LAPORAN PRAKTIKUM

(PENGOLAHAN CITRA)

(MODUL 9 dan 10)

Disusun oleh :

Nama	**:**	**Azka Ainurridho**
NIM	**:**	**20220810124**
Kelas	**:**	**TINFC 2022 04**

FAKULTAS ILMU KOMPUTER

UNIVERSITAS KUNINGAN

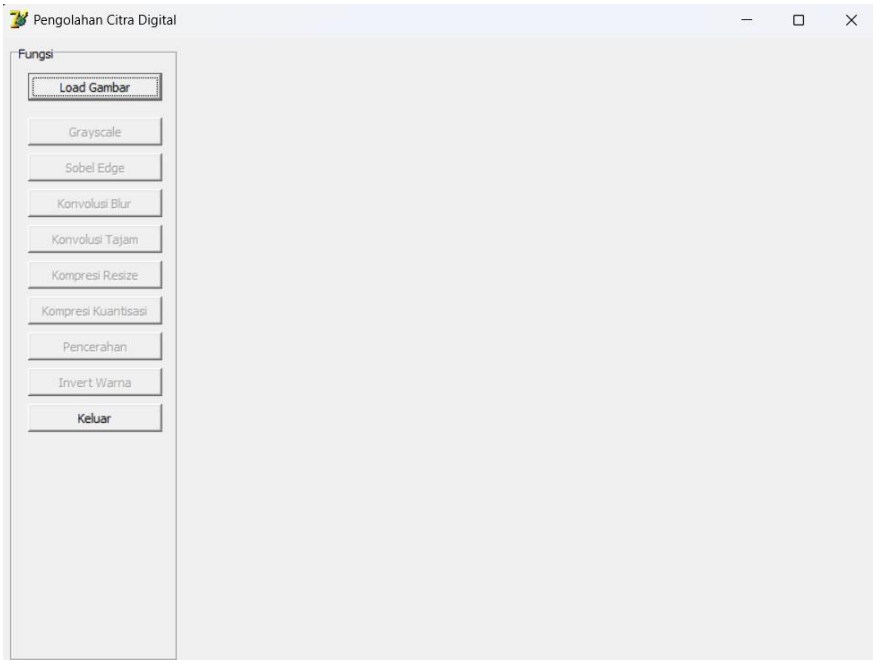
2025/2026

I. Tujuan Praktikum

1. Mahasiswa dapat memahami analisis citra
2. Mahasiswa mampu memahami cara pembuatan program feature extraction

II. PRAKTEK

1. Buka program Aplikasi Delphi
2. Buat Proyek Baru dan beri nama SimpanVideo.dpr
3. Tambahkan Komponen-komponen berikut ke dalam Form1



4. Masukan kode program dibawah ini: Source Code 1:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, Buttons, ExtCtrls, StdCtrls, ExtDlgs;

type
  TForm1 = class(TForm)
    GroupBox1: TGroupBox;
    SpeedButton1: TSpeedButton; // Grayscale
    SpeedButton2: TSpeedButton; // Sobel
    SpeedButton3: TSpeedButton; // Konvolusi
    SpeedButton4: TSpeedButton; // Konvolusi 2
    SpeedButton5: TSpeedButton; // Kompresi 1
    SpeedButton6: TSpeedButton; // Kompresi 2
    SpeedButton7: TSpeedButton; // Treatment
    SpeedButton8: TSpeedButton; // Invert
    SpeedButton9: TSpeedButton; // Exit
```

```
    BitBtn1: TBitBtn;          // Load Image
    Image1: TImage;
    OpenPictureDialog1: TOpenPictureDialog;

    procedure FormCreate(Sender: TObject);
  procedure BitBtn1Click(Sender: TObject);
  procedure SpeedButton1Click(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
  procedure SpeedButton3Click(Sender: TObject);
  procedure SpeedButton4Click(Sender: TObject);
  procedure SpeedButton5Click(Sender: TObject);
  procedure SpeedButton6Click(Sender: TObject);
  procedure SpeedButton7Click(Sender: TObject);
  procedure SpeedButton8Click(Sender: TObject);
  procedure SpeedButton9Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

