

UTS

PENGOLAHAN CITRA



INTELLIGENT COMPUTING

NAMA : Muhamad Fauzi Akbar

NIM : 202331192

KELAS : F

DOSEN : Dr.Dra.Dwina Kuswardani,M.Kom

NO.PC : 10

ASISTEN : 1. Sasikirana Ramadhanty Setiawan Putri

2. Rizqy Amanda

3. Ridho Chaerullah

4. Sakura Amastasya Salsabila Setiyanto

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2024/2025

DAFTAR ISI

| | |
|--|-----------|
| DAFTAR ISI..... | 2 |
| BAB I PENDAHULUAN | 3 |
| 1.1 Rumusan Masalah..... | 3 |
| 1.2 Tujuan Masalah | 3 |
| 1.3 Manfaat Masalah | 3 |
| BAB II LANDASAN TEORI..... | 4 |
| 2.1 Pengolahan Citra Digital..... | 4 |
| 2.2 Model warna dalam pengolahan citra HSV dan RGB..... | 5 |
| 2.3 Histogram dalam citra | 5 |
| 2.4 Thresholding dalam citra..... | 7 |
| 2.5 Greyscale citra | 8 |
| BAB III HASIL | 10 |
| 3.1 SOAL 1..... | 10 |
| 3.2 SOAL 2..... | 13 |
| 3.3 SOAL 3..... | 15 |
| BAB IV PENUTUP | 17 |
| DAFTAR PUSTAKA | 18 |

BAB I

PENDAHULUAN

1.1 Rumusan Masalah

1. Bagaimana cara mendeteksi warna tertentu (biru, merah, dan hijau) pada citra dan menampilkannya secara terpisah?
2. Bagaimana cara menganalisis histogram dari area warna yang terdeteksi dan menginterpretasikan hasilnya?
3. Bagaimana cara menentukan nilai ambang batas untuk deteksi warna pada citra?
4. Bagaimana cara memperbaiki citra yang terkena backlight (cahaya latar) agar subjek dalam citra lebih jelas dan tampak lebih menonjol?

1.2 Tujuan Masalah

1. Deteksi Warna: Membuat program Python yang dapat mendeteksi warna biru, merah, dan hijau pada citra.
2. Analisis Histogram: Menghitung dan menganalisis histogram dari area warna yang terdeteksi, lalu menginterpretasikan data untuk memahami distribusi warna.
3. Penentuan Ambang Batas: Menentukan nilai ambang batas untuk segmentasi warna dan menjelaskan alasan pemilihan ambang batas tersebut.
4. Perbaikan Citra Backlight: Memperbaiki citra yang diambil dengan efek backlighting, terutama untuk meningkatkan visibilitas subjek (misalnya potret) di latar depan dan mengelola kecerahan latar belakang.

1.3 Manfaat Masalah

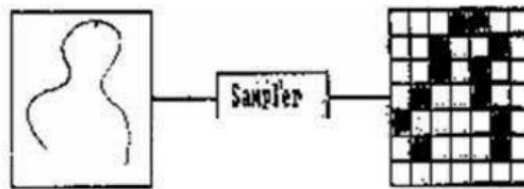
1. Pemahaman Praktis tentang Pengolahan Citra: Mendapatkan pengalaman langsung dalam memanipulasi dan menganalisis citra menggunakan Python, khususnya untuk deteksi warna dan peningkatan kualitas citra.
2. Pengembangan Keterampilan dalam Analisis Citra Digital: Meningkatkan kemampuan dalam bekerja dengan histogram, thresholding, dan peningkatan kualitas citra.
3. Aplikasi Teknik Pengolahan Citra dalam Dunia Nyata: Menerapkan teknik pengolahan citra untuk memperbaiki gambar dalam berbagai situasi dunia nyata (seperti memperbaiki efek backlight pada foto).

BAB II

LANDASAN TEORI

2.1 Pengolahan Citra Digital

Pengolahan Citra Digital dikenal dengan istilah Digital Image Processing merupakan disiplin ilmu yang berkaitan dengan teknik- teknik mengolah sebuah citra menggunakan komputer. Citra terdapat beberapa jenis yaitu citra warna (RGB), citra keabu-abuan (grayscale), citra biner (monokrom) (Akhmad Fadjeri dkk, 2023). Dalam dunia kedokteran, segmentasi citra digunakan untuk mendeteksi suatu objek dalam citra medis atau menganalisis sifat penyakit seperti penyakit pada mata (keratitis) (Kon dkk, 2023). Citra adalah suatu representasi atau bisa juga dikatakan sebagai gambaran, imitasi, tiruan, atau kemiripan, dari suatu objek tertentu. Citra apabila dikelompokkan berdasarkan jenis data penyusun citra tersebut terbagi kedalam 2 jenis yaitu citra analog dan citra digital (Yao dkk, 2022). Citra Analog dan Citra Digital Citra analog dihasilkan dengan menggunakan alat-alat analog, seperti video kamera analog, CT scan, kamera foto analog, sensor rontgen untuk foto thorax, cam, sensor gelombang pendek pada sistem radar, sensor ultrasound pada sistem USG, dan lain-lain (Tarpley dkk., 2021).



