



# UNIVERSITAS PELITA BANGSA

Berdasarkan Surat Keputusan Menristek DIKTI No : 664/KPT/I/2019 Tanggal 02 Agustus 2019

Jl. Inspeksi Kalimalang Tegal Danas Arah DELTAMAS, Cikarang Pusat - Kab. Bekasi  
Telp. (021) 2851 8181, 82, 83, 84, Fax. (021) 2851 8180

[www.pelitabangsa.ac.id](http://www.pelitabangsa.ac.id); [www.ecampus.pelitabangsa.ac.id](http://www.ecampus.pelitabangsa.ac.id); [www.pmb.pelitabangsa.ac.id](http://www.pmb.pelitabangsa.ac.id)

## SOAL UJIAN TENGAH SEMESTER (UTS)

|                |   |
|----------------|---|
| Mata Kuliah    | : Pengolahan Citra Digital  |
| Program Studi  | : Teknik Informatika  |
| Dosen Pengampu | : M. Fatchan, S.Kom, M.Kom  |
| Kelas          | : <b>TI.22.A.3, TI.22.A.1 4 TI.22.A.2 TI.22.B.2, TI.22.B.1, TI.22.A.4<br/>TI.22.A.5, TI.22.C.8, TI.22.C.6</b> |
| Waktu          | :   |
| Sifat Ujian    | : Mini Project  |
| Hari/Tanggal   | : -   |

### **Petunjuk Umum:**

1. Dahulukan menulis nama, NIM, dan kelas pada lembar jawaban
2. Semua jawaban dikerjakan di lembar jawaban yang telah disediakan
3. Baca dengan teliti setiap soal sebelum anda menjawabnya
4. Dilarang bekerja sama!

### **Soal:**

1. Kerjakan Project Web App untuk memanipulasi gambar citra menggunakan Streammlite dengan ketentuan mengubah
  - RGB menjadi HSV
  - Menghitung Histogram
  - Brignest dan Contrasts
  - Contour

Mengetahui

Kaprodi Teknik Informatika

Wahyu Hadikristanto, S.Kom. M.Kom

----- SELAMAT MENGERJAKAN -----

Nama : Sandi Bintara

NIM : 312010039

Kelas : TI20B1

Kerjakan Project Web App untuk memanipulasi gambar citra menggunakan Streammlite dengan ketentuan mengubah

## RGB menjadi HSV

```
import streamlit as st
from PIL import Image, ImageEnhance, ImageOps, ImageFilter
import numpy as np
import colorsys

# Fungsi untuk mengubah gambar dari RGB ke HSV
def rgb_to_hsv(image):
    # Ubah gambar ke array numpy
    img_array = np.array(image)

    # Ambil ukuran gambar
    height, width, _ = img_array.shape

    # Buat array kosong untuk menyimpan nilai HSV
    hsv_image = np.zeros((height, width, 3))

    # Iterasi melalui setiap piksel gambar dan ubah ke ruang warna HSV
    for y in range(height):
        for x in range(width):
            # Dapatkan nilai RGB dari piksel
            r, g, b = img_array[y, x]

            # Konversi nilai RGB ke HSV menggunakan colorsys
            h, s, v = colorsys.rgb_to_hsv(r / 255.0, g / 255.0, b / 255.0)

            # Simpan nilai HSV ke dalam array
            hsv_image[y, x] = [h, s, v]

    return hsv_image

# Tampilkan judul aplikasi
st.title('Konversi Gambar dari RGB ke HSV')
```

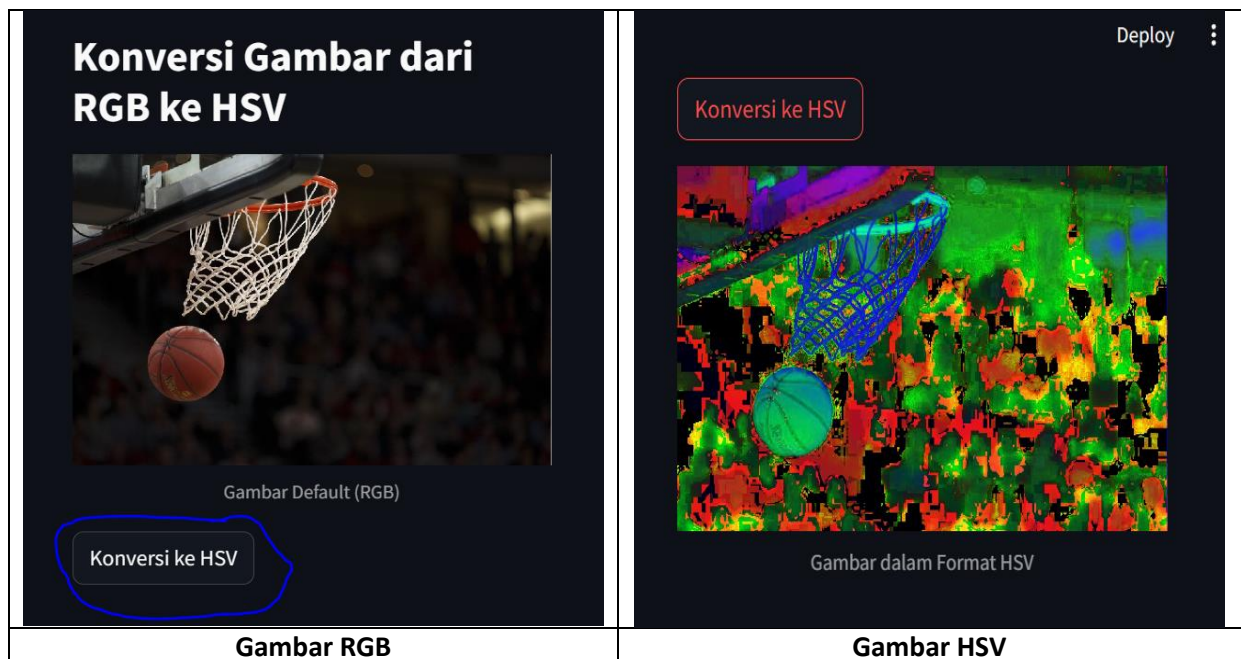
```
# Pilih gambar default
default_image = Image.open("C:/Users/sabin/Downloads/UTS
CITRA/foto/basketball.jpg")

# Tampilkan gambar default
st.image(default_image, caption='Gambar Default (RGB)', use_column_width=True)

# Tombol untuk konversi
if st.button('Konversi ke HSV'):
    # Konversi gambar default ke HSV
    hsv_image = rgb_to_hsv(default_image)

    # Tampilkan gambar yang telah dikonversi
    st.image(hsv_image, caption='Gambar dalam Format HSV',
use_column_width=True)
```