  var
   Form1: TForm1;
  gambar: TBitmap;

  implementation

  {$R *.dfm}

  function Clamp(Value, MinVal, MaxVal: Integer):
  Integer; begin   if Value < MinVal then Result := MinVal
  else if Value > MaxVal then Result := MaxVal   else
  Result := Value; end;

  procedure TForm1.FormCreate(Sender: TObject);
  begin
   SpeedButton1.Enabled := False;
   SpeedButton2.Enabled := False;
   SpeedButton3.Enabled := False;
   SpeedButton4.Enabled := False;
   SpeedButton5.Enabled := False;
   SpeedButton6.Enabled := False;
  SpeedButton7.Enabled := False;
  SpeedButton8.Enabled := False;




end;
```

```
procedure TForm1.BitBtn1Click(Sender:
TObject); begin   if OpenPictureDialog1.Execute
then   begin     if Assigned(gambar) then
gambar.Free;     gambar := TBitmap.Create;
   gambar.LoadFromFile(OpenPictureDialog1.FileName);
   Image1.Picture.Bitmap := gambar;

   SpeedButton1.Enabled := True;
   SpeedButton2.Enabled := True;
   SpeedButton3.Enabled := True;
   SpeedButton4.Enabled := True;
   SpeedButton5.Enabled := True;
   SpeedButton6.Enabled := True;
   SpeedButton7.Enabled :=
True;     SpeedButton8.Enabled
:= True;   end; end;

procedure TForm1.SpeedButton1Click(Sender:
TObject); var   i, j: Integer;
  temp: PByteArray;   gray: Byte;
begin   if not Assigned(gambar)
then Exit;   gambar.PixelFormat
:= pf24bit;

  for j := 0 to gambar.Height - 1
do   begin     temp :=
gambar.ScanLine[j];
   i := 0;
   while i < 3 * gambar.Width do
begin
     gray := Round(0.3 * temp[i + 2] + 0.59 * temp[i + 1] + 0.11 *
temp[i]);       temp[i] := gray;       temp[i + 1] := gray;       temp[i + 2] :=
gray;       Inc(i, 3);     end;   end;




end;
```

```
Image1.Picture.Bitmap := gambar;
gambar.SaveToFile('Grayscale.bmp'); procedure
TForm1.SpeedButton2Click(Sender: TObject); const
  SobelX: array[0..2, 0..2] of Integer = ((-1, 0, 1), (-2, 0, 2), (-1, 0,
1));   SobelY: array[0..2, 0..2] of Integer = ((-1, -2, -1), (0, 0, 0), (1,
2, 1)); var   y, x, i, j, gx, gy, gray: Integer;   row: array[-1..1] of
PByteArray;   hasil: TBitmap;   line: PByteArray; begin   if not
Assigned(gambar) then Exit;

  hasil := TBitmap.Create;
hasil.Assign(gambar);
hasil.PixelFormat := pf24bit;

  for y := 1 to gambar.Height - 2 do
begin     for i := -1 to 1 do
row[i] := gambar.ScanLine[y + i];

    line := hasil.ScanLine[y];     for x := 1 to
gambar.Width - 2 do     begin       gx := 0;       gy := 0;
for i := -1 to 1 do         for j := -1 to 1 do           gx := gx +
SobelX[i + 1, j + 1] * row[i]^[3 * (x + j)];       for i := -1
to 1 do         for j := -1 to 1 do           gy := gy + SobelY[i
+ 1, j + 1] * row[i]^[3 * (x + j)];

      gray := Abs(gx) + Abs(gy);
gray := Clamp(gray, 0, 255);

      line^[3 * x] := gray;
line^[3 * x + 1] := gray;
line^[3 * x + 2] := gray;
end;   end;

