

Tujuan

- Memberikan gambaran jenis-jenis pre-processing

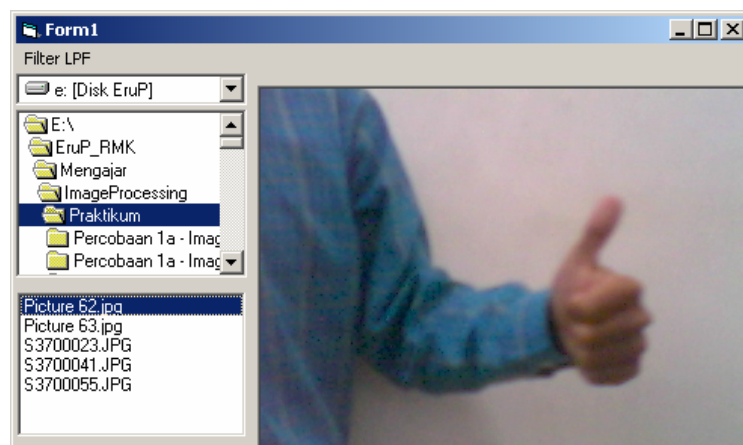
Teori Penunjang

Prosedur Percobaan

- Pada percobaan beikut ini merupakan kelanjutan dari percobaan berikutnya mengenai pre-processing
- Berikut ini percobaan filter LPF menggunakan kernel 5 x 5 dengan nilai matriks

$$Kernel = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- 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.



4. Tambahkan menu "Filter LPF" dengan nama mnuLPF.
5. Masukkan program berikut pada file Module1. Pada program tersebut digunakan buffer vImage dengan ukuran dinamis (ukuran array belum ditentukan) dan tipe data tRGB24.

```
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 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

Global vImage() As tRGB24
```

6. 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 mnuLPF_Click()
    Dim x As Integer, y As Integer
    Dim xx As Integer, yy As Integer
    Dim p As Long
    Dim R As Integer, G As Integer, B As Integer
    Dim Gray As Integer
    Dim Kernel As Integer
    Kernel = 5
    ReDim vImage(-Kernel \ 2 To Picture1.ScaleWidth + Kernel \ 2 - 1, _
        -Kernel \ 2 To Picture1.ScaleHeight + Kernel \ 2 - 1) As tRGB24
    For y = 0 To Picture1.ScaleHeight - 1
        For x = 0 To Picture1.ScaleWidth - 1
            p = GetPixel(Picture1.hdc, x, y)
            vImage(x, y).R = p And &HFF
            vImage(x, y).G = (p \ &H100) And &HFF
            vImage(x, y).B = (p \ &H10000) And &HFF
        Next
    Next
    For y = 0 To Picture1.ScaleHeight - 1
        For x = 0 To Picture1.ScaleWidth - 1
            R = 0
```

```

    G = 0
    B = 0
    For yy = -Kernel \ 2 To Kernel \ 2
        For xx = -Kernel \ 2 To Kernel \ 2
            R = R + vImage(x + xx, y + yy).R
            G = G + vImage(x + xx, y + yy).G
            B = B + vImage(x + xx, y + yy).B
        Next
    Next
    R = R \ (Kernel * Kernel)
    G = G \ (Kernel * Kernel)
    B = B \ (Kernel * Kernel)
    If R > 255 Then R = 255
    If G > 255 Then G = 255
    If B > 255 Then B = 255
    SetPixel Picture1.hdc, x, y, RGB(R, G, B)
Next
Next
Picture1.Refresh
MsgBox "Selesai"
End Sub

```

7. Mengingat program tersebut memerlukan waktu komputasi yang cukup berat, maka sebaiknya anda menjadikan program tersebut ke dalam bentuk *.exe dan menjalankan dari file *.exe untuk mempercepat waktu proses.
8. Jalankan program dan pilih gambar yang akan diproses
9. Tekan menu "Filter LPF" untuk mulai memproses gambar.
10. Ubah ukuran kernel menjadi 3, 7, 9 dan 11. Perhatikan, bahwa ukuran kernel harus ganjil.
11. Berikut ini adalah bentuk lain dari filter menggunakan fungsi smoothing.
12. Buat proyek baru dan dapat menyalin dari proyek sebelumnya. Tambahkan menu "Smoothing" dengan nama "mnuSmoothing".
13. Nilai smoothing ditentukan berdasarkan persamaan:

$$y(n) = \alpha x(n) + (1 - \alpha) y(n - 1)$$
, dengan nilai koefisien smoothing $\alpha = 0,5$ akan didapatkan bentuk

$$y(n) = \frac{x(n) + y(n - 1)}{2}$$
14. Program pada Module1 dapat menggunakan program pada percobaan sebelumnya.
15. Program pada Form1 juga dapat menggunakan program pada percobaan sebelumnya dengan menambahkan program smooting sebagai berikut.