Hasil



## Menghitung Histogram

```
# Fungsi untuk menghitung histogram gambar
def calculate_histogram(image):
    # Konversi gambar ke array numpy dan ubah menjadi grayscale jika perlu
    img_array = np.array(image)
    if len(img_array.shape) == 3:
        img_array = np.mean(img_array, axis=2).astype(np.uint8)

    # Hitung histogram menggunakan numpy
    histogram, bins = np.histogram(img_array.flatten(), bins=256,
range=[0,256])

    return histogram



# Tampilkan judul aplikasi
st.title('Hitung Histogram Gambar Default')

# Pilih gambar default
default_image = Image.open("C:/Users/sabin/Downloads/UTS
CITRA/foto/basketball.jpg")

# Tampilkan gambar default
st.image(default_image, caption='Gambar Default', use_column_width=True)

# Tombol untuk hitung histogram
if st.button('Hitung Histogram'):
    # Hitung histogram gambar
    histogram = calculate_histogram(default_image)

    # Tampilkan histogram
    st.write("Histogram:", histogram)
```

|  |   |
|--|---|
|  <p>The screenshot shows a dark-themed application window titled 'Hitung Histogram Gambar Default'. It features a basketball image with a caption 'Gambar Default' and a red 'Hitung Histogram' button at the bottom.</p> |  <p>The screenshot shows the application window after clicking the 'Hitung Histogram' button. The title bar now says 'Hitung Histogram'. Below the title, it says 'Histogram:' followed by a list of values: 3,047, 15,783, 19,275, 18,466, 16,432, 16,365, 16,346, 17,148, 17,865, and 18,087.</p> |
| Gambar   | Hasil perhitungan histogramnya  |