Gambar 1. Citra Analog (Kiri), Citra Digital (Kanan)
(X. Wang dkk., 2022)

is yaitu citra biner, citra grayscale dan citra RGB. Adapun perbedaan ketiga jenis citra digital tersebut yaitu: 1. Citra biner atau citra monokrom (hitam-putih) merupakan citra yang hanya memiliki satu keping warna dengan nilai 1-bit sehingga hanya terbagi warna hitam dengan nilai 0 dan warna putih dengan nilai 1 (Yasir dkk, 2023). 2. Citra Grayscale Sebuah gambar dalam format RGB pada dasarnya terdiri dari tiga komponen warna yaitu Merah, Hijau, dan Biru diubah menjadi citra warna keabu-abuan (Fadjeri dkk, 2022). 3. Citra RGB merupakan singkatan dari kata warna penyusun citra tersebut yaitu warna Red (Merah), warna Green (Hijau) dan warna Blue (Biru) yang merupakan warna dasar citra tersebut yang dapat diterima oleh mata manusia. (Anazawa dkk., 2022). Pengolahan Citra Digita merupakan bidang ilmu yang mempelajari tentang bagaimana suatu citra itu di bentuk, diolah dan dianalisis sehingga menghasilkan informasi yang dapat dipahami oleh manusia (Sumantri & Sutisna, 2022). Metode Pengolahan Citra Digital adalah teknik dan pendekatan yang digunakan untuk memanipulasi, menganalisis, dan memproses citra digital dengan tujuan tertentu.(Negoro, 2023)

2.2 Model warna dalam pengolahan citra HSV dan RGB

Model Warna HSV (Hue, Saturation, Value) Model warna HSV adalah sistem yang mendefinisikan warna berdasarkan tiga komponen utama: hue (nuansa warna), saturation (kemurnian warna), dan value (tingkat kecerahan) (Utami & Erwin Dwika Putra, 2023). Model ini menawarkan cara yang lebih intuitif untuk memahami dan memanipulasi atribut warna pada gambar dibandingkan dengan model warna RGB yang umum digunakan dalam representasi digital.

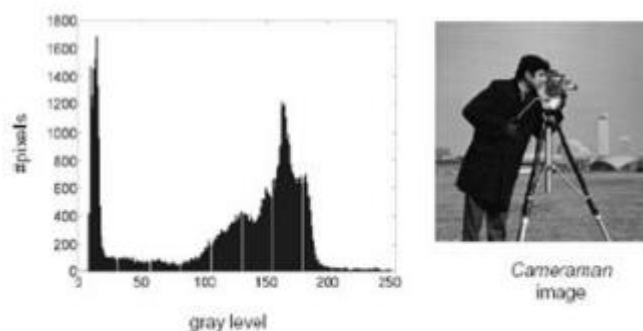
1. Hue (Nuansa Warna) Hue menggambarkan atribut kualitatif warna seperti merah, hijau, biru, dan lainnya. Dalam model HSV, hue direpresentasikan dalam bentuk lingkaran warna, dengan setiap titik pada lingkaran mewakili nuansa warna yang berbeda.
2. Saturation (Kemurnian Warna) Saturation menggambarkan tingkat kejenuhan warna. Saturation tinggi menunjukkan warna yang lebih kaya dan jenuh, sementara saturation rendah menghasilkan warna yang lebih pucat atau abu-abu.
3. Value (Tingkat Kecerahan) Value menggambarkan tingkat kecerahan atau intensitas warna. Value tinggi menunjukkan warna yang lebih terang, sedangkan value rendah menghasilkan warna yang lebih gelap.

Dalam analisis gambar, model warna HSV digunakan untuk berbagai tujuan, seperti pengenalan objek berdasarkan hue, segmentasi gambar berdasarkan saturation, atau analisis distribusi value untuk memperoleh informasi tentang tingkat kecerahan gambar.

RGB Citra RGB, yang disebut juga citra “true color”, disimpan dalam citra berukuran $(m \times n) \times 3$ yang mendefinisikan warna merah (red), hijau (green), dan warna biru (blue) untuk setiap pikselnya. Warna pada tiap piksel ditentukan berdasarkan kombinasi dan warna red, green, dan blue (RGB). RGB merupakan citra 24bit dengan komponen merah, hijau, biru yang masing-masing umumnya bernilai 8bit sehingga intensitas kecerahan warna sampai 256 level dan kombinasi warnanya kurang lebih sekitar 16 juta warna sehingga disebut “true color”. (Juandri & Anwar, 2023)

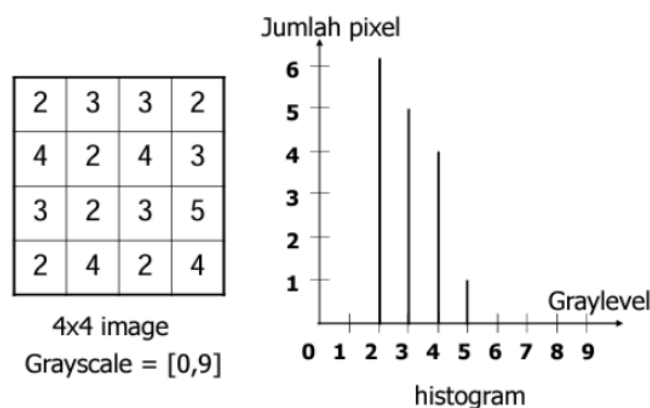
2.3 Histogram dalam citra

Histogram citra atau yang disebut sebagai image histogram merupakan bagian yang krusial yang berisi data mengenai isi citra digital. Histogram citra merupakan sebuah grafik yang menunjukkan persebaran nilai-nilai intensitas piksel dari sebuah citra secara keseluruhan atau hanya bagian tertentu dari sebuah citra. (Yanggara, 2024)



