PENGOLAHAN CITRA DIGITAL

"Disusun untuk Memenuhi Tugas Penglohan Citra Digital"

Dosen Pengampu : Leni Fitriani, ST. M.Kom.

Disusun Oleh :

Zaenal Syamsyul Arief

2206073

TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI GARUT

2024

DAFTAR ISI

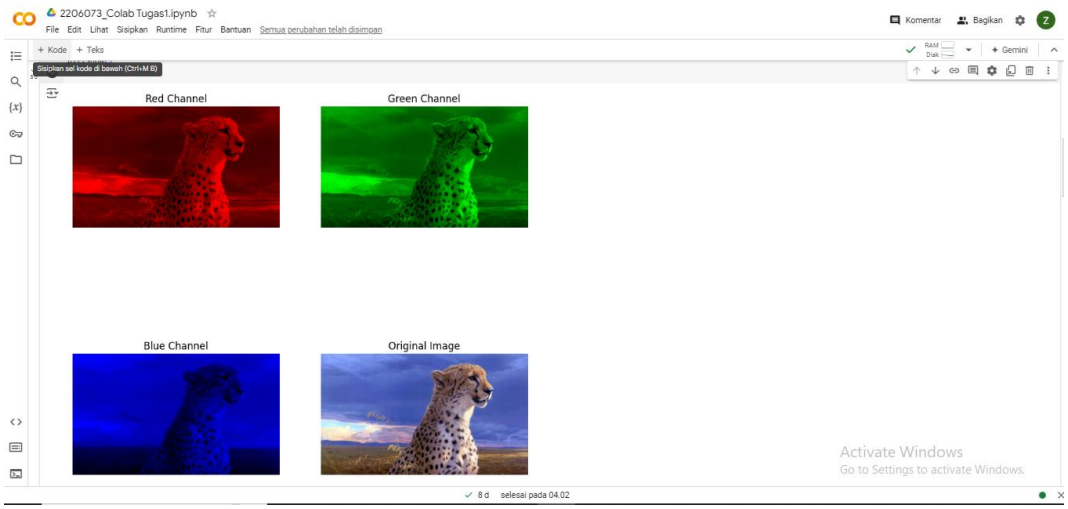
PENGOLAHAN CITRA

- Code Program Pertama

```
#dekomposisi R, G, B dari citra berwarna
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Step 1: Load an image (Replace the path with the actual
image file path)
image_path = '/content/cheatah.jpg'  # Ganti dengan path
gambar Anda
image = cv2.imread(image_path)

# Step 2: Convert the image from BGR (OpenCV default) to RGB
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Step 3: Split the image into its Red, Green, and Blue
channels
R, G, B = cv2.split(image_rgb)

# Step 4: Create a zero matrix to help in visualization
zeros = np.zeros(image.shape[:2], dtype="uint8")

# Step 5: Visualize the R, G, B channels separately
plt.figure(figsize=(10, 10))

# Red channel
plt.subplot(2, 2, 1)
red_image = cv2.merge([R, zeros, zeros])  # R channel with G,
B as 0
plt.imshow(red_image)
plt.title('Red Channel')
plt.axis('off')

# Green channel
plt.subplot(2, 2, 2)
green_image = cv2.merge([zeros, G, zeros])  # G channel with
R, B as 0
plt.imshow(green_image)
plt.title('Green Channel')
plt.axis('off')

# Blue channel
plt.subplot(2, 2, 3)
blue_image = cv2.merge([zeros, zeros, B])  # B channel with
R, G as 0
```

```
plt.imshow(blue_image)
plt.title('Blue Channel')
plt.axis('off')

# Original image for comparison
plt.subplot(2, 2, 4)
plt.imshow(image_rgb)
plt.title('Original Image')
plt.axis('off')

plt.show()
```

output :



➢ Analisis Program Pertama

Pada program pertama, dilakukan langkah-langkah sebagai berikut:

1. **Load Gambar**: Program memuat sebuah gambar berwarna menggunakan OpenCV dan mengubahnya dari format BGR (format default OpenCV) menjadi format RGB agar sesuai dengan standar warna umum dalam visualisasi.
2. **Dekomposisi Saluran Warna**: Gambar dipecah menjadi tiga saluran (Red, Green, Blue) dengan memisahkan komponen R, G, dan B menggunakan fungsi cv2.split(). Ini dilakukan untuk melihat setiap komponen warna secara terpisah.
3. **Pembuatan Citra Zero Matrix**: Matriks nol (zeros) digunakan untuk membantu dalam visualisasi saluran yang terisolasi. Hal ini memastikan

bahwa ketika kita ingin menampilkan satu saluran (misalnya Red), maka komponen Green dan Blue diatur menjadi nol.

