

# Laporan Praktikum Pengolahan Citra Digital

---

Nama: Akhlish Khairul Anam

NIM: 123456789

Program Studi: S1 Informatika

Mata Kuliah: Pengolahan Citra Digital

Tanggal: 13 Juni 2025

## Pendahuluan

Pengolahan citra digital adalah teknik memanipulasi gambar menggunakan komputer. Pada praktikum ini, dilakukan tiga jenis pemrosesan citra yaitu konversi citra ke biner, negatif, dan flipping. Tujuannya adalah memahami bagaimana setiap transformasi tersebut memengaruhi tampilan visual gambar.

## Tujuan Praktikum

- Mengubah citra ke grayscale dan biner dengan beberapa nilai ambang.
- Membuat citra negatif dari gambar RGB.
- Melakukan flipping citra secara vertikal, horizontal, dan titik asal.
- Menganalisis hasil pengolahan berdasarkan perbedaan parameter.

## Kode Program

### 1. Konversi Citra Biner

#### Konversi Citra Biner

```
def konversi_biner(image_path, thresholds, output_folder):  
    img = Image.open(image_path).convert('L') # grayscale  
  
    for t in thresholds:  
        biner = img.point(lambda p: 255 if p > t else 0)  
        output_file = f"{output_folder}/gambar_biner_{t}.jpg"  
        biner.save(output_file)  
        print(f"Disimpan: {output_file}")  
  
# Jalankan fungsi  
konversi_biner(image_path, [50, 128, 200, 230], "hasil")
```

### 2. Konversi Citra Negatif

#### Konversi Citra Negatif


```
def konversi_negatif(image_path, output_folder):  
    img = Image.open(image_path)  
    negatif = Image.eval(img, lambda p: 255 - p)  
    output_file = f"{output_folder}/gambar_negatif.jpg"  
    negatif.save(output_file)  
    print(f"Disimpan: {output_file}")  
  
konversi_negatif(image_path, "hasil")
```

Disimpan: hasil/gambar\_negatif.jpg

### 3. Flipping Citra

#### Flipping Citra

```
[4] def flipping(image_path, output_folder):  
    img = Image.open(image_path)  
  
    img.transpose(Image.FLIP_TOP_BOTTOM).save(f"{output_folder}/gambar_flip_vertikal.jpg")  
    img.transpose(Image.FLIP_LEFT_RIGHT).save(f"{output_folder}/gambar_flip_horizontal.jpg")  
    img.transpose(Image.ROTATE_180).save(f"{output_folder}/gambar_flip_titik_asal.jpg")  
  
    print("Gambar flipping disimpan.")  
  
    flipping(image_path, "hasil")
```

 Gambar flipping disimpan.

#### **4. Hasil Gambar**

gambar\_biner\_128.jpg

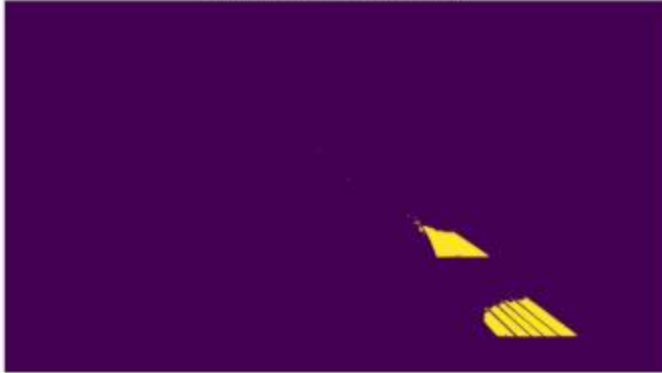


gambar\_biner\_200.jpg



gambar\_biner\_230.jpg

gambar\_biner\_255.jpg



gambar\_biner\_50.jpg



gambar\_flip\_horizontal.jpg





gambar\_flip\_titik\_asal.jpg



gambar\_flip\_vertikal.jpg



### **Analisis dan Kesimpulan**

- Nilai ambang rendah (misalnya 50) menghasilkan gambar biner yang lebih gelap, sedangkan nilai tinggi (230) lebih terang.
- Citra negatif membalik warna gambar, bagian terang menjadi gelap dan sebaliknya.
- Flipping vertikal membalik gambar atas ke bawah, horizontal membalik kiri ke kanan, dan titik asal membalik 180 derajat.

Dengan percobaan ini, diperoleh pemahaman tentang dampak setiap operasi terhadap struktur visual citra.

**antarmuka GUI sederhana**



## Setup Streamlit dan Pyngrok di Google Colab

```
[6] !pip install streamlit pyngrok --quiet
```



```
44.3/44.3 kB 1.7 MB/s eta 0:00:00
9.9/9.9 MB 35.9 MB/s eta 0:00:00
6.9/6.9 MB 57.2 MB/s eta 0:00:00
79.1/79.1 kB 5.7 MB/s eta 0:00:00
```

## Tambahkan Ngrok Authtoken

```
[10] !ngrok config add-authtoken 2yRogubJ6WcmH4gD6c7mvkZrcKu_24dQYzdXfPkbXUvXLwNA
```



```
Authtoken saved to configuration file: /root/.config/ngrok/ngrok.yml
```

## Simpan Aplikasi Streamlit GUI (dengan Citra) ke app.py

```
[11] code = '''
import streamlit as st
from PIL import Image
import numpy as np
import cv2

st.title("🖼️ Pengolahan citra: Binar, negatif, dan Flipping")

uploaded_file = st.file_uploader("upload gambar", type=["jpg", "png", "jpeg"])

if uploaded_file:
    image = Image.open(uploaded_file).convert('RGB')
    st.image(image, caption="Gambar Asli", use_column_width=True)

    # Convert to grayscale
    gray = np.array(image.convert('L'))

    st.subheader("🔍 Threshold Miner")
    threshold_manual = st.slider("Pilih threshold manual", 0, 255, 128)
    bin_manual = (gray > threshold_manual) * 255
    st.image(bin_manual.astype(np.uint8), caption=f"Miner Manual ({threshold_manual})", use_column_width=True)

    st.subheader("🧠 Otak's Thresholding")
    otak_thresh, bin_otak = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
    st.image(bin_otak, caption=f"Miner Otak ({otak_thresh:.2f})", use_column_width=True)

    st.subheader("🌑 Negatif")
    negatif = 255 - np.array(image)
    st.image(negatif, caption="Citra Negatif", use_column_width=True)

    st.subheader("🔄 Flipping")
    col1, col2, col3 = st.columns(3)
    with col1:
        flip_v = np.flipud(np.array(image))
        st.image(flip_v, caption="Vertikal")
    with col2:
        flip_h = np.fliplr(np.array(image))
        st.image(flip_h, caption="Horizontal")
    with col3:
        flip_180 = np.rot90(np.array(image), 2)
        st.image(flip_180, caption="Putik Asal")
    ...

with open("app.py", "w") as f:
    f.write(code)
```

## Jalankan Streamlit di Colab via ngrok

```
from pyngrok import ngrok

# Jalankan streamlit di background
!streamlit run app.py && cat /dev/null &

# Gunakan koneksi yang sesuai
public_url = ngrok.connect("http://localhost:8501")
print("Buka aplikasi GUI di sini 🖼️", public_url)

Buka aplikasi GUI di sini 🖼️ NgrokTunnel: "https://6757-34-186-72-142.ngrok-free.app" -> "http://localhost:8501"
```

LINK GOGGLE COLAB

<https://colab.research.google.com/drive/1q3nYagn6SsfTF2fsQAT0-Mw3XI792vni?usp=sharing>