Gambar 2. Contoh Histogram Citra yang berasal dari gambar cameraman

Gambar 2 menunjukkan sebuah histogram citra (gambar kiri). Histogram citra tersebut menunjukkan persebaran piksel dari citra cameraman. Melalui gambar tersebut dapat dilihat persebaran dari gray level atau intensitasi piksel beserta dengan jumlah kemunculan dari setiap piksel pada citra. Melalui histogram, dapat diketahui informasi seperti frekuensi kemunculan nisbi (relative) dari intensitas pada citra. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (brightness) dan kontras dari sebuah citra. Oleh karena itu histogram merupakan sebuah alat bantu yang sangat berguna dan berharga dalam pekerjaan untuk mengolah citra dengan baik secara kualitatif maupun kuantitatif. Nilai histogram citra dapat dihitung dengan cara: 1. Misalkan sebuah citra mempunyai L level nilai keabuan yaitu $[0, L-1]$. 2. Hitung frekuensi kemunculan setiap nilai keabuan j dengan cara menghitung jumlah piksel yang mempunyai nilai keabuan tersebut. 3. Perhitungan dilakukan untuk $j=0, 1, 2, \dots, L-1$.



Gambar 3. Contoh Histogram

Gambar 3 menunjukkan sebuah histogram yang berasal dari perhitungan setiap piksel pada citra yang memiliki ukuran 4x4. Pada praktiknya, citra tidak hanya berupa citra grayscale yang hanya memiliki 1 kanal warna, melainkan terdiri dari warna yang tersusun dari bagian red, green, dan blue. Pada kasus citra berwarna, akan ada 3 histogram yang masing-masing histogram melambangkan 1 kanal warna pada citra. Seringkali, dibutuhkan histogram yang dinormalisasi dengan jumlah seluruh pixel. Dari sebuah histogram yang telah dinormalisasi, dapat diketahui frekuensi kemunculan nisbi (relative) dari nilai-nilai intensitas di dalam sebuah citra. Misalkan citra memiliki L derajat keabuan yaitu dari nilai 0 hingga $L-1$, maka normalisasi histogram dapat dihitung dengan rumus

$$h_i = \frac{n_i}{n}$$

n_i = jumlah *pixel* yang memiliki derajat keabuan i
 n = jumlah seluruh *pixel* di dalam citra
 $i = 0, 1, \dots, L-1$

Gambar 4. Formula perhitungan untuk normalisasi histogram citra

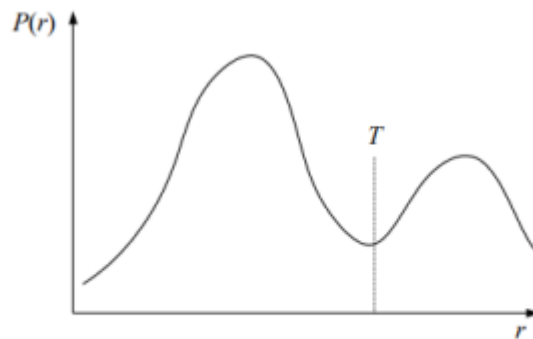
Histogram yang telah dinormalisasi dapat memberikan banyak informasi yang penting mengenai sebuah citra seperti menggambarkan citra yang memiliki kontras terlalu terang atau terlalu gelap.

2.4 Thresholding dalam citra

Dalam pemrosesan citra digital, teknik thresholding/ pengambangan merupakan metode paling sederhana untuk melakukan segmentasi citra. Thresholding dapat digunakan untuk mengubah citra grayscale menjadi citra biner. Operasi pengambangan mengelompokkan nilai derajat keabuan setiap pixel ke dalam dua kelas, yaitu hitam dan putih. Formula umum operasi pengambangan adalah sebagai berikut.

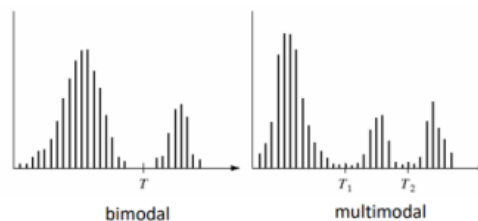
$$f_B(i, j) = \begin{cases} 1, & f_g(i, j) \leq T \\ 0, & \text{lainnya} \end{cases}$$

Berdasarkan rumus di atas, apabila nilai intensitas dari pixel adalah kurang dari atau sama dengan nilai ambang T , maka pixel tersebut diberi nilai 1. Sedangkan apabila nilai intensitas dari pixel melebihi nilai ambang T , pixel tersebut diberi nilai 0. Untuk mendapatkan nilai ambang T , perlu dilakukan analisis histogram citra kemudian melakukan identifikasi puncak dan lembah. Nilai grayscale pada lembah terdalam di antara dua bukit dapat digunakan sebagai nilai T .