```

Private Sub mnuSmoothing_Click()
    Dim x As Integer, y As Integer
    Dim xx As Integer, yy As Integer
    Dim p As Long
    Dim R As Integer, G As Integer, B As Integer
    ReDim vImage(0 To Picture1.ScaleWidth - 1, _
        0 To Picture1.ScaleHeight - 1) As tRGB24
    For y = 0 To Picture1.ScaleHeight - 1
        For x = 0 To Picture1.ScaleWidth - 1
            p = GetPixel(Picture1.hdc, x, y)

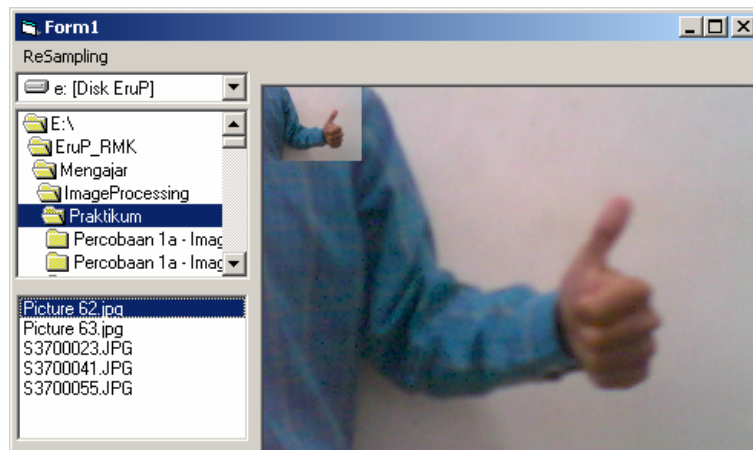
```

```

        vImage(x, y).R = p And &HFF
        vImage(x, y).G = (p \ &H100) And &HFF
        vImage(x, y).B = (p \ &H10000) And &HFF
    Next
Next
For y = 0 To Picture1.ScaleHeight - 1
    R = 0
    G = 0
    B = 0
    For x = 0 To Picture1.ScaleWidth - 1
        R = (R + vImage(x, y).R) / 2
        G = (G + vImage(x, y).G) / 2
        B = (B + vImage(x, y).B) / 2
        vImage(x, y).R = R
        vImage(x, y).G = G
        vImage(x, y).B = B
    Next
Next
For x = 0 To Picture1.ScaleWidth - 1
    R = 0
    G = 0
    B = 0
    For y = 0 To Picture1.ScaleHeight - 1
        R = (R + vImage(x, y).R) / 2
        G = (G + vImage(x, y).G) / 2
        B = (B + vImage(x, y).B) / 2
        vImage(x, y).R = R
        vImage(x, y).G = G
        vImage(x, y).B = B
    Next
Next
For y = 0 To Picture1.ScaleHeight - 1
    For x = 0 To Picture1.ScaleWidth - 1
        SetPixel Picture1.hdc, x, y, _
            RGB(vImage(x, y).R, vImage(x, y).G, vImage(x, y).B)
    Next
Next
Picture1.Refresh
MsgBox "Selesai"
End Sub

```

16. Jalankan program tersebut dan bandingkan kecepatan prosesnya dengan filter LPF.
17. Ubah koefisien smoothing dengan nilai 0,1, 0,2, 0,8 dan coba lagi hasilnya.
18. Berikut ini percobaan mengubah ukuran gambar dengan cara resampling.
 Tambahkan menu "ReSampling" dengan nama mnuReSampling.



19. Gunakan program module1 dan form1 seperti pada percobaan sebelumnya dengan menambahkan program berikut pada form1.

