

**UJIAN PENGOLAHAN CITRA DIGITAL**

**“DATA WAREHOUSE”**

**Dosen Pengampuh : Dwi Shinta Angreni, S.Si., M.Kom**



**DISUSUN OLEH :**

**FEBRIYADI**

**F55121082**

**A**

**PROGRAM STUDI S1-TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS TADULAKO**

**2023**

1. GUI Aplikasi Pengolahan citra metode(gray scale, Thresholding correction, brightness, difference\_image, sharpening, noise\_reduction, median\_filter)

a. Kode Program

```
1  #Nama : febriyadi
2  #Nim : F5521082
3  #Kelas : A Informatika universitas Tadulaka
4
5  import tkinter as tk
6  from tkinter import filedialog
7  from PIL import Image, ImageTk
8  import cv2
9  import numpy as np
10 from scipy.ndimage import gaussian_filter
11
12 # fungsi untuk memproses citra dengan metode Median filter
13
14 def median_filter(img):
15     median_img = cv2.medianBlur(img, 5)
16     return median_img
17
18 # fungsi untuk memproses citra dengan metode difference_image
19 def difference_image(img):
20     # konversi citra ke grayscale
21     gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
22
23     # proses perbaikan citra menggunakan metode Difference Image
24     blur_img = cv2.GaussianBlur(gray_img, (5, 5), 0)
25     diff_img = cv2.absdiff(gray_img, blur_img)
26     _, thresh_img = cv2.threshold(diff_img, 30, 255, cv2.THRESH_BINARY)
27
28     return thresh_img
29
30 # fungsi untuk memproses citra dengan metode grayscale
31 def grayscale(img):
32     gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
33     return gray_img
34
35 # fungsi untuk memperbaiki citra dengan metode thresholding
36 def thresholding_correction(img):
37     gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
38     _, threshold_img = cv2.threshold(gray_img, 0, 255, cv2.THRESH_BINARY+cv2.THRESH_OTSU)
39
40     return threshold_img
41
42 def sharpening(img):
43     kernel = np.array([[ -1, -1, -1], [-1, 9, -1], [-1, -1, -1]])
44     sharpened_img = cv2.filter2D(img, -1, kernel)
45     return sharpened_img
46
47 def noise_reduction(img):
48     denoised_img = cv2.fastNlMeansDenoisingColored(img, None, 10, 10, 7, 21)
49     return denoised_img
50
51 # fungsi untuk memperbaiki citra dengan metode peningkatan kecerahan
52 def brightness_correction(img):
53     brightness = 50
54     corrected_img = cv2.add(img, brightness)
55     return corrected_img
```

```

56 # fungsi untuk menampilkan gambar dalam kotak
57 def show_image(img, x, y, title):
58     img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
59     img = Image.fromarray(img)
60     img = ImageTk.PhotoImage(img)
61     label = tk.Label(root, image=img)
62     label.image = img
63     label.place(x=x, y=y)
64     title_label = tk.Label(root, text=title)
65     title_label.place(x=x, y=y-20)
66
67 # fungsi untuk memproses citra dan menampilkan hasilnya
68 def process_image(method):
69     global original_img
70     if method == 'grayscale':
71         corrected_img = grayscale(original_img)
72         show_image(corrected_img, 360, 170, 'Hasil Metode grayscale')
73     elif method == 'thresholding_correction':
74         corrected_img = thresholding_correction(original_img)
75         show_image(corrected_img, 620, 170, 'Hasil Metode thresholding_correction')
76     elif method == 'brightness':
77         corrected_img = brightness_correction(original_img)
78         show_image(corrected_img, 880, 170, 'Hasil Metode brightness_correction')
79     elif method == 'difference_image':
80         corrected_img = difference_image(original_img)
81         show_image(corrected_img, 70, 450, 'Hasil Metode difference image')
82     elif method == 'sharpening':
83         corrected_img = sharpening(original_img)
84         show_image(corrected_img, 360, 450, 'Hasil Metode sharpening image')
85     elif method == 'noise_reduction':
86         corrected_img = noise_reduction(original_img)
87         show_image(corrected_img, 620, 450, 'Hasil Metode noise_reduction image')
88     elif method == 'median_filter':
89         corrected_img = median_filter(original_img)
90         show_image(corrected_img, 880, 450, 'Hasil Metode median_filter image')
91
92
93 # fungsi untuk menampilkan informasi pembuat program
94 def show_creator():
95     creator_label = tk.Label(root, text='NAMA : FEBRIYADI | NIM : F55121082 | KELAS : A')
96     creator_label.place(x=480, y=100)
97
98 # fungsi untuk membuka gambar
99 def open_image():
100     global original_img
101     file_path = filedialog.askopenfilename()
102     if file_path:
103         original_img = cv2.imread(file_path)
104         original_img = cv2.resize(original_img, (250, 250))
105         show_image(original_img, 70, 170, 'Gambar Original')
106         size_label.config(format(original_img.shape[1], original_img.shape[0]))
107
108 # membuat jendela utama
109 root = tk.Tk()
110 root.geometry('1200x900')
111 root.title('GUI Aplikasi Pengolahan Citra')
112
113 # menambahkan judul gambar original
114 title_label = tk.Label(root, text='Original image')
115 title_label.place(x=50, y=20)