  Image1.Picture.Bitmap := hasil;
hasil.SaveToFile('Sobel.bmp');
hasil.Free;



end;
```

```
procedure TForm1.SpeedButton3Click(Sender: TObject);
const
  Kernel: array[0..2, 0..2] of Integer = (
    (1, 1, 1),
    (1, 1, 1),
    (1, 1, 1)   ); var   x, y, i, j, r, g,
b: Integer;   row: array[-1..1] of
PByteArray;   hasil: TBitmap;
line: PByteArray; begin   hasil
:= TBitmap.Create;
hasil.Assign(gambar);
hasil.PixelFormat := pf24bit;

  for y := 1 to gambar.Height - 2 do
begin     for i := -1 to 1 do
row[i] := gambar.ScanLine[y + i];

    line := hasil.ScanLine[y];

    for x := 1 to gambar.Width - 2 do     begin       r := 0; g
:= 0; b := 0;       for i := -1 to 1 do         for j := -1 to 1 do
begin           r := r + row[i]^[3 * (x + j) + 2] * Kernel[i +
1][j + 1];           g := g + row[i]^[3 * (x + j) + 1] * Kernel[i
+ 1][j + 1];           b := b + row[i]^[3 * (x + j) + 0] *
Kernel[i + 1][j + 1];         end;       r := r div 9;       g := g
div 9;       b := b div 9;

      line^[3 * x + 2] :=
r;       line^[3 * x + 1]
:= g;       line^[3 * x +
0] := b;     end;   end;

  Image1.Picture.Bitmap := hasil;
hasil.SaveToFile('Konvolusi_Blur.bmp');
hasil.Free;
```

```
end;

procedure TForm1.SpeedButton4Click(Sender: TObject);
const
  Kernel: array[0..2, 0..2] of Integer = (
    (0, -1, 0),
    (-1, 5, -1),
    (0, -1, 0)   ); var   x, y, i, j, r,
g, b: Integer;   row: array[-1..1]
of PByteArray;   hasil:
TBitmap;   line: PByteArray;
begin   hasil := TBitmap.Create;
hasil.Assign(gambar);
hasil.PixelFormat := pf24bit;

  for y := 1 to gambar.Height - 2 do
begin     for i := -1 to 1 do
row[i] := gambar.ScanLine[y + i];

    line := hasil.ScanLine[y];

    for x := 1 to gambar.Width - 2 do     begin       r := 0; g
:= 0; b := 0;       for i := -1 to 1 do         for j := -1 to 1 do
begin           r := r + row[i]^[3 * (x + j) + 2] * Kernel[i +
1][j + 1];           g := g + row[i]^[3 * (x + j) + 1] * Kernel[i
+ 1][j + 1];           b := b + row[i]^[3 * (x + j) + 0] *
Kernel[i + 1][j + 1];         end;       r := Clamp(r, 0, 255);
g := Clamp(g, 0, 255);       b := Clamp(b, 0, 255);

      line^[3 * x + 2] :=
r;       line^[3 * x + 1]
:= g;       line^[3 * x +
0] := b;     end;   end;

Image1.Picture.Bitmap := hasil;
hasil.SaveToFile('Konvolusi_Sharpen.bmp');   hasil.Free;
end;

procedure TForm1.SpeedButton5Click(Sender:
TObject); var   kecil: TBitmap; begin   kecil :=
TBitmap.Create;   kecil.Width := gambar.Width div 2;
kecil.Height := gambar.Height div 2;
kecil.PixelFormat := pf24bit;

  kecil.Canvas.StretchDraw(Rect(0, 0, kecil.Width, kecil.Height),
gambar);
```

```
  Image1.Picture.Bitmap := kecil;




kecil.SaveToFile('Kompresi_Resize.bmp');
kecil.Free; end;
```

```
procedure TForm1.SpeedButton6Click(Sender: TObject);
var   i, j: Integer;   p: PByteArray; begin   for j := 0 to
gambar.Height - 1 do   begin     p := gambar.ScanLine[j];
for i := 0 to gambar.Width - 1 do     begin       p[i * 3] :=
(p[i * 3] div 64) * 64;       p[i * 3 + 1] := (p[i * 3 + 1] div
64) * 64;       p[i * 3 + 2] := (p[i * 3 + 2] div 64) * 64;
end;   end;
```

```
  Image1.Picture.Bitmap := gambar;
gambar.SaveToFile('Kompresi_Kuantisasi.bmp'); end;

procedure TForm1.SpeedButton7Click(Sender:
TObject); var   i, j: Integer;   p: PByteArray;   bright:
Integer; begin
bright := 40;
```