```
Private Sub mnuReSampling_Click()  
    Dim x As Integer, y As Integer  
    Dim p As Long  
    Dim ReSampling As Integer  
    ReSampling = 5  
    For y = 0 To Picture1.ScaleHeight - 1 Step ReSampling  
        For x = 0 To Picture1.ScaleWidth - 1 Step ReSampling  
            p = GetPixel(Picture1.hdc, x, y)  
            SetPixel Picture1.hdc, x \ ReSampling, y \ ReSampling, p  
        Next  
    Next  
    Picture1.Refresh  
    MsgBox "Selesai"  
End Sub
```

20. Coba program tersebut dan ubah-ubah nilai resampling untuk mendapatkan ukuran yang berbeda.
21. Pada percobaan tersebut gambar asli masih tampak. Coba lakukan perubahan program agar hanya menampilkan gambar hasil saja dan gambar asli dapat dibuang.
22. Percobaan berikut digunakan untuk melakukan pengubahan ukuran gambar dengan cara lainnya, yaitu dengan melakukan kuantisasi. Buat proyek baru dengan menyalin program sebelumnya.
23. Tambahkan menu "Quantization" dengan nama mnuQuantization pada form tersebut.
24. Masukkan sub program berikut pada form1.

```
Private Sub mnuQuantization_Click()  
    Dim x As Integer, y As Integer  
    Dim xx As Integer, yy As Integer  
    Dim p As Long  
    Dim R As Integer, G As Integer, B As Integer  
    Dim Quantization As Integer  
    Quantization = 5  
    ReDim vImage(0 To Picture1.ScaleWidth - 1, _  
        0 To Picture1.ScaleHeight - 1) As tRGB24  
    For y = 0 To Picture1.ScaleHeight - 1  
        For x = 0 To Picture1.ScaleWidth - 1  
            p = GetPixel(Picture1.hdc, x, y)  
            vImage(x, y).R = p And &HFF  
            vImage(x, y).G = (p \ &H100) And &HFF  
            vImage(x, y).B = (p \ &H10000) And &HFF  
        Next  
    Next  
    For y = 0 To Picture1.ScaleHeight - 1 Step Quantization  
        For x = 0 To Picture1.ScaleWidth - 1 Step Quantization  
            R = 0  
            G = 0  
            B = 0  
            For yy = 0 To Quantization - 1  
                For xx = 0 To Quantization - 1  
                    R = R + vImage(x + xx, y + yy).R  
                    G = G + vImage(x + xx, y + yy).G  
                    B = B + vImage(x + xx, y + yy).B  
                Next  
            Next  
        Next  
    Next
```

```

        R = R \ (Quantization * Quantization)
        G = G \ (Quantization * Quantization)
        B = B \ (Quantization * Quantization)
        If R > 255 Then R = 255
        If G > 255 Then G = 255
        If B > 255 Then B = 255
        SetPixel Picture1.hdc, x \ Quantization, y \ Quantization, _
            RGB(R, G, B)
    Next
Next
Picture1.Refresh
MsgBox "Selesai"
End Sub

```

25. Uji dengan membuka suatu gambar tertentu.
26. Ubah nilai quantization menjadi 2, 10, 20.
27. Bandingkan hasilnya dengan metode resampling.

Tugas

1. Mengapa ukuran kernel dari filter LPF harus bernilai ganjil ?
2. Mengapa ukuran dari buffer vImage harus bersifat dinamis ?
3. Perhatikan pada program LPF, re deklarasi variable vImage menggunakan batasan $-\text{Kernel} \setminus 2$ sampai ukuran Gambar $+ \text{Kernel} \setminus 2 - 1$. Mengapa tidak dari 0 sampai ukuran gambar -1 saja ?
4. Apa perbedaan penggunaan penskalaan menggunakan resampling dengan quantization ? Terangkan dari prinsip kerja, hasil dan kecepatan.