## Brignest dan Contras

```
# Fungsi untuk mengatur kecerahan dan kontras gambar
def adjust_brightness_contrast(image, brightness, contrast):
    # Konversi gambar ke mode 'RGB' jika tidak dalam mode tersebut
    if image.mode != 'RGB':
        image = image.convert('RGB')

    # Buat objek ImageEnhance untuk mengatur kecerahan dan kontras
    enhancer = ImageEnhance.Brightness(image)
    image = enhancer.enhance(brightness)

    enhancer = ImageEnhance.Contrast(image)
    image = enhancer.enhance(contrast)

    return image

# Tampilkan judul aplikasi
st.title('Atur Kecerahan dan Kontras Gambar Default')

# Pilih gambar default
default_image = Image.open("C:/Users/sabin/Downloads/UTS
CITRA/foto/basketball.jpg")

# Tampilkan gambar default
st.image(default_image, caption='Gambar Default', use_column_width=True)

# Tambahkan slider untuk kecerahan dan kontras
brightness = st.slider("Kecerahan", 0.0, 2.0, 1.0, 0.1)
contrast = st.slider("Kontras", 0.0, 2.0, 1.0, 0.1)

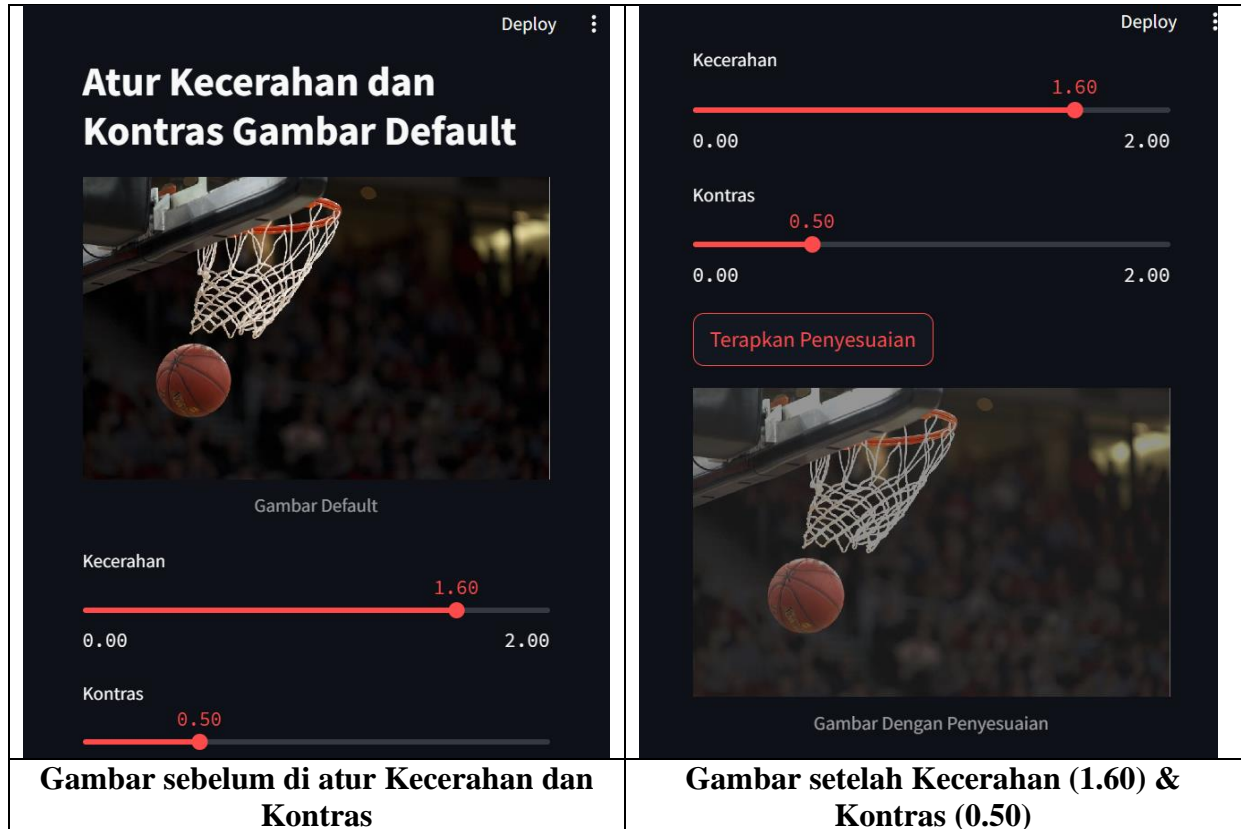
# Tombol untuk menerapkan penyesuaian
if st.button('Terapkan Penyesuaian'):
    # Terapkan penyesuaian ke gambar default
    adjusted_image = adjust_brightness_contrast(default_image, brightness,
    contrast)

    # Tampilkan gambar yang telah disesuaikan
    st.image(adjusted_image, caption='Gambar Dengan Penyesuaian',
    use_column_width=True)

# Fungsi untuk mengatur kontras gambar
def adjust_contrast(image, contrast):
    # Konversi gambar ke mode 'RGB' jika tidak dalam mode tersebut
    if image.mode != 'RGB':
        image = image.convert('RGB')
```

```
# Buat objek ImageEnhance untuk mengatur kontras
enhancer = ImageEnhance.Contrast(image)
image = enhancer.enhance(contrast)

return image
```



## Contour

```
# Fungsi untuk mengubah kontur gambar
def apply_contour(image, threshold):
    # Konversi gambar ke grayscale
    gray_image = ImageOps.grayscale(image)

    # Terapkan filter threshold
    binary_image = gray_image.point(lambda p: p > threshold and 255)

    # Dapatkan kontur menggunakan filter emboss
    contour_image = binary_image.filter(ImageFilter.FIND_EDGES)

    return contour_image

# Tampilkan judul aplikasi
st.title('Ubah Kontur Gambar Default')

# Pilih gambar default
default_image = Image.open("C:/Users/sabin/Downloads/UTS
CITRA/foto/basketball.jpg")

# Tampilkan gambar default
st.image(default_image, caption='Gambar Default', use_column_width=True)

# Tambahkan slider untuk threshold
threshold = st.slider("Threshold", 0, 255, 100)

# Tombol untuk menerapkan perubahan kontur
if st.button('Terapkan Perubahan Kontur'):
    # Terapkan perubahan kontur ke gambar default
    contoured_image = apply_contour(default_image, threshold)

    # Tampilkan gambar yang telah diubah kontur
    st.image(contoured_image, caption='Gambar Dengan Kontur',
use_column_width=True)
```

## Ubah Kontur Gambar Default



Gambar Default

Threshold

0 222 255

Terapkan Perubahan Kontur

Threshold

0 222 255

Terapkan Perubahan Kontur



Gambar Dengan Kontur

Gambar Sebelum di kenakan Kontur

Gambar sesudah di kenakan kontur sebesar (222)