

DAFTAR PUSTAKA

- Alfiah, F., Setiadi, A., Supriadi, A., Maulana, I., Informatika, T., & Raharja, S. (2017). Discrete Cosine Transform DCT Methods on Compression RGB and Grayscale image. *International Journal of Computer Techniques*, 4(6), 24–29. <http://www.ijctjournal.org>
- Basuki, A. (2005). *Metode Numerik dan Algoritma Komputasi*. ANDI.
- Cok, D. R. (1996). *Cloning Technique For Digital Image Retouching* (Patent No. 5,555,194). United States Patent.
- Febrianda, D. A., Andreswari, D., & Purwandari, E. P. (2018). *Sistem Autentifikasi Citra Digital Terintegrasi Dengan Error Level Analysis (Ela) Dan Color Filter Array(Cfa) Berbasis Web*. 4(March 2016), 45–56. <https://ejournal.unib.ac.id/index.php/rekursif/article/view/952>
- Gunawan, T. S., Hanafiah, S. A. M., Kartiwi, M., Ismail, N., Za'bah, N. F., & Nordin, A. N. (2017). *Development of photo forensics algorithm by detecting photoshop manipulation using error level analysis*. 7(1), 131–137. <https://doi.org/10.11591/ijeecs.v7.i1.pp131-137>
- Hussain, M., Saleh, S. Q., Aboalsamh, H., Muhammad, G., & Bebis, G. (2014). *Comparison between WLD and LBP descriptors for non-intrusive image forgery detection*. 197–204. <https://doi.org/10.1109/INISTA.2014.6873618>
- Krasmala, R., Budimansyah, A., & Lenggana, U. T. (2017). Kompresi Citra Dengan Menggabungkan Metode Discrete Cosine Transform (DCT) dan Algoritma Huffman. *Jurnal Online Informatika*, 2(1), 1. <https://doi.org/10.15575/join.v2i1.79>
- Krawetz, N. (2008). *A Picture 's Worth ... Digital Image Analysis and Forensics Version 2*. 1–43.
- Lundh, F. (1995). *Pillow : ImageEnchane Module*.

<https://pillow.readthedocs.io/en/stable/reference/ImageEnhance.html>

- Murpratiwi, S. I., & Widyantara, I. M. O. (2018). Pemilihan Algoritma Kompresi Optimal untuk Citra Digital Bitmap. *Majalah Ilmiah Teknologi Elektro*, 17(1), 94–101. <https://doi.org/10.24843/MITE.2018.v17i01.P13>
- Putra, D. (2010). *Pengolahan Citra Digital* (Weststringsih (ed.); ed. I). ANDI. <https://books.google.co.id/books?id=NectMutqXJAC&printsec=frontcover#v=onepage&q&f=false>
- Qazi, T., Hayat, K., Khan, S. U., Madani, S. A., Khan, I. A., Kołodziej, J., Li, H., Lin, W., Yow, K. C., & Xu, C. Z. (2013). *Survey on blind image forgery detection*. 7(7), 660–670. <https://doi.org/10.1049/iet-ipr.2012.0388>
- Redi, J. A., Taktak, W., & Dugelay, J. L. (2011). *Digital image forensics: A booklet for beginners* (Vol. 51, Nomor 1). <https://doi.org/10.1007/s11042-010-0620-1>
- Sadeghi, S., Jalab, H., & Dadkhah, S. (2012). *Efficient Copy-Move Forgery Detection for Digital Images*. 71(11), 542–546. <http://waset.org/publications/10220>
- Saifullah, S., -, S., & Yudhana, A. (2016). *Analisis Perbandingan Pengolahan Citra Asli Dan Hasil Cropping Untuk Identifikasi Telur*. 2(3), 341–350. <https://doi.org/10.28932/jutisi.v2i3.512>
- Sari, T., Riadi, I., & Fadlil, A. (2016). *Forensik Citra untuk Deteksi Rekayasa File Menggunakan Error Level Analysis*. 2(1), 133–138. <http://ars.ilkom.unsri.ac.id>
- Shivakumar, B. L., & Baboo, S. S. (2011). *Detecting Copy-Move Forgery in Digital Images: A Survey and Analysis of Current Methods*. 10(7), 61–65.
- Sutoyo, T., Mulyanto, E., Suhartono, V., & Nurhayati, O. (2009). *Teori Pengolahan Citra Digital*. ANDI.
- Zhao, X., Li, J., Li, S., & Wang, S. (2011). *Detecting Digital Image Splicing in*

























*Chroma Spaces*. 6526 LNCS, 12–22. [https://doi.org/10.1007/978-3-642-18405-5\\_2](https://doi.org/10.1007/978-3-642-18405-5_2)





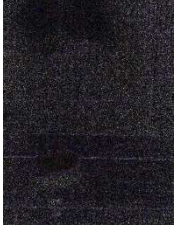



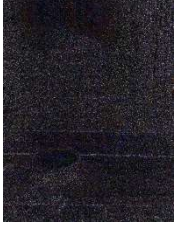







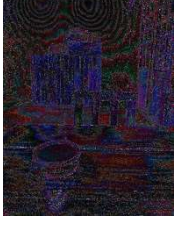


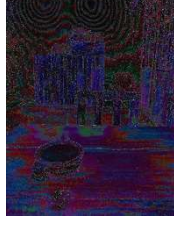
Zulfan, Arnia, F., & Muharar, R. (2016). *Deteksi Pemalsuan Citra dengan Teknik Copy-Move Menggunakan Metode Ordinal Measure dari Koefisien Discrete Cosine Transform*. 5(2), 165. <https://doi.org/10.25077/jnte.v5n2.230.2016>