4. **Visualisasi Saluran R, G, B**: Setiap saluran warna divisualisasikan dengan membuat citra terpisah untuk Red, Green, dan Blue di mana dua saluran lainnya diatur menjadi nol. Program juga menampilkan gambar asli untuk perbandingan.

Analisis Output:

- Output dari program ini adalah tiga gambar yang menunjukkan distribusi intensitas warna Red, Green, dan Blue pada gambar asli. Gambar ini berguna untuk memahami bagaimana masing-masing komponen warna membentuk keseluruhan citra.
- **Red Channel** (Kanal Merah):
    - Pada citra di bagian ini, hanya komponen merah yang ditampilkan, sementara nilai komponen hijau dan biru diatur menjadi nol (hitam). Oleh karena itu, area yang memiliki intensitas tinggi di saluran merah (seperti tubuh cheetah dan langit) tampak terang, sedangkan area dengan sedikit komponen merah akan tampak gelap.
- **Green Channel** (Kanal Hijau):
    - Di sini hanya komponen hijau yang ditampilkan, dengan komponen merah dan biru diatur menjadi nol. Area dengan intensitas tinggi dalam kanal hijau, seperti bagian rumput di bawah cheetah dan tubuh hewan itu sendiri, akan tampak terang, sedangkan area lain seperti langit yang memiliki sedikit intensitas hijau akan tampak lebih gelap.
- **Blue Channel** (Kanal Biru):
    - Citra ini hanya menunjukkan komponen biru, dengan komponen merah dan hijau diatur menjadi nol. Area yang memiliki intensitas biru tinggi, seperti langit di latar belakang, tampak lebih terang, sedangkan objek lain yang memiliki lebih sedikit komponen biru, seperti tubuh cheetah dan tanah, tampak lebih gelap.

- **Original Image** (Citra Asli):
    - Citra ini menampilkan gambar asli dengan semua komponen warna (merah, hijau, dan biru) digabungkan kembali. Ini memberikan perbandingan langsung dengan hasil dekomposisi saluran warna sebelumnya.

Setiap saluran warna (R, G, B) menyumbang bagian tertentu dari warna gambar asli. Misalnya, saluran biru mendominasi warna langit, saluran hijau terlihat pada rumput, dan saluran merah berpengaruh pada keseluruhan citra.Dengan memisahkan saluran warna ini, kita dapat memahami bagaimana masing-masing komponen membentuk warna akhir dalam citra penuh.