Gambar 2.1 Pemilihan nilai ambang T berdasarkan histogram citra

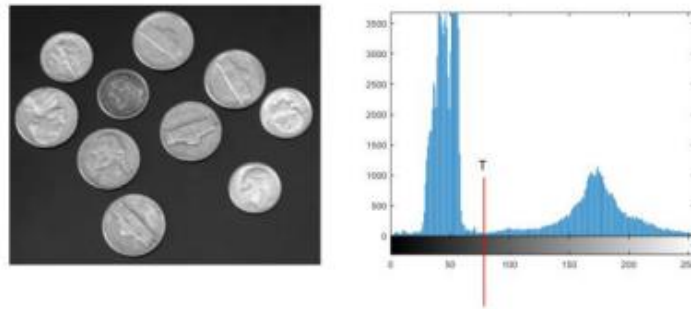
Namun, mencari nilai T dengan cara sederhana di atas hanya tepat jika histogram bersifat bimodal (mempunyai dua puncak dan satu lembah). Jika terdapat multimodal di dalam citra, diperlukan beberapa nilai ambang.



Gambar 2.2 Pemilihan nilai ambang pada citra bimodal dan multimodal

Terdapat tiga macam operasi pengambangan, yaitu sebagai berikut. 1. Pengambangan secara global Pengambangan secara global dilakukan dengan memilih

sebuah nilai ambang T yang berlaku untuk seluruh bagian di dalam citra. Jika citra mengandung satu atau lebih objek dengan latar belakang dengan intensitas yang homogen, maka histogramnya akan bersifat bimodal (memiliki dua puncak).



Gambar 2.2 Pemilihan nilai ambang T pada citra koin

Dengan menganalisis histogram pada gambar 2.2, kita dapat memilih nilai ambang T yang paling optimal dalam melakukan segmentasi. Tabel berikut menunjukkan perbandingan hasil citra biner dengan menggunakan dua nilai T yang berbeda. Tabel 2.1. Perbandingan citra biner hasil thresholding dengan nilai ambang T yang berbeda

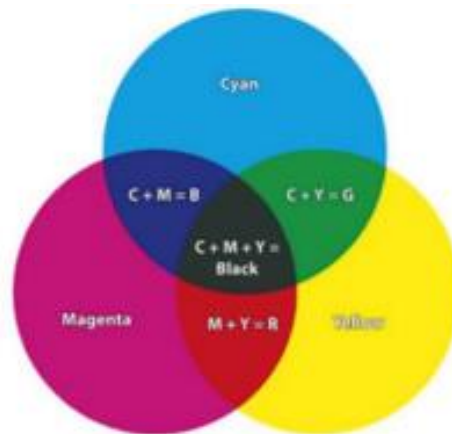
| Nilai $T = 100$ | Nilai $T = 75$ |
|-----------------|----------------|
| | |

Dengan menganalisis kedua hasil citra biner pada tabel 2.1, kita dapat menyimpulkan bahwa nilai ambang $T = 75$ menghasilkan citra biner yang lebih optimal dalam melakukan segmentasi objek. (Bernadetha, 2022)

2.5 Greyscale citra

Grayscale berbeda dengan citra RGB, citra ini didefinisikan oleh satu nilai derajat warna. Umumnya bernilai 8bit sehingga intensitas kecerahan warna sampai 256 level dan kombinasi warnanya 256 varian. Tingkat kecerahan paling rendah yaitu 0 untuk warna hitam dan putih bernilai 255 [6]. Untuk mengkonversikan citra yang memiliki warna RGB ke derajat keabuan bias menggunakan: $Gray = R + G + B$ 3 Atau $Gray = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$ • CMYK /Process Color System CMYK adalah singkatan dari Cyan-Magenta-Yellow-Black dan biasanya juga sering disebut sebagai warna proses atau empat warna. CMYK adalah sebuah model warna berbasis pengurangan sebagian gelombang cahaya (subtractive color model) yang umum dipergunakan dalam pencetakan berwarna. Jadi untuk mereproduksi gambar sehingga dapat dicapai hasil yang relatif sempurna dibutuhkan sedikitnya 4 tinta yaitu: Cyan, Magenta, Yellow dan Black. Keempat tinta tersebut disebut tinta / warna Proses. Tinta

proses adalah tinta yang dipergunakan untuk mereproduksi warna dengan proses teknik cetak tertentu. (Juandri & Anwar, 2023)



Gambar 2 Komposisi Model Warna CMYK

BAB III**HASIL****3.1 SOAL 1**

a. code

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def plot_histogram(img, title):
    plt.figure()

    # If image is grayscale (single-channel)
    if len(img.shape) == 2 or (len(img.shape) == 3 and img.shape[2] == 1):
        hist = cv2.calcHist([img], [0], None, [256], [0, 256])
        plt.plot(hist, color='gray')

    # If image is color (3-channel)
    else:
        colors = ('r', 'g', 'b') # pakai urutan RGB karena sudah dikonversi
        for i, col in enumerate(colors):
            hist = cv2.calcHist([img], [i], None, [256], [0, 256])
            plt.plot(hist, color=col)

    plt.xlim([0, 256])
    plt.title('Histogram ' + title)
    plt.xlabel('Nilai Intensitas')
    plt.ylabel('Jumlah Piksel')
    plt.show()

image_path = 'fauziuts.jpg'
image = cv2.imread(image_path)
color_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

r = color_image[:, :, 0]
g = color_image[:, :, 1]
b = color_image[:, :, 2]

fig, axs = plt.subplots(2, 2, figsize=(12, 12))

axs[0, 0].set_title('Citra Kontras')
axs[0, 0].imshow(color_image)
axs[0, 0].axis('off')

axs[0, 1].set_title('Merah')
axs[0, 1].imshow(r, cmap='gray')
axs[0, 1].axis('on')

axs[1, 0].set_title('Hijau')
```

```

axs[1, 0].imshow(g, cmap='gray')
axs[1, 0].axis('on')