```
  for j := 0 to gambar.Height - 1 do   begin     p :=
gambar.ScanLine[j];     for i := 0 to gambar.Width - 1
do     begin       p[i * 3] := Clamp(p[i * 3] + bright, 0,
255);       p[i * 3 + 1] := Clamp(p[i * 3 + 1] + bright, 0,
255);       p[i * 3 + 2] := Clamp(p[i * 3 + 2] + bright, 0,
255);     end;   end;

  Image1.Picture.Bitmap := gambar;
gambar.SaveToFile('Treatment_Pencerahan.bmp'); end;

procedure TForm1.SpeedButton8Click(Sender: TObject);
var   i, j: Integer;   p: PByteArray; begin   for j := 0 to
gambar.Height - 1 do   begin     p := gambar.ScanLine[j];
for i := 0 to gambar.Width - 1 do     begin       p[i * 3] :=
255 - p[i * 3];       p[i * 3 + 1] := 255 - p[i * 3 + 1];       p[i
* 3 + 2] := 255 - p[i * 3 + 2];     end;   end;

  Image1.Picture.Bitmap := gambar;
gambar.SaveToFile('Invert.bmp'); end;

procedure TForm1.SpeedButton9Click(Sender: TObject);
begin   Close; end;

end.
```

Source Code 2:

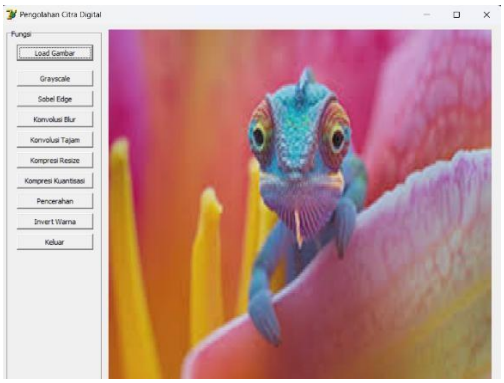
```
object Form1: TForm1
  Left = 200
  Top = 100
  Width = 800
  Height = 600
  Caption = 'Pengolahan Citra Digital'
  Color = clBtnFace
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'Tahoma'
  Font.Style = []
  OldCreateOrder = False
  OnCreate = FormCreate
  PixelsPerInch = 96
TextHeight = 13
  object GroupBox1: TGroupBox
    Left = 8
    Top = 8
    Width = 150
    Height = 550
    Caption = 'Fungsi'
TabOrder = 0
    object BitBtn1: TBitBtn
      Left = 16
      Top = 24
      Width = 120
      Height = 25
      Caption = 'Load Gambar'
      TabOrder = 0
      OnClick = BitBtn1Click
end
    object SpeedButton1: TSpeedButton
      Left = 16
      Top = 64
      Width = 120
      Height = 25
      Caption = 'Grayscale'
      OnClick = SpeedButton1Click
end
    object SpeedButton2: TSpeedButton
      Left = 16
      Top = 96
      Width = 120
      Height = 25
      Caption = 'Sobel Edge'
```

```
    OnClick = SpeedButton2Click
end    object SpeedButton3:
TSpeedButton
    Left = 16
    Top = 128
    Width = 120
    Height = 25
    Caption = 'Konvolusi Blur'
OnClick = SpeedButton3Click    end
object SpeedButton4: TSpeedButton
    Left = 16
    Top = 160
    Width = 120
    Height = 25
    Caption = 'Konvolusi Tajam'
OnClick = SpeedButton4Click    end
object SpeedButton5: TSpeedButton
    Left = 16
    Top = 192
    Width = 120
    Height = 25
    Caption = 'Kompresi Resize'
OnClick = SpeedButton5Click    end
object SpeedButton6: TSpeedButton
    Left = 16
    Top = 224
    Width = 120
    Height = 25
    Caption = 'Kompresi Kuantisasi'
OnClick = SpeedButton6Click    end
object SpeedButton7: TSpeedButton
    Left = 16
    Top = 256
    Width = 120
    Height = 25
    Caption = 'Pencerahan'
OnClick = SpeedButton7Click    end
object SpeedButton8: TSpeedButton
    Left = 16
    Top = 288
    Width = 120
    Height = 25
```