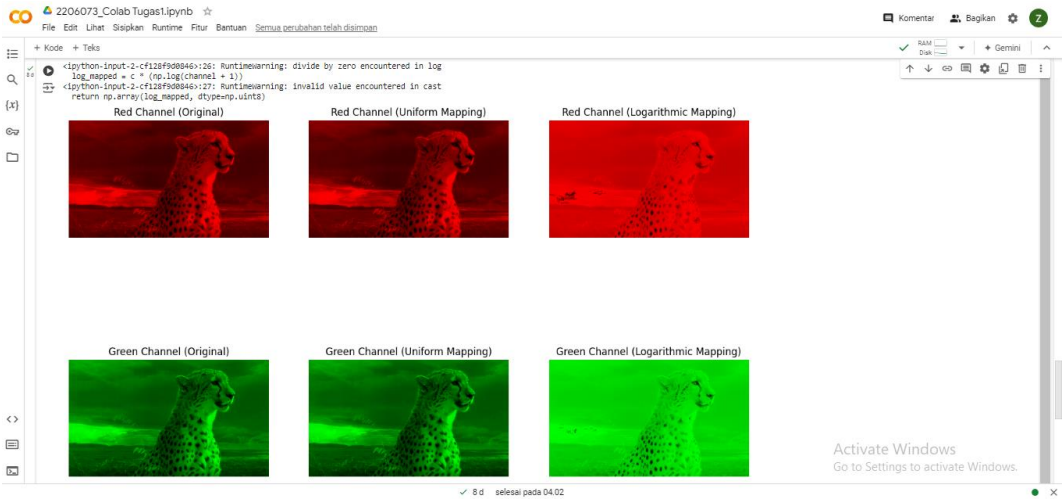
➢ Code Program Kedua

```
#Uniform Mapping dan Logarithmic Mapping
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Step 1: Load an image (Replace the path with the actual
image file path)
image_path = '/content/cheatah.jpg'  # Ganti dengan path
gambar Anda
image = cv2.imread(image_path)

# Step 2: Convert the image from BGR (OpenCV default) to RGB
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Step 3: Split the image into its Red, Green, and Blue
channels
R, G, B = cv2.split(image_rgb)

# Step 4: Create a zero matrix to help in visualization
zeros = np.zeros(image.shape[:2], dtype="uint8")

# Uniform Mapping (Normalize intensity between 0 and 255)
def uniform_mapping(channel):
    return cv2.normalize(channel, None, 0, 255,
cv2.NORM_MINMAX)

# Logarithmic Mapping
def logarithmic_mapping(channel):
    c = 255 / np.log(1 + np.max(channel))
    log_mapped = c * (np.log(channel + 1))
    return np.array(log_mapped, dtype=np.uint8)
```

```
# Step 5: Apply Uniform and Logarithmic Mapping to each
channel
R_uniform = uniform_mapping(R)
G_uniform = uniform_mapping(G)
B_uniform = uniform_mapping(B)

R_log = logarithmic_mapping(R)
G_log = logarithmic_mapping(G)
B_log = logarithmic_mapping(B)

# Step 6: Visualize the R, G, B channels separately and with
Uniform/Logarithmic mappings
plt.figure(figsize=(15, 15))

# Original Red channel
plt.subplot(3, 3, 1)
red_image = cv2.merge([R, zeros, zeros])  # R channel with G,
B as 0
plt.imshow(red_image)
plt.title('Red Channel (Original)')
plt.axis('off')

# Uniform Red
plt.subplot(3, 3, 2)
uniform_red_image = cv2.merge([R_uniform, zeros, zeros])
plt.imshow(uniform_red_image)
plt.title('Red Channel (Uniform Mapping)')
plt.axis('off')

# Logarithmic Red
plt.subplot(3, 3, 3)
log_red_image = cv2.merge([R_log, zeros, zeros])
plt.imshow(log_red_image)
plt.title('Red Channel (Logarithmic Mapping)')
plt.axis('off')

# Original Green channel
plt.subplot(3, 3, 4)
green_image = cv2.merge([zeros, G, zeros])  # G channel with
R, B as 0
plt.imshow(green_image)
plt.title('Green Channel (Original)')
plt.axis('off')

# Uniform Green
plt.subplot(3, 3, 5)
```

```
uniform_green_image = cv2.merge([zeros, G_uniform, zeros])
plt.imshow(uniform_green_image)
plt.title('Green Channel (Uniform Mapping)')
plt.axis('off')

# Logarithmic Green
plt.subplot(3, 3, 6)
log_green_image = cv2.merge([zeros, G_log, zeros])
plt.imshow(log_green_image)
plt.title('Green Channel (Logarithmic Mapping)')
plt.axis('off')

# Original Blue channel
plt.subplot(3, 3, 7)
blue_image = cv2.merge([zeros, zeros, B])  # B channel with
R, G as 0
plt.imshow(blue_image)
plt.title('Blue Channel (Original)')
plt.axis('off')

# Uniform Blue
plt.subplot(3, 3, 8)
uniform_blue_image = cv2.merge([zeros, zeros, B_uniform])
plt.imshow(uniform_blue_image)
plt.title('Blue Channel (Uniform Mapping)')
plt.axis('off')

# Logarithmic Blue
plt.subplot(3, 3, 9)
log_blue_image = cv2.merge([zeros, zeros, B_log])
plt.imshow(log_blue_image)
plt.title('Blue Channel (Logarithmic Mapping)')
plt.axis('off')

plt.show()
```

Output :



➢ Analisis Program Kedua:

Pada program kedua, dilakukan pemetaan intensitas warna dengan dua teknik berbeda, yaitu **Uniform Mapping** dan **Logarithmic Mapping**, setelah dekomposisi R, G, B dari gambar.

1. **Uniform Mapping**:
    - Teknik ini melakukan normalisasi nilai intensitas piksel di setiap saluran dari rentang nilai piksel yang ada (misalnya 0-255) agar menjadi lebih tersebar merata di antara nilai minimum dan maksimum. Ini memungkinkan distribusi intensitas lebih rata di seluruh spektrum.
2. **Logarithmic Mapping**:
    - Teknik ini menerapkan transformasi logaritmik pada intensitas warna. Hal ini digunakan untuk meningkatkan detail pada daerah dengan intensitas rendah karena sifat fungsi logaritma yang memperbesar nilai rendah lebih dari nilai tinggi.
3. **Visualisasi Mapping**:
    - Hasil dari setiap pemetaan divisualisasikan untuk tiap saluran warna, baik untuk saluran Red, Green, dan Blue. Program membandingkan intensitas asli, hasil uniform mapping, dan hasil logarithmic mapping pada tiap saluran warna.