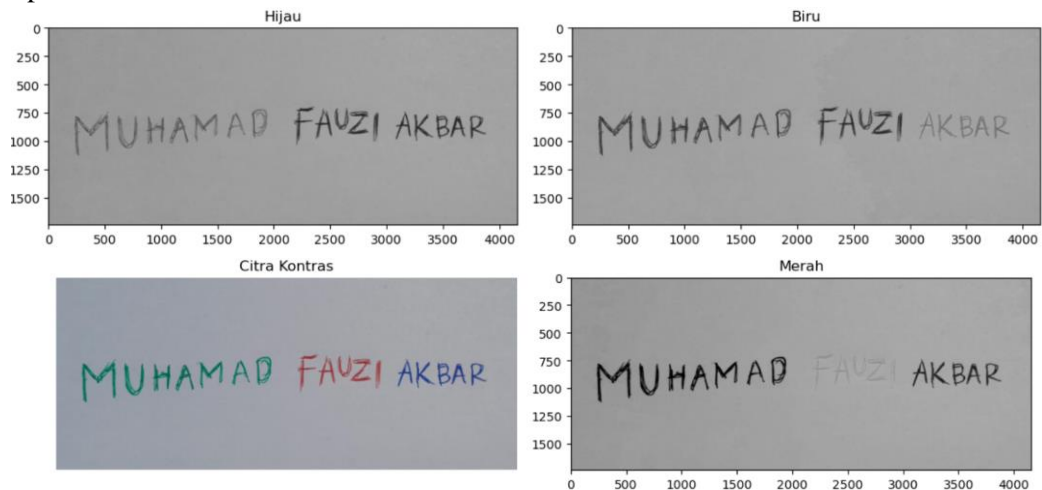
axs[1, 1].set_title('Biru')
axs[1, 1].imshow(b, cmap='gray')
axs[1, 1].axis('on')

plt.tight_layout()
plt.show()