```
      Caption = 'Invert Warna'
      OnClick = SpeedButton8Click
end
    object SpeedButton9: TSpeedButton
      Left = 16
      Top = 320
      Width = 120
      Height = 25
      Caption = 'Keluar'
      OnClick = SpeedButton9Click
end   end
  object Image1: TImage
    Left = 168
    Top = 8
    Width = 600
    Height = 550
Stretch = True    end
  object OpenPictureDialog1: TOpenPictureDialog
    Filter = 'Bitmap (*.bmp)|*.bmp|JPEG (*.jpg)|*.jpg;*.jpeg|PNG (*.png)|*.png|All
Files (*.*)|*.*'
    Left = 600
Top = 560
end end
```

III. Hasil Output :

1. Normal



2. Grayscale

Tujuan:
Deteksi tepi (edge detection), khususnya gradien perubahan intensitas pixel.
Cara kerja:
Menggunakan dua kernel konvolusi:

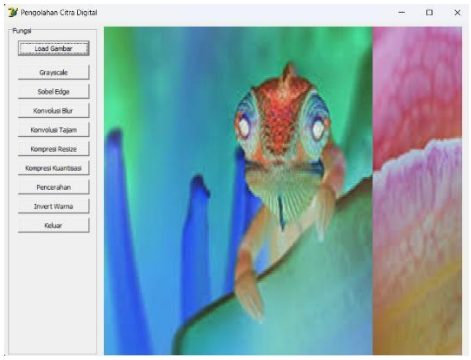
a. SobelX: mendeteksi tepi vertical
b. SobelY: mendeteksi tepi horizontal

Nilai akhir adalah penjumlahan absolut dari kedua gradien:
|Gx| + |Gy|

Manfaat:

- Menyoroti batas objek
- Penting dalam segmentasi dan pelacakan bentuk

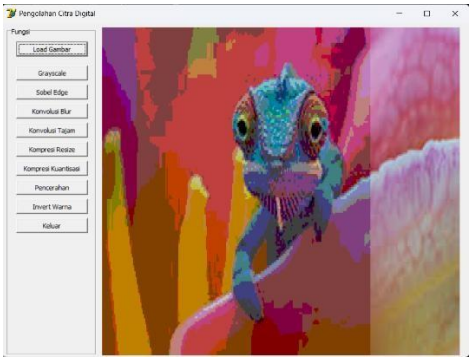
3. Invert



Tujuan:
Membalikkan warna gambar (efek negatif).
Cara kerja:
Setiap kanal warna dikurangi dari 255:
p[i] := 255 - p[i] Manfaat:

- Digunakan untuk efek visual
- Kadang berguna untuk menyoroti fitur tersembunyi dalam gambar

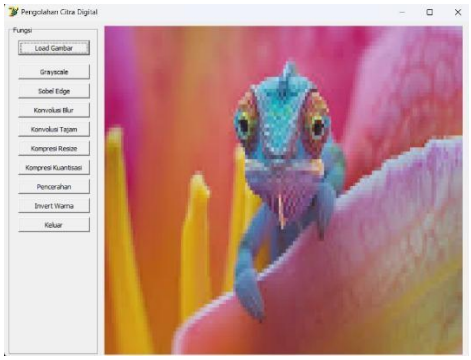
4. Kompresi Kuantisasi



Tujuan:
Mengurangi jumlah warna (depth) dalam gambar.
Cara kerja:
Membulatkan setiap kanal RGB ke kelipatan 64:
p[i] := (p[i] div 64) * 64 Manfaat:

- Menghemat penyimpanan
- Digunakan dalam teknik kompresi lossy seperti JPEG

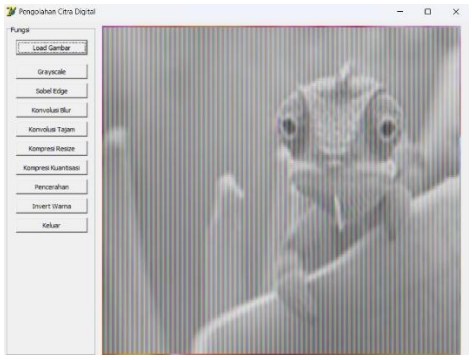
5. Kompresi Resize



Tujuan:
