

LAPORAN UJIAN PENGOLAHAN CITRA DIGITAL (PCD)

Dosen Pengampu : Shinta Dwi Anggreni, S. Si.,m.Kom

“APLIKASI PERBAIKAN CITRA MENGGUNAKAN INTERFACE/GUI ”



DIBUAT OLEH :

Dendy Kurniawan

F55121088

PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

FAKULTAS TEKNIK

UNIVERSITAS TADULAKO

2023

1. Kode Program

- a. Import package yang akan digunakan

```
import tkinter as tk
from tkinter import filedialog
from PIL import Image, ImageTk
```

- b. Kemudian membuat jendela utama

```
class ImageEditor(tk.Tk):
    def __init__(self):
        super().__init__()

        self.title("Aplikasi Perbaikan Citra")
        self.geometry("500x500")
```

- c. Lalu membuat variabel global untuk objek gambar

```
original_image = None
enhanced_image = None
```

- d. Lalu membuat kode program yang berfungsi untuk membuka file gambar menggunakan dialog file

```
# create labels for displaying the images
self.original_label = tk.Label(self, text="Original Image")
self.original_label.pack(side=tk.LEFT, padx=10, pady=10)

self.processed_label = tk.Label(self, text="Processed Image")
self.processed_label.pack(side=tk.RIGHT, padx=10, pady=10)
```

- e. Mendapatkan jalur file dari pengguna

```
file_path = filedialog.askopenfilename()
```

- f. Memuat gambar dan menampilkannya

```
original_image = Image.open(file_path)
original_image = original_image.convert("RGB")
image = ImageTk.PhotoImage(original_image)
panel1.config(image=image)
panel1.image = image
```

- g. Kemudian menyetel ulang gambar

```
enhanced_image = None
```

- h. Selanjutnya kode untuk menerapkan method 1. Dimana memeriksa apakah gambar dimuat dan menerapkan pada format file gambar

```
def select_image(self):
    # show file dialog to select the image
    file_path = filedialog.askopenfilename(filetypes=[("Image Files", "*.jpg;*.jpeg;*.png;*.bmp")])
    if file_path:
        self.image_path = file_path
        self.image = Image.open(file_path)

        # display the original image
        self.show_image(self.image, self.original_label)

        # enable the image processing buttons
        self.gray_button.config(state=tk.NORMAL)
        self.flip_button.config(state=tk.NORMAL)
```

- i. Untuk membuat dan memproses tampilan dari Grayscale

```
def convert_to_gray(self):
    if self.image:
        # apply grayscale to the image
        gray_image = self.image.convert('L')

        # display the processed image
        self.show_image(gray_image, self.processed_label)

def flip_image(self):
    if self.image:
        # flip the image horizontally
        flipped_image = self.image.transpose(Image.FLIP_LEFT_RIGHT)

        # display the processed image
        self.show_image(flipped_image, self.processed_label)
```

- j. Selanjutnya kode untuk membuat dan menerapkan file menggunakan metode flip

```
def flip_image(self):
    if self.image:
        # flip the image horizontally
        flipped_image = self.image.transpose(Image.FLIP_LEFT_RIGHT)

        # display the processed image
        self.show_image(flipped_image, self.processed_label)

def show_image(self, image, label):
    # resize the image to fit in the label
    width, height = image.size
    ratio = min(1.0, 300.0 / max(width, height))
    new_width = int(ratio * width)
    new_height = int(ratio * height)
    image = image.resize((new_width, new_height), Image.ANTIALIAS)

    # convert the image to PhotoImage and display it in the label
    photo_image = ImageTk.PhotoImage(image)
    label.config(image=photo_image)
    label.image = photo_image
```

- k. Yang terakhir adalah kode untuk menjalankan loop utama

```
if __name__ == '__main__':
    app = ImageEditor()
    app.mainloop()
```

2. Hasil Run

- a. Tombol button



- b. Metode 1, dimana menggunakan filter Grayscale dengan menggunakan method Self Image Convers pada library PIL. Method ini menghasilkan citra gambar hitam putih.



- c. Metode 2, menggunakan filip untuk mengubah distribusi gambar terbalik pada citra.