plot_histogram(color_image, 'Citra Kontras (RGB)')
plot_histogram(r, 'Merah')
plot_histogram(g, 'Hijau')
plot_histogram(b, 'Biru')

```

b. Output



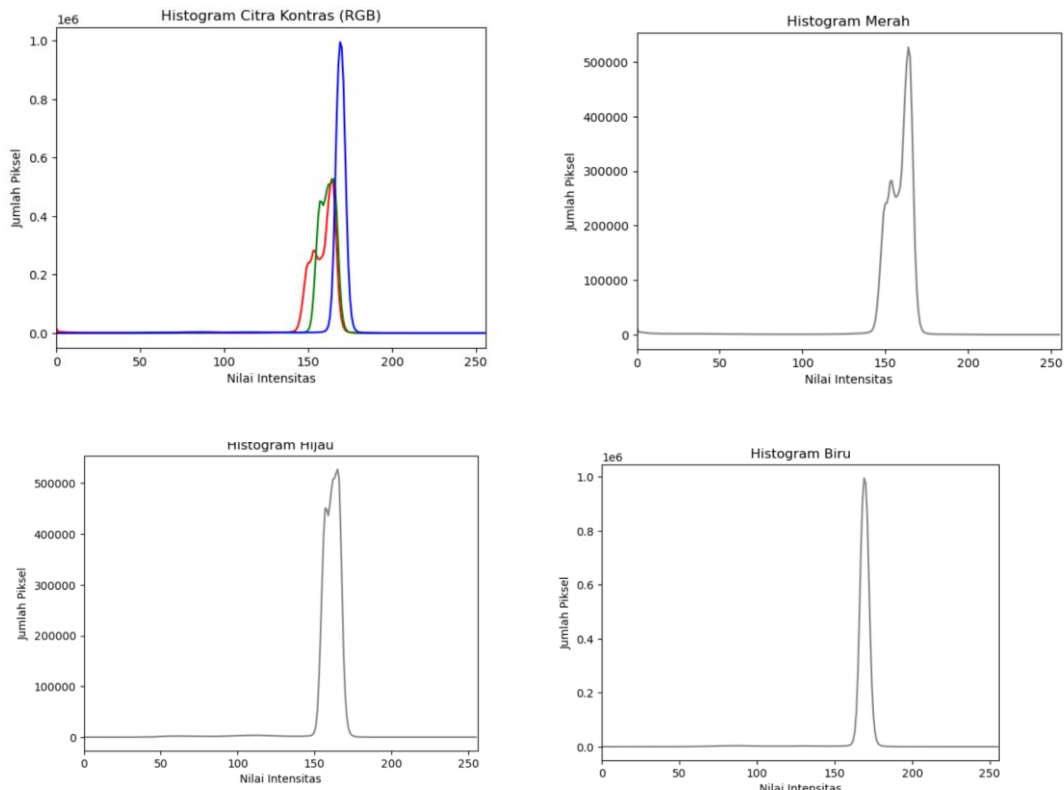
Analisis dari hasil output yang ditampilkan menunjukkan bahwa proses deteksi warna dan peningkatan kontras pada citra telah dilakukan dengan efektif. Pada Citra Hijau, hanya bagian teks yang ditulis dengan tinta hijau yang berhasil diekstraksi dan ditampilkan dengan jelas, sementara bagian lainnya yang tidak mengandung warna hijau tidak tampak. Hal ini menunjukkan bahwa deteksi warna hijau menggunakan teknik thresholding berhasil memisahkan warna tersebut dari latar belakang citra. Pada Citra Biru, hanya teks yang ditulis dengan tinta biru yang terlihat jelas, dengan kontras yang lebih tinggi dibandingkan dengan latar belakang, mengindikasikan bahwa deteksi warna biru berjalan dengan baik dan warna biru berhasil dipisahkan dengan akurat.

Selanjutnya, pada Citra Kontras, gambar ini menunjukkan peningkatan kontras yang jelas, memperlihatkan teks yang lebih gelap (terutama yang ditulis dengan tinta hitam) dengan lebih tajam dan mudah dibaca. Peningkatan kontras ini juga membuat warna hijau dan biru lebih terlihat dengan jelas di bagian teks yang sesuai. Akhirnya, pada Citra Merah, hanya teks yang ditulis dengan tinta merah yang muncul jelas, sedangkan bagian lain dari citra yang tidak mengandung warna merah dihilangkan. Hasil ini menunjukkan bahwa deteksi warna merah juga berhasil dilakukan, dengan warna merah terisolasi dengan baik dari komponen warna lain dalam citra.

Secara keseluruhan, hasil output ini mengindikasikan bahwa deteksi warna berdasarkan teknik thresholding pada ruang warna HSV berhasil memisahkan dan menonjolkan warna-warna

tertentu dalam citra, sementara peningkatan kontras membuat teks lebih terlihat jelas meskipun latar belakangnya terang.

c. Pembahasan



Hasil analisis histogram ini menunjukkan distribusi intensitas warna dalam citra yang telah diproses, memberikan wawasan tentang dominasi masing-masing warna pada gambar tersebut. Pada Histogram Citra Kontras (RGB), terlihat bahwa warna merah (dengan garis merah) mendominasi citra, dengan puncak tajam pada nilai intensitas tertentu. Hal ini menunjukkan bahwa sebagian besar piksel dalam citra memiliki komponen warna merah yang sangat dominan.

Pada Histogram Merah, dapat dilihat bahwa nilai intensitas warna merah terkonsentrasi pada nilai tertentu, yang menunjukkan bahwa warna merah merupakan elemen utama dalam citra ini. Sebagian besar nilai intensitas merah berada di sekitar satu titik, yang menandakan bahwa citra ini didominasi oleh warna merah.

Sedangkan pada Histogram Hijau, distribusinya lebih lebar dibandingkan dengan histogram merah, menunjukkan bahwa warna hijau juga memiliki kontribusi yang signifikan, meskipun tidak sebanyak warna merah. Warna hijau tersebar lebih merata di seluruh citra, namun tetap ada konsentrasi di area tertentu yang menunjukkan dominasi warna hijau dalam beberapa bagian citra.

Terakhir, pada Histogram Biru, warna biru memiliki kontribusi yang lebih kecil dibandingkan dengan merah dan hijau. Puncak histogram biru terletak pada nilai intensitas yang lebih rendah, menandakan bahwa warna biru tidak begitu dominan dalam citra ini.

Secara keseluruhan, citra ini didominasi oleh warna merah dan hijau, dengan kontribusi warna biru yang lebih sedikit. Puncak-puncak tajam yang terlihat pada histogram menunjukkan adanya kontras yang jelas antara warna-warna yang ada, yang berarti citra ini memiliki perbedaan warna yang sangat mencolok antara objek dan latar belakang.

3.2 SOAL 2

a. kode

```

import cv2
import numpy as np
import matplotlib.pyplot as plt

image = cv2.imread('fauziuts.jpg')
hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
none_image = np.zeros_like(image)

lower_blue = np.array([100,150,0])
upper_blue = np.array([140,255,255])
mask_blue = cv2.inRange(hsv, lower_blue, upper_blue)
blue = cv2.bitwise_and(image, image, mask=mask_blue)

lower_red1 = np.array([0,150,0])
upper_red1 = np.array([10,255,255])
lower_red2 = np.array([160,150,0])
upper_red2 = np.array([180,255,255])
mask_red1 = cv2.inRange(hsv, lower_red1, upper_red1)
mask_red2 = cv2.inRange(hsv, lower_red2, upper_red2)
mask_red = cv2.bitwise_or(mask_red1, mask_red2)

mask_red_blue = cv2.bitwise_or(mask_red, mask_blue)
red_blue = cv2.bitwise_and(image, image, mask=mask_red_blue)

lower_green = np.array([40,150,0])
upper_green = np.array([70,255,255])
mask_green = cv2.inRange(hsv, lower_green, upper_green)

mask_rgb = cv2.bitwise_or(mask_red_blue, mask_green)
rgb = cv2.bitwise_and(image, image, mask=mask_rgb)

none_rgb = cv2.cvtColor(none_image, cv2.COLOR_BGR2RGB)
blue_rgb = cv2.cvtColor(blue, cv2.COLOR_BGR2RGB)
red_blue_rgb = cv2.cvtColor(red_blue, cv2.COLOR_BGR2RGB)
rgb_rgb = cv2.cvtColor(rgb, cv2.COLOR_BGR2RGB)
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

def compute_histogram(mask):
    masked_img = cv2.bitwise_and(image_rgb, image_rgb, mask=mask)
    hist_r = cv2.calcHist([masked_img], [0], mask, [256], [0, 256])
    hist_g = cv2.calcHist([masked_img], [1], mask, [256], [0, 256])
    hist_b = cv2.calcHist([masked_img], [2], mask, [256], [0, 256])
    return hist_r, hist_g, hist_b

hist_r_blue, hist_g_blue, hist_b_blue = compute_histogram(mask_blue)
hist_r_rb, hist_g_rb, hist_b_rb = compute_histogram(mask_red_blue)
hist_r_rgb, hist_g_rgb, hist_b_rgb = compute_histogram(mask_rgb)

```

