

DAFTAR PUSTAKA

- Agus, A. A., & Riskawati. (2016). Penanganan Kasus *Cyber Crime* Di Kota Makassar (Studi pada Kantor Kepolisian Resort Kota Besar Makassar). *Jurnal Supremasi*, 11(1), 20-29.
- Anam, K., Cahyadi, W., Azmi, I., Senjarini, K., & Oktarianti, R. (2021). Analisis Hasil Elektroforesis DNA dengan *Image Processing* Menggunakan Metode *Gaussian Filter*. *Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)*, Vol.11, No.1, 37-48.
- Andono, P. N., Sutojo, T., & Muljono. (2017). Pengolahan Citra Digital. Penerbit Andi.
- Annisa, S., Lubis, Z., & Najmita, A. (2020). Perancangan Aplikasi Jaringan Syaraf Tiruan (*Neural Networks*) untuk Pendekripsi Keaslian Uang Kertas. *Journal of Electrical Technology*, 5(1).
- Dewi, S. N., Cholissodin, I., & Santoso, E. (2018). Prediksi Jumlah Kriminalitas Menggunakan Metode *Extreme Learning Machine* (Studi Kasus di Kabupaten Probolinggo). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIK) Universitas Brawijaya*, 2(11), 4687-4693.
- Effendi, M., Fitriyah, F., & Effendi, U. (2017). Identifikasi Jenis dan Mutu Teh Menggunakan Pengolahan Citra Digital dengan Metode Jaringan Syaraf Tiruan. *Jurnal Teknotan*, 11(2), 67.
- Erik, M., Hariyanto, B., & Liyanthy, M. (2008). Eksploritasi Antarmuka Grafis Pemakaian *Tkinter* Pada Lingkungan Bahasa Python. *Institutional Repositories & Scientific Journals, Thesis*.
- Fang, M., Yue, G., & Yu, Q. (2009). *The Study on an Application of Otsu Method in Canny Operator*. *Int. Symposium. Inf*, vol. 2, no. 2, 109-112.
- Hasoloan, J. (2010). Pengantar Ilmu Ekonomi. Deepublish.
- Husni, H. S., Arifin, F., & Yuliyanti. (2013). Logika Fuzzy untuk Audit Sistem Informasi. *ComTech*, 4(1), 68-75.
- Irianto, K. D. (2010). Pendekripsi Gerak Berbasis Kamera Menggunakan OpenCV Pada Ruangan. *J. Ilm. KomuniTi*, 2(1).
- Kadir, A. (2005). *Dasar Pemrograman Python*. Yogyakarta: Andi Offset.

- Kaur, E. K., Mutenja, V., & Gill, I. S. (2010). *Fuzzy Logic Based Image Edge Detection Algorithm in MATLAB*. *International Journal of Computer Application*, 1(22), 55-58.
- Khaire, P. A., & Thakur, N. (2012). *A Fuzzy Set Approach for Edge Detection*. *International Journal of Image Processing (IJIP)*, Vol. 6, Issue 6, 406-412.
- Kurniawan, H., Setiyono, B., & Isnanto, R. R. (2011). Aplikasi Penjawab Pesan Singkat Automatis dengan Bahasa Python. *Doctoral dissertation, Jurusan Teknik Elektro Fakultas Teknik Undip*.
- Kusumadewa, C. C., & Supatman. (2018). Identifikasi Citra Daun Teh Menggunakan Metode Histogram untuk Deteksi Dini Serangan Awal Hama Empoasca. *Jurnal Multimedia & Artificial Intelligence*, 2(1), 27-36.
- Laucereno, S. F. (2020). Peredaran Uang Palsu di RI Turun, Ini Datanya. Jakarta: *detikFinance*.
- Lutz, M. (2010). *Programming Python*. Fourth Edition ed. Sebastopo: O'Reilly Media, Inc.
- Mamta, J., & P. Singh, S. (2009). *Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain*. *International Journal of Computer Theory and Engineering*, 1(5).
- Miladiah, Umar, R., & Riadi, I. (2019). Implementasi Local Binary Pattern untuk Deteksi Keaslian Mata Uang Rupiah. *Jurnal Edukasi dan Penelitian Informatika*, 197-201.
- Mistry, K., & Saluja, A. (2016). *An introduction to OpenCV using Python with Ubuntu*. Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., 1(2), 65-68.
- Muwardi, F., & Fadlil, A. (2018). Sistem Pengenalan Bunga Berbasis Pengolahan Citra dan Pengklasifikasi Jarak. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, 3(2), 124.
- Nachtegael, M., Van Der Weken, D., Van De Ville, E. E., & Kerre. (2003). *Fuzzy Filters for Image Processing*. Springer, 178-194.
- Nafi'iyah, N. (2015). Algoritma Kohonen dalam Mengubah Citra Graylevel Menjadi Citra Biner. *Jurnal Ilmiah Teknologi Informasi Asia*, 9(2), 49-55.
- Nasution, H. (2012). Implementasi Logika Fuzzy pada Sistem Kecerdasan Buatan. *Jurnal ELKHA*, 4(2), 4-8.