Mengurangi dimensi gambar (width dan height) untuk kompresi data visual.
Cara kerja:
Menggunakan StretchDraw untuk memperkecil ukuran gambar menjadi separuhnya.
Manfaat:

- Mengurangi ukuran file gambar
- Cocok untuk preview, thumbnails, atau pengiriman cepat

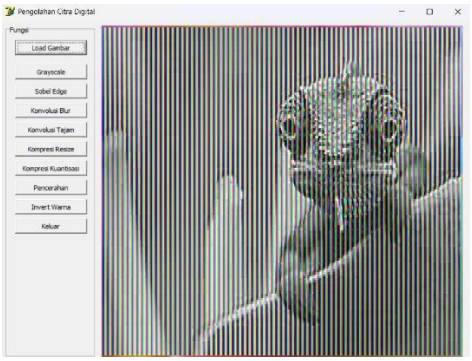
6. Konvolusi Blur



Tujuan:
Melakukan perataan (smoothing) atau mengaburkan gambar (blur).
Cara kerja:
Kernel 3x3 dengan nilai seragam (semua 1) menghasilkan efek blur ringan:
rata-rata = jumlah 9 pixel sekitarnya / 9 Manfaat:

- Mengurangi noise
- Membantu persiapan sebelum segmentasi atau deteksi tepi

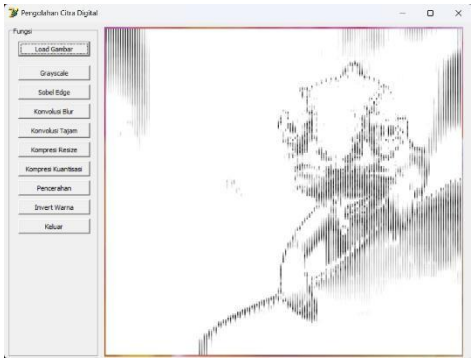
7. Konvolusi Sharpen



Tujuan:
Menajamkan gambar dengan memperkuat kontras tepi.
Cara kerja: Kernel
penajaman:
diff
Copy code
0 -1 0
1 5 -1
0 -1 0
Menekankan pixel tengah dan mengurangi pengaruh tetangga.
Manfaat:

- Menonjolkan fitur penting dalam citra
- Berguna untuk visualisasi atau pre-processing analisis

8. Sobel



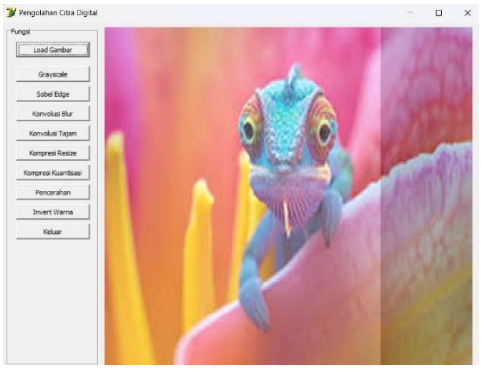
Tujuan:
Deteksi tepi (edge detection), khususnya gradien perubahan intensitas pixel.
Cara kerja:
Menggunakan dua kernel konvolusi:

- SobelX: mendeteksi tepi vertical
- SobelY: mendeteksi tepi horizontal

Nilai akhir adalah penjumlahan absolut dari kedua gradien:
|Gx| + |Gy| Manfaat:

- Menyoroti batas objek
- Penting dalam segmentasi dan pelacakan bentuk

9. Treatment



Tujuan:
Meningkatkan kecerahan gambar.
Cara kerja:
Menambahkan nilai konstan (misal 40) ke setiap kanal warna, dengan Clamp() agar tetap dalam batas 0–255.
Manfaat:

- Membantu visibilitas objek dalam gambar gelap
- Persiapan sebelum proses lain seperti OCR