Analisis Output:

- **Uniform Mapping**: Hasil pemetaan ini memperlihatkan distribusi intensitas yang lebih merata, sehingga kontras gambar menjadi lebih tinggi. Ini sangat efektif dalam meningkatkan distribusi warna pada gambar yang cenderung memiliki rentang intensitas sempit.
- **Logarithmic Mapping**: Hasil transformasi logaritmik memperlihatkan penguatan detail pada area yang memiliki intensitas rendah. Transformasi ini berguna dalam gambar yang memiliki banyak bayangan atau detail kecil yang sulit dilihat pada pemetaan linear.
- **Red Channel (Original)**:

  Ini adalah tampilan dari saluran warna merah asli, tanpa modifikasi. Gambar ini menampilkan distribusi intensitas warna merah yang ada pada citra, mirip dengan hasil dari program pertama.
- **Red Channel (Uniform Mapping)**:

  Pada gambar ini, pemetaan intensitas **Uniform** telah diterapkan ke saluran merah. **Uniform Mapping** melakukan normalisasi intensitas warna sehingga distribusi intensitas menjadi lebih rata di seluruh gambar. Akibatnya, citra ini tampak lebih cerah dan memiliki rentang intensitas merah yang lebih merata.
- **Red Channel (Logarithmic Mapping)**:

  Di sini, pemetaan **Logarithmic** diterapkan pada saluran merah. Transformasi logaritmik ini lebih memperkuat intensitas pada piksel dengan nilai rendah, sehingga detail pada area yang lebih gelap menjadi lebih terlihat. Namun, gambar mungkin terlihat lebih datar di area yang memiliki intensitas tinggi karena sifat dari fungsi logaritmik.
- **Green Channel (Original)**:

  Ini adalah saluran hijau asli dari gambar. Sama seperti saluran merah, ini menampilkan distribusi intensitas warna hijau tanpa modifikasi.
- **Green Channel (Uniform Mapping)**:

  Pemetaan **Uniform** diterapkan pada saluran hijau. Distribusi intensitas hijau pada gambar ini menjadi lebih seragam. Area yang sebelumnya kurang terlihat mungkin tampak lebih cerah setelah proses normalisasi ini.

- **Green Channel (Logarithmic Mapping)**:
  Pada citra ini, pemetaan **Logarithmic** diterapkan pada saluran hijau. Seperti halnya dengan saluran merah, logaritma memperkuat detail di area dengan intensitas rendah, memungkinkan detail lebih terlihat di bagian gelap dari gambar.

Uniform Mapping membuat distribusi intensitas pada setiap saluran warna menjadi lebih rata, sehingga hasilnya sering kali terlihat lebih cerah dan kontras. Ini sangat berguna untuk meningkatkan keseluruhan tampilan gambar dengan memperluas rentang intensitas piksel.

Logarithmic Mapping, di sisi lain, sangat baik untuk meningkatkan detail pada area dengan intensitas rendah, tetapi bisa menyebabkan detail pada area dengan intensitas tinggi menjadi kurang menonjol. Hal ini terlihat pada perbedaan hasil untuk saluran merah dan hijau.

Kesimpulan

Kesimpulan dari analisis kedua program ini menunjukkan bahwa dekomposisi citra berwarna dan pemetaan intensitas merupakan teknik yang sangat penting dalam pengolahan citra digital. Pada program pertama, dekomposisi citra menjadi tiga saluran warna dasar (Red, Green, Blue) membantu kita memahami kontribusi setiap komponen warna terhadap keseluruhan citra. Visualisasi terpisah dari setiap saluran memungkinkan analisis mendalam tentang distribusi intensitas warna di dalam gambar, yang penting dalam berbagai aplikasi seperti segmentasi dan deteksi objek.

Sementara itu, pada program kedua, dua metode pemetaan intensitas diperkenalkan, yaitu Uniform Mapping dan Logarithmic Mapping. Uniform Mapping berfungsi meratakan distribusi intensitas di setiap saluran warna, meningkatkan kontras dan membuat gambar tampak lebih cerah, cocok untuk gambar dengan rentang intensitas sempit. Di sisi lain, Logarithmic Mapping memperkuat intensitas pada area gelap sehingga detail yang sebelumnya tersembunyi menjadi lebih terlihat, meskipun area terang mungkin tampak kurang menonjol. Secara keseluruhan, kedua program ini memberikan dasar yang kuat untuk analisis lebih lanjut dalam pengolahan citra, di mana teknik dekomposisi warna dan pemetaan intensitas memainkan peran kunci dalam meningkatkan kualitas visual dan informasi yang dapat diekstraksi dari citra digital.