$

10. Berikut ini percobaan yang sama tetapi dengan menggunakan cara copy buffer image.
11. Masukkan program berikut ini pada Module1

```
Option Explicit

' Deklarasi Jenis type Data RGB, untuk keperluan Image Processing
Public Type tRGB24
```

```
    B As Byte
    G As Byte
    R As Byte
End Type

Public Type BITMAP '14 bytes
    bmType As Long
    bmWidth As Long
    bmHeight As Long
    bmWidthBytes As Long
    bmPlanes As Integer
    bmBitsPixel As Integer
    bmBits As Long
End Type

Public Declare Function GetObject Lib "gdi32" Alias "GetObjectA" ( _
    ByVal hObject As Long, ByVal nCount As Long, lpObject As Any) As Long
Public Declare Sub RtlMoveMemory Lib "kernel32" ( _
    ByVal hpvDest As Long, ByVal hpvSource As Long, ByVal cbCopy As Long)

Global RGB8() As Byte
Global RGB16() As Integer
Global RGB24()  As tRGB24
Global RGB32() As Long
```

12. Masukkan program berikut ini pada Form1

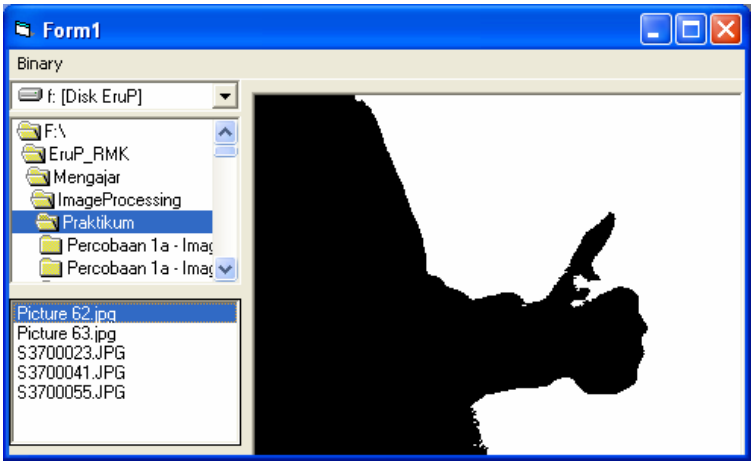
```
Option Explicit

Private Sub Dir1_Change()
    File1.Path = Dir1.Path
End Sub

Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub

Private Sub File1_Click()
    Picture1.Picture = LoadPicture(File1.Path + "\" + File1.FileName)
End Sub

Private Sub Form_Load()
    File1.Pattern = "*.bmp;*.jpg;*.jpeg;*.gif;*.tif"
    Picture1.ScaleMode = 3
    Picture1.AutoSize = True
End Sub

Private Sub mnuGrayScale_Click()
    Dim x As Integer, y As Integer
    Dim p As Long
    Dim R As Integer, G As Integer, B As Integer
    Dim Gray As Integer
    Dim bmp As BITMAP, u As Long
    Dim L As Integer
    GetObject Picture1.Picture.Handle, Len(bmp), bmp
    u = bmp.bmWidthBytes * bmp.bmHeight
    L = bmp.bmBitsPixel \ 8
    Select Case L
        Case 1
        Case 2
        Case 3
            ReDim RGB24(0 To bmp.bmWidth - 1, 0 To bmp.bmHeight - 1) As tRGB24
            RtlMoveMemory VarPtr(RGB24(0, 0)), bmp.bmBits, u
            For y = 0 To Picture1.ScaleHeight - 1
                For x = 0 To Picture1.ScaleWidth - 1
                    R = RGB24(x, y).R
                    G = RGB24(x, y).G
                    B = RGB24(x, y).B
                    Gray = (R + G + B) / 3
```

```
                    If Gray > 255 Then Gray = 255
                    RGB24(x, y).R = Gray
                    RGB24(x, y).G = Gray
                    RGB24(x, y).B = Gray
                Next
            Next
            RtlMoveMemory bmp.bmBits, VarPtr(RGB24(0, 0)), u
        Case 4
    End Select
    Picture1.Refresh
End Sub
```