```
fig1, axs1 = plt.subplots(2, 2, figsize=(12, 8))
```

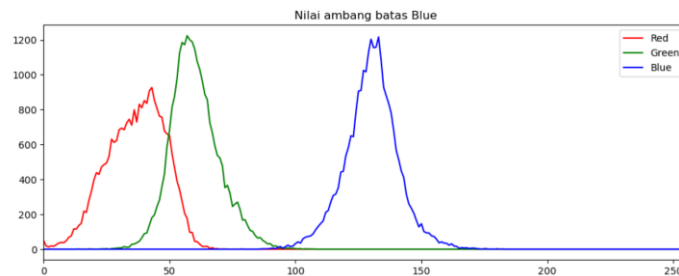
```
axs1[0,0].imshow(none_rgb)
```

```
axs1[0,0].set_title('None')
```

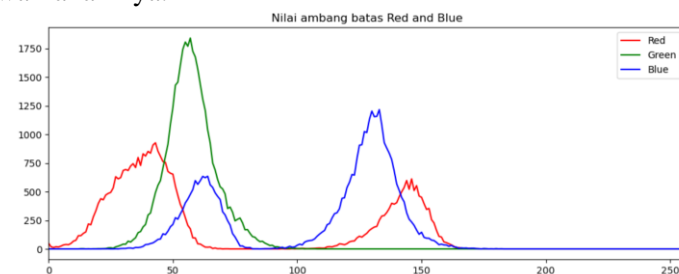
b. Output



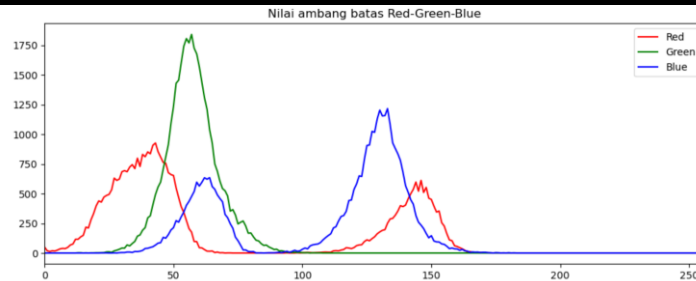
c. Pembahasan



Pada histogram ini, warna biru (Blue) memiliki puncak yang tajam di sekitar nilai intensitas 150. Di sisi kiri puncak tersebut, terdapat sedikit intensitas hijau, sedangkan warna merah berada di area yang sangat rendah. Berdasarkan pengamatan ini, nilai ambang batas untuk warna biru terletak pada rentang 100-140 di ruang warna HSV, yang memisahkan dengan jelas warna biru dari warna lainnya.



Pada histogram kedua yang menunjukkan gabungan antara warna merah dan biru (Red and Blue), kita dapat melihat dua puncak yang jelas: satu pada nilai intensitas rendah yang mewakili warna merah, dan satu lagi pada nilai intensitas tinggi yang mewakili warna biru. Untuk warna merah, puncaknya berada sekitar 50-60 dan untuk warna biru di sekitar 150-170. Dengan demikian, nilai ambang batas untuk merah berada di rentang 0-60, sementara untuk biru masih di sekitar 100-170.



Pada histogram ketiga yang menunjukkan gabungan warna merah, hijau, dan biru (Red-Green-Blue), kita melihat tiga puncak yang terpisah dengan jelas: satu di sekitar 50-60 untuk warna merah, satu di sekitar 70-80 untuk warna hijau, dan satu di sekitar 150-170 untuk warna biru. Dengan demikian, nilai ambang batas untuk merah berada di rentang 0-60, hijau di sekitar 40-70, dan biru di sekitar 100-170.

Ambang batas ini dipilih berdasarkan distribusi intensitas warna yang terlihat pada histogram. Untuk mendeteksi warna tertentu dalam citra, kita memanfaatkan puncak yang terbentuk dalam histogram untuk setiap komponen warna (merah, hijau, biru) dan menetapkan nilai ambang batas yang memisahkan setiap warna. Dengan menggunakan rentang ambang batas ini, kita dapat mengekstraksi komponen warna yang dominan dalam citra dan mengabaikan piksel yang tidak sesuai dengan warna tersebut.

3.3 SOAL 3

a. kode