- Pambudi, A. R., Garno, & Purwantoro. (2020). Deteksi Keaslian Uang Kertas Berdasarkan *Watermark* dengan Pengolahan Citra Digital. *JIP (Jurnal Informatika Polinema)*, 69-74.
- Prasetyo, E. (2011). *Pengolahan Citra Digital dan Aplikasinya menggunakan Matlab*. Yogyakarta: Penerbit ANDI.
- Putra, D. (2010). *Pengolahan Citra Digital*. Andi: Yogyakarta.
- Putra, D., Kushartantya, & Sugiharto, A. (2017). Pendekripsi Tepi Citra Digital dengan Logika Fuzzy. *Jurnal Masyarakat Informatika*, 4(7), 11-21.
- R., F. G. (2013). Pengolahan Citra Digital untuk Menghitung Luas Daerah Bekas Penambangan Timah. *J. Nas. Tek. Elektro*, vol. 2, no. 2, 27-34.
- Reja, I. D., & Santoso, A. J. (2013). Pengenalan Motif Sarung (Utan Maumere) menggunakan Deteksi Tepi.
- Romindo, & Khairina, N. (2017). Analisa Perbandingan Metode *Edge Detection*. *Seminar Nasional Teknologi Informatika*, 244-251.
- Sani, K., Wahyu Winarno, W., & Fauziati, S. (2016). Analisis Perbandingan *Algoritma Classification* untuk *Authentication* Uang Kertas (Studi Kasus: *Banknote Authentication*). *Jurnal Informatika*, 10(1), 1130-1139.
- Simanjuntak, N. P. (2012). Aplikasi *Fuzzy Logic Controller* pada Pengontrolan Lampu Lalu Lintas. *Makalah IF4058 Topik Khusus Informatika I - Sem. II*.
- Sutikno, & Waspada, I. (2016). Perbandingan Metode Defuzzifikasi Sistem Kendali Logika Fuzzy Model Mamdani pada Motor DC. *Jurnal Masyarakat Informatika*, 2(3), 27-37.
- Thedean, H., & Sugiarto, M. (2008). Penerapan Fuzzy *If-Then Rules* untuk Peningkatan Kontras pada Citra Hasil Mammografi. *Jurnal Informatika*, 9(1), 1-7.
- Umar, R., Riadi, I., & Miladiah. (2018). Sistem Identifikasi Keaslian Uang Kertas Rupiah menggunakan Metode *K-Means Clustering*. *Techno.COM*, 17(2), 179-185.
- Utama, J. (2011). Akuisisi Citra Digital Menggunakan Pemrograman Matlab. *Majalah Ilmiah UNIKOM*, 72.
- Wibowo, R. E., Isnanto, R. R., & Zahra, A. A. (2014). Perbandingan Kinerja Operator Sobel dan *Laplacian Of Gaussian* (LOG) terhadap Acuan Canny

- untuk Mendeteksi Tepi Citra. *TRANSIENT: Jurnal Ilmiah Teknik Elektro*, 3(1), 51-56.
- Wicaksono, D. (2008). Perangkat Lunak Identifikasi Nilai Nominal dan Keaslian Uang Kertas Rupiah menggunakan Jaringan Syaraf Tiruan Backpropagation. Depok: Jurusan Teknik Elektro Fakultas Teknik Universitas Indonesia.
- Yulianto, S., Nataliani, & Kurniawan, A. (2009). Penerapan Logika Fuzzy pada Sistem Deteksi Tepi Aplikasi *Computer Assistant Diagnosis* Kanker Payudara. *Jurnal Teknologi Informasi*, 6(1), 1-10.
- Zalukhu, A. (2016). Implementasi Metode Canny dan Sobel untuk Mendeteksi Tepi Citra. *JURIKOM (Jurnal Riset Komputer)*, 3(6).