```

```

117 # menambahkan tombol untuk membuka gambar
118 open_button = tk.Button(root, text='Select an image', command=open_image)
119 open_button.place(x=50, y=50)
120
121 # menambahkan kotak untuk metode perbaikan citra
122 correction_box = tk.LabelFrame(root, text='Metode Perbaikan Citra', padx=5, pady=5)
123 correction_box.place(x=50, y=760, width=1100, height=70)
124
125 # tombol untuk metode Transformasi Negatif
126 grayscale_button = tk.Button(correction_box, text='grayscale', command=lambda: process_image('grayscale'))
127 grayscale_button.pack(side=tk.LEFT, padx=5)
128
129 # tombol untuk perbaikan metode smoothing
130 thresholding_correction_button = tk.Button(correction_box, text='thresholding_correction', command=lambda: process_image('thresholding_correction'))
131 thresholding_correction_button.pack(side=tk.LEFT, padx=5)
132
133 # tombol untuk perbaikan metode Peningkatan Kecerahan
134 brightness_button = tk.Button(correction_box, text='Peningkatan Kecerahan', command=lambda: process_image('brightness'))
135 brightness_button.pack(side=tk.LEFT, padx=5)
136
137 # tombol untuk perbaikan metode difference_image
138 difference_image_button = tk.Button(correction_box, text='difference_image', command=lambda: process_image('difference_image'))
139 difference_image_button.pack(side=tk.LEFT, padx=5)
140
141 # tombol untuk perbaikan metode sharpening
142 sharpening_button = tk.Button(correction_box, text='sharpening', command=lambda: process_image('sharpening'))
143 sharpening_button.pack(side=tk.LEFT, padx=5)
144
145 # tombol untuk perbaikan metode noise_reduction
146 noise_reduction_button = tk.Button(correction_box, text='noise_reduction', command=lambda: process_image('noise_reduction'))
147 noise_reduction_button.pack(side=tk.LEFT, padx=5)
148
149 # tombol untuk perbaikan metode noise_reduction
150 median_filter_button = tk.Button(correction_box, text='median_filter', command=lambda: process_image('median_filter'))
151 median_filter_button.pack(side=tk.LEFT, padx=5)
152
153 # menambahkan kotak untuk menampilkan hasil perbaikan citra
154 result_box = tk.LabelFrame(root, text='Output Perbaikan Citra', padx=5, pady=5)
155 result_box.place(x=50, y=100, width=1100, height=650)
156
157 # menambahkan kotak untuk informasi pembuat program
158 creator_box = tk.LabelFrame(root, text='Creator', padx=5, pady=5)
159 creator_box.place(x=460, y=80, width=315, height=60)
160
161 # menampilkan informasi pembuat program
162 show_creator()
163
164 # menjalankan aplikasi
165 root.mainloop()

```

## b. Hasil Aplikasi Pengelola Citra