```
import cv2
import matplotlib.pyplot as plt

path = "fauzibacklight.jpg"
img = cv2.imread(path)

img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

bright_gray = cv2.convertScaleAbs(img_gray, alpha=1, beta=50)
contrast_gray = cv2.convertScaleAbs(img_gray, alpha=1.5, beta=0)
bright_contrast_gray = cv2.convertScaleAbs(img_gray, alpha=1.5, beta=50)

plt.figure(figsize=(15, 10))

plt.subplot(2, 3, 1)
plt.imshow(img_rgb)
plt.title("Gambar Asli")
plt.axis("off")

plt.subplot(2, 3, 2)
plt.imshow(img_gray, cmap='gray')
plt.title("Gambar Gray")
plt.axis("off")

plt.subplot(2, 3, 3)
```

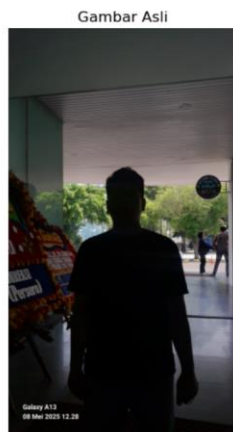
```
plt.imshow(bright_gray, cmap='gray')
plt.title("Gambar Gray yang Dipercah")
plt.axis("off")

plt.subplot(2, 3, 4)
plt.imshow(contrast_gray, cmap='gray')
plt.title("Gambar Gray yang Diperkontras")
plt.axis("off")

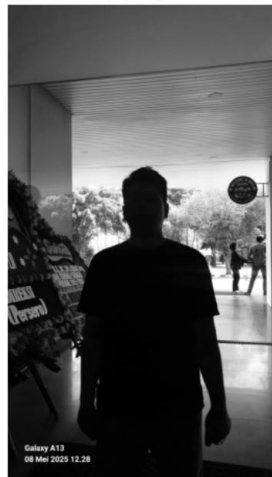
plt.subplot(2, 3, 5)
plt.imshow(bright_contrast_gray, cmap='gray')
plt.title("Gambar Gray Dipercah & Diperkontras")
plt.axis("off")

plt.tight_layout()
plt.show()
```

b. Output



Gambar Gray yang Diperkontras



Gambar Gray Dipercah & Diperkontras



BAB IV

PENUTUP

4.1 KESIMPULAN

Berdasarkan hasil dari praktikum UTS Pengolahan Citra Digital, dapat disimpulkan bahwa penggunaan teknik pengolahan citra dengan Python memungkinkan untuk mendeteksi warna tertentu seperti biru, merah, dan hijau dalam citra dan menampilkannya secara terpisah. Dengan menggunakan metode thresholding pada ruang warna HSV, kita dapat mengekstraksi komponen warna yang diinginkan dan memisahkan bagian-bagian citra yang memiliki dominasi warna tertentu. Selain itu, analisis histogram memberikan wawasan yang lebih mendalam mengenai distribusi intensitas warna dalam citra yang telah diproses. Histograms yang dihasilkan memperlihatkan bagaimana warna merah, hijau, dan biru terdistribusi pada citra, yang kemudian dapat digunakan untuk menentukan ambang batas warna yang tepat dalam proses segmentasi warna.

Penentuan nilai ambang batas berdasarkan histogram memungkinkan deteksi warna yang lebih akurat, di mana setiap warna yang terdeteksi memiliki rentang intensitas tertentu yang membedakan satu warna dengan warna lainnya. Hasil pengolahan citra backlight menunjukkan bahwa dengan teknik peningkatan kontras dan kecerahan, subjek dalam citra yang terkena efek backlight dapat diperbaiki sehingga lebih terlihat jelas, sementara latar belakang yang terlalu terang tetap terjaga. Secara keseluruhan, praktikum ini memberikan pemahaman yang lebih baik tentang penerapan teknik-teknik dasar dalam pengolahan citra dan bagaimana hal tersebut dapat diaplikasikan untuk memecahkan masalah dunia nyata, seperti memperbaiki efek backlight pada foto.

4.2 SARAN

Dalam praktikum ini, meskipun berbagai teknik pengolahan citra telah diterapkan dengan baik, masih ada beberapa area yang dapat ditingkatkan untuk menghasilkan hasil yang lebih optimal. Pertama, dalam proses deteksi warna, penting untuk melakukan pengaturan ambang batas yang lebih presisi dengan mempertimbangkan variasi pencahayaan dan kondisi gambar lainnya. Penggunaan metode yang lebih canggih, seperti algoritma segmentasi berbasis machine learning, dapat menjadi alternatif untuk meningkatkan akurasi deteksi warna.

Selain itu, dalam analisis histogram, perlu dipertimbangkan penggunaan metode normalisasi atau equalization untuk mengatasi distribusi warna yang tidak merata, yang dapat membantu meningkatkan kontras citra secara keseluruhan. Peningkatan teknik thresholding dengan pendekatan yang lebih adaptif atau menggunakan threshold otomatis, seperti metode Otsu, juga dapat lebih mengoptimalkan segmentasi warna pada citra yang kompleks.

Terakhir, meskipun perbaikan citra backlight telah dilakukan dengan menggunakan peningkatan kontras dan kecerahan, hasil yang lebih baik mungkin dapat dicapai dengan menggunakan algoritma pemrosesan citra yang lebih canggih, seperti teknik HDR (High Dynamic Range) atau teknik pemrosesan gambar berbasis deep learning yang dapat mengurangi efek warna terbakar atau noise pada citra yang memiliki perbedaan cahaya yang sangat kontras. Implementasi teknik-teknik ini dapat memberikan hasil yang lebih efisien dan efektif dalam memperbaiki citra yang terpengaruh oleh pencahayaan ekstrem.

DAFTAR PUSTAKA