LAMPIRAN

```
# Import Packages
import numpy as np
from tkinter import *
from tkinter import filedialog
import cv2 as cv
from PIL import ImageTk, Image

class gui:
    """
        Tampilan GUI untuk program
    """

    def __init__(self):
        """
            Fungsi constructor ketika class dipanggil
        """

        # Inisiasi window
        self.window = Tk()
        self.window.title('Deteksi Watermark')
        w, h = 750, 700
        self.window.minsize(width=w, height=h)
        self.window.maxsize(width=w+200, height=h)

        # Inisiasi frame
        frame0 = Frame(self.window, pady=20, padx=3, height=80)
        frame0.grid(row=0, sticky='ew')
        frame1 = Frame(self.window, pady=3, padx=3)
        frame1.grid(row=1, sticky='ew')

        # Label Text Frame
        text_label = Label(frame0, text='Deteksi Watermark', pady=40)
        text_label.config(font=('Arial', 20))
        text_label.place(relx=.5, rely=.5, anchor="center")

        # Button Upload
        upload_button = Button(frame1, text='Input Citra',
                               command=self.upload_image)
        upload_button.grid(row=5, column=0, sticky='nw')
```

```
# Process Button
process_button = Button(frame1, text='Proses',
                        command=self.processing_image)
process_button.grid(row=5, column=0, sticky='n', padx=(100, 0))

# Show Uploaded Image Label
self.img_list = []
zero_img = PhotoImage()

self.upload_image = Label(
    frame1, image=zero_img, width=200, height=200, borderwidth=
2, relief='sunken')
self.upload_image.grid(row=0, column=0)

self.uploaded_image_label = Label(frame1, text='Gambar Awal')
self.uploaded_image_label.grid(
    row=1, column=0, pady=(0, 20), sticky='n')

# Processed Image
# Blurred Frame
self.blurred_label = Label(
    frame1, image=zero_img, width=200, height=200, borderwidth=
2, relief='sunken')
self.blurred_label.grid(row=0, column=1, padx=(30, 0))

self.blurred_image_label = Label(frame1, text='Gaussian Blur')
self.blurred_image_label.grid(
    row=1, column=1, pady=(0, 20), sticky='n')

# Grayscale Frame
self.grayscaled_label = Label(
    frame1, image=zero_img, width=200, height=200, borderwidth=
2, relief='sunken')
self.grayscaled_label.grid(row=0, column=2, padx=(30, 0))

self.grayscaled_image_label = Label(frame1, text='Grayscale')
self.grayscaled_image_label.grid(
    row=1, column=2, pady=(0, 20), sticky='n')

# Binarized Frame
self.binarized_label = Label(
    frame1, image=zero_img, width=200, height=200, borderwidth=
2, relief='sunken')
self.binarized_label.grid(row=0, column=3, padx=(30, 0))
```

```
self.binarized_image_label = Label(frame1, text='Binarized')
self.binarized_image_label.grid(
    row=1, column=3, pady=(0, 20), sticky='n')

# Canny Frame
self.canny_label = Label(
    frame1, image=zero_img, width=200, height=200, borderwidth=
2, relief='sunken')
self.canny_label.grid(row=2, column=1, padx=(30, 0))

self.canny_image_label = Label(frame1, text='Canny')
self.canny_image_label.grid(row=3, column=1, pady=(0, 20), stic
ky='n')

# Fuzzy Frame
self.contoured_label = Label(
    frame1, image=zero_img, width=200, height=200, borderwidth=
2, relief='sunken')
self.contoured_label.grid(row=2, column=2, padx=(30, 0))

self.contoured_image_label = Label(frame1, text='Fuzzy')
self.contoured_image_label.grid(
    row=3, column=2, pady=(0, 20), sticky='n')

# Hasil pemrosesan - Canny
self.hasil_name_canny = Label(frame1, text='Hasil Pemrosesan Ca
nny')
self.hasil_name_canny.grid(row=5, column=1, padx=(2, 0))

self.hasil_label_canny = Label(frame1, text='-
', width=20, bg='white')
self.hasil_label_canny.grid(row=6, column=1, padx=(0, 0))

# Hasil pemrosesan - Fuzzy
self.hasil_name = Label(frame1, text='Hasil Pemrosesan Fuzzy')