13. Berikut ini percobaan konversi gambar dari RGB ke Biner.
14. Tambahkan menu ”Binary” dengan nama mnuBinary pada form sebelumnya, dan tambahkan program berikut.

```
Private Sub mnuBinary_Click()
    Dim x As Integer, y As Integer
    Dim p As Long
    Dim R As Integer, G As Integer, B As Integer
    Dim Gray As Integer
    For y = 0 To Picture1.ScaleHeight - 1
        For x = 0 To Picture1.ScaleWidth - 1
            p = GetPixel(Picture1.hdc, x, y)
            R = p And &HFF
            G = (p \ &H100) And &HFF
            B = (p \ &H10000) And &HFF
            Gray = (R + G + B) / 3
            If Gray > 128 Then Gray = 255 Else Gray = 0
            SetPixel Picture1.hdc, x, y, RGB(Gray, Gray, Gray)
        Next
    Next
    Picture1.Refresh
End Sub
```

15. Coba ubah-ubah nilai dari threshod yang digunakan.



16. Percobaan berikut digunakan untuk melakukan pengaturan keseimbangan warna putih.
17. Tambahkan menu ”White Balance” dengan nama mnuWhiteBalance pada form sebelumnya.
18. Lakukan beberapa tambahan dan modifikasi program pada file Form1.
19. Uji dengan membuka suatu gambar tertentu.

20. Klik menu ”White Balance” dan jika muncul peringatan, klik OK.
21. Klik pada daerah gambar yang dapat dianggap sebagai referensi warna putih (gambar tersebut seharusnya warna putih).

```
Option Explicit

Dim pR As Integer, pG As Integer, pB As Integer
Dim SelesaiWB As Boolean

Private Sub Dir1_Change()
    File1.Path = Dir1.Path
End Sub

Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub

Private Sub File1_Click()
    Picture1.Picture = LoadPicture(File1.Path + "\" + File1.FileName)
End Sub

Private Sub Form_Load()
    File1.Pattern = "*.bmp;*.jpg;*.jpeg;*.gif;*.tif"
    Picture1.ScaleMode = 3
    Picture1.AutoSize = True
    Picture1.AutoRedraw = True
End Sub

Private Sub mnuWhiteBalance_Click()
    Dim X As Integer, Y As Integer
    Dim p As Long
    Dim R As Integer, G As Integer, B As Integer
    Dim Gray As Integer
    MsgBox "Klik pada pixel yang dianggap berwarna putih"
    SelesaiWB = False
    While Not SelesaiWB
        DoEvents
    Wend
    For Y = 0 To Picture1.ScaleHeight - 1
        For X = 0 To Picture1.ScaleWidth - 1
            p = GetPixel(Picture1.hdc, X, Y)
            R = (p And &HFF) + pR
            G = ((p \ &H100) And &HFF) + pG
            B = ((p \ &H10000) And &HFF) + pB
            If R > 255 Then R = 255
            If R < 0 Then R = 0
            If G > 255 Then G = 255
            If G < 0 Then G = 0
            If B > 255 Then B = 255
            If B < 0 Then B = 0
            SetPixel Picture1.hdc, X, Y, RGB(R, G, B)
        Next
    Next
    Picture1.Refresh
End Sub

Private Sub Picture1_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)
    Dim p As Long
    Dim Max As Integer
    p = GetPixel(Picture1.hdc, X, Y)
    pR = p And &HFF
    pG = (p \ &H100) And &HFF
    pB = (p \ &H10000) And &HFF
    If pR > pG Then Max = pR Else Max = pG
    If pB > Max Then Max = pB
```

```
    pR = Max - pR
    pG = Max - pG
    pB = Max - pB
    SelesaiWB = True
End Sub
```

Tugas

1. Buat program untuk melakukan filtering menggunakan cara LPF dan smoothing. Gunakan program seperti yang ada pada percobaan Robot Vision
2. Buat program untuk melakukan proses pengubahan ukuran pixel dengan menggunakan cara resampling dan kuantisasi. Gunakan program seperti yang ada pada percobaan Robot Vision.