## **LAMPIRAN**

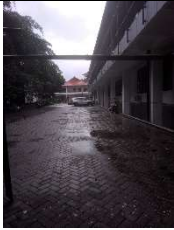






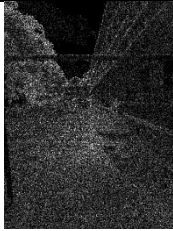
















Lampiran 1. Hasil Pembuatan Citra ELA





















A. Visualisasi Citra A

<i>Kualitas (%)</i>	<i>Asli</i>	<i>Splicing</i>	<i>Copy-move</i>	<i>Retouching</i>
				
95				
85				
75				
65				
55				

45				
35				
25				
15				
5				

## B. Visualisasi Citra B

<i>Kualitas (%)</i>	<i>Asli</i>	<i>Splicing</i>	<i>Copy-move</i>	<i>Retouching</i>
				
95				
85				
75				
65				
55				

45				
35				
25				
15				
5				



```
from tkinter import *
from PIL import ImageTk, Image, ImageChops, ImageEnhance
from tkinter import filedialog
import cv2
import numpy as np
from numpy import array, asarray
import sys

def open_img():
    ## Select the Imagename from a folder
    x = openfilename()
    label = Label(root, text=x)
    label.grid(row=1, column=2)
    resaved = x + '.resaved.jpg'
    ela = x + '.ela.jpg'
    ela_scale = x + '.scale.jpg'

    ## opens the image
    img = Image.open(x)

    ## create compressed image
    img.save(resaved, 'JPEG', quality=95)
    img2 = Image.open(resaved)

    ## calculate the difference & create ela image
    ela_im = ImageChops.difference(img, img2)
    ela_im.save(ela)

    ## normalize image brightness
    ycbcr = ela_im.convert('YCbCr')
    konver = array(ycbcr)
    xx, y, z = konver.shape
    maks, mins = -sys.maxsize, sys.maxsize
    for m in range(xx):
        for n in range(y):
```

(lanjutan)

```
        if(konver[m, n, 0] >= maks):
            maks = konver[m, n, 0]
        if(konver[m, n, 0] <= mins):
            mins = konver[m, n, 0]
factor = (255.0/maks)
print(factor)
ela_im = ImageEnhance.Brightness(ela_im).enhance(factor)
ela_im.save(ela_scale)

## reopen image for calculate modified area
img3 = Image.open(ela)
img4 = Image.open(ela_scale)

## resize the image and apply a high-
quality down sampling filter
img = img.resize((300, 400), Image.ANTIALIAS)
img2 = img2.resize((300, 400), Image.ANTIALIAS)
img3 = img3.resize((300, 400), Image.ANTIALIAS)
img4 = img4.resize((300, 400), Image.ANTIALIAS)

# PhotoImage class is used to add image to widgets, icons
etc
img = ImageTk.PhotoImage(img)
img2 = ImageTk.PhotoImage(img2)
img3 = ImageTk.PhotoImage(img3)
img4 = ImageTk.PhotoImage(img4)

# create a label
panel = Label(root, image=img)
panel2 = Label(root, image=img2)
panel3 = Label(root, image=img3)
panel4 = Label(root, image=img4)

# set the image as img
panel.image = img
```

(lanjutan)

```
panel.grid(row=2, column=1)
label1 = Label(root, text="Citra Uji").grid(row=3, column=1)
panel2.image = img2
panel2.grid(row=2, column=2)
label2 = Label(root, text="Resaved").grid(row=3, column=2)
panel3.image = img3
panel3.grid(row=2, column=3)
label3 = Label(root, text="ELA").grid(row=3, column=3)
panel4.image = img4
panel4.grid(row=2, column=4)
label4 = Label(root, text="Normalisasi").grid(row=3, column=4)

# calculate manipulation percentage
test = [ela]
persen = [0.0, 0.0]
for isi in range(len(test)):
    citra = cv2.imread(test[isi])
    im_arr = asarray(citra, dtype='float')
    x, y, z = im_arr.shape
    avg = 0
    blok = 0
    temp = 0

    for i in range(x):
        for j in range(y):
            avg += (im_arr[i, j, 0]+im_arr[i, j, 1]+im_arr[i, j, 2])
            blok += 1
        avg //= (x*y*z)

    for i in range(1, (x//8)+1):
        for j in range(1, (y//8)+1):
            avg_blok = 0
```

(lanjutan)

```
for n in range((i*8)-7, i*8):
    for m in range((j*8)-7, j*8):
        avg_blok += im_arr[n-1][m-1][0] + \
                    im_arr[n-1][m-1][1]+im_arr[n-
1][m-1][2]
        avg_blok /= 3*64
        if(avg_blok > avg):
            temp += 1
            blok += 1
        persen[isi] = ((temp/blok)*100)
        print(persen[isi])
    label_persen = Label(root, text="Persentase Manipulasi")
    label_persen.grid(row=4, column=1)
    label_persen1 = Label(root, text=(" %.2f" % (persen[0])))
)
    label_persen1.grid(row=4, column=3)

def openfilename():
    # open file dialog box to select image
    # The dialogue box has a title "Open"
    filename = filedialog.askopenfilename(title='ELA')
    return str(filename)

# Create a window
root = Tk()

# Set Title as Image Loader
root.title("Error Level Analysis")

# Set the resolution of window
root.geometry("1210x500")

# Allow Window to be resizable
root.resizable(width=True, height=True)
```

(lanjutan)

```
# Create a button and place it into the window using grid layout
```

```
btn = Button(root, text='open image', command=open_img).grid  
(row=1, column=1)
```

```
root.mainloop()
```