self.hasil_name.grid(row=5, column=2, padx=(0, 0), stick='n')

self.hasil_label = Label(frame1, text='-
', width=20, bg='white')
self.hasil_label.grid(row=6, column=2, padx=(0, 0), stick='n')

self.window.mainloop()
```

```
@staticmethod
def upload():
    ...
    Upload button

    ...
    imgname = filedialog.askopenfilename(title='Choose image')
    return imgname

def upload_image(self):
    ...
    Upload image yang diinginkan

    ...
    imgname = self.upload()
    img = cv.imread(imgname, cv.IMREAD_UNCHANGED)

    # Resize
    resized_img = cv.resize(img, (200, 200))

    # Rearrange cv to PIL img array
    b, g, r = cv.split(resized_img)
    arranged_img = cv.merge((r, g, b))

    # cv to PIL
    pil_img = Image.fromarray(arranged_img)

    imgs = ImageTk.PhotoImage(pil_img)
    self.upload_image.configure(image=imgs)
    self.upload_image.image = imgs

    # Append to Array
    self.img_list.append(resized_img)

def processing_image(self):

    # Gaussian Blur to reduce noise
    blur_arr = cv.GaussianBlur(self.img_list[-1], (9, 9), 0)
    b, g, r = cv.split(blur_arr)
    arranged.blur = cv.merge((r, g, b))
    blur_img = Image.fromarray(arranged.blur)
    blur_pil = ImageTk.PhotoImage(image=blur_img)
    self.blurred_label.configure(image=blur_pil)
    self.blurred_label.image = blur_pil
    self.blurred_image_label.configure(text='Gaussian Blur')
```

```

# Grayscaled image
gray_arr = cv.cvtColor(blur_arr, cv.COLOR_BGR2GRAY)
gray_img = Image.fromarray(gray_arr)
gray_pil = ImageTk.PhotoImage(gray_img)
self.grayscaled_label.configure(image=gray_pil)
self.grayscaled_label.image = gray_pil
self.grayscaled_image_label.configure(text='Grayscale')

# Binarized image
ret, thresh_arr = cv.threshold(
    gray_arr, 200, 255, cv.THRESH_BINARY+cv.THRESH_OTSU)
thresh_img = Image.fromarray(thresh_arr)
thresh_pil = ImageTk.PhotoImage(image=thresh_img)
self.binarized_label.configure(image=thresh_pil)
self.binarized_label.image = thresh_pil
self.binarized_image_label.configure(text='Binarized')

# Canny image form
canny_arr = cv.Canny(thresh_arr, 200, 300)
contours, hierarchy = cv.findContours(
    canny_arr, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)
print(len(contours))
cannyContour = len(contours)
canny_img = Image.fromarray(canny_arr)
canny_pil = ImageTk.PhotoImage(image=canny_img)
self.canny_label.configure(image=canny_pil)
self.canny_label.image = canny_pil
self.canny_image_label.configure(text='Canny')

# Fuzzy image form
blur = cv.GaussianBlur(gray_arr, (5, 5), 0)
ret, otsu_threshold = cv.threshold(
    gray_arr, 0, 255, cv.THRESH_BINARY+cv.THRESH_OTSU)
contours, hierarchy = cv.findContours(
    otsu_threshold, cv.RETR_TREE, cv.CHAIN_APPROX_SIMPLE)
print(len(contours))
fuzzyContour = len(contours)
ret, cont_arr = cv.threshold(
    thresh_arr, 255, 0, cv.THRESH_BINARY)
cont = cv.drawContours(cont_arr, contours, -
1, (255, 255, 255), 1)
contoured_img = Image.fromarray(cont)
contoured_pil = ImageTk.PhotoImage(image=contoured_img)
self.contoured_label.configure(image=contoured_pil)
self.contoured_label.image = contoured_pil
self.contoured_image_label.configure(text='Fuzzy')

```

```
# Calculation to meet condition --> There is a watermark or not

# Canny
if cannyContour <= 45 or cannyContour >= 125:
    text1 = 'Tanda Air tidak Terdeteksi!'
else:
    text1 = 'Tanda Air Terdeteksi!'

# Fuzzy
if fuzzyContour <= 17 or fuzzyContour >= 45:
    text = 'Tanda Air tidak Terdeteksi!'
else:
    text = 'Tanda Air Terdeteksi!'

# Canny
self.hasil_label_canny.configure(text=text1)
self.hasil_label.text = text

# Fuzzy
self.hasil_label.configure(text=text)
self.hasil_label.text = text

if __name__ == '__main__':
    gui()
```