

UTS
PENGOLAHAN CITRA



NAMA : Yolanda Kristine Br Sembiring

NIM : 202331166

KELAS : D

DOSEN : Darma Rusjdi,Ir.M.kom

NO.PC : 4

ASISTEN : 1. Fakhrol Fauzi Nugraha Tarigan

2. Muhammad Hanief Febriansyah

3. Clarenca Sweetdiva Pereira

4. Sakura Amastasya Salsabila Setiyanto

INSTITUT TEKNOLOGI PLN
TEKNIK INFORMATIKA
2024/2025

DAFTAR ISI

BAB 1 PENDAHULUAN.....	1
1.1 Rumusan Masalah.....	1
1.2 Tujuan Masalah.....	1
1.3 Manfaat Masalah.....	1
BAB II LANDASAN TEORI.....	2
BAB III HASIL.....	4
1. DETEKSI WARNA PADA CITRA.....	4
2. HISTOGRAM.....	5
3. NILAI AMBANG BATAS CITRA.....	6
4. MEMPERBAIKI GAMBAR BACKLIGHT.....	8
BAB IV PENUTUP.....	10
Kesimpulan.....	10
DAFTAR PUSTAKA.....	11

BAB I

PENDAHULUAN

1.1 Rumusan Masalah

- Bagaimana cara mendeteksi warna dalam citra digital menggunakan pemisahan kanal warna RGB?
- Bagaimana visualisasi histogram dapat membantu dalam analisis distribusi intensitas warna pada gambar?
- Bagaimana teknik nilai ambang (thresholding) dalam ruang warna HSV dapat digunakan untuk mendeteksi warna spesifik dalam gambar?
- Bagaimana cara memperbaiki kualitas gambar yang memiliki pencahayaan belakang (backlight) agar objek dalam gambar lebih terlihat jelas?

1.2 Tujuan Masalah

- **Mendeteksi Warna:** Telah berhasil memisahkan dan menganalisis kanal warna Merah, Hijau, dan Biru dalam sebuah citra untuk melihat kontras dan intensitas masing-masing kanal.
- **Histogram Warna:** Telah dibuat dan dianalisis histogram dari setiap kanal warna untuk memahami distribusi piksel dalam gambar, yang menunjukkan bahwa gambar memiliki dominasi warna terang dan pola yang seimbang.
- **Thresholding (HSV):** Telah diterapkan metode thresholding di ruang warna HSV untuk mendeteksi dan menampilkan area gambar berdasarkan warna utama (merah, hijau, biru), menunjukkan bahwa gambar tersebut memang menggunakan kombinasi RGB.
- **Perbaikan Backlight:** Gambar dengan pencahayaan belakang telah berhasil diperbaiki melalui teknik konversi ke grayscale serta peningkatan kecerahan dan kontras untuk memperjelas objek utama dalam gambar.

1.3 Manfaat Masalah

- Memberikan pemahaman praktis mengenai bagaimana warna dalam citra digital dapat diurai dan dianalisis melalui kanal RGB.
- Membantu memahami bagaimana histogram digunakan untuk menganalisis kualitas warna dan distribusi piksel dalam gambar.
- Menunjukkan efektivitas penggunaan ruang warna HSV dalam mendeteksi warna tertentu, yang sangat berguna dalam segmentasi objek berbasis warna.
- Menawarkan solusi pengolahan citra untuk memperbaiki gambar berkualitas rendah akibat pencahayaan buruk, yang bermanfaat dalam bidang fotografi, pengawasan, dan pengenalan citra.

BAB II

LANDASAN TEORI

1. Citra Digital

Citra digital adalah gambar dua dimensi yang dihasilkan dari analog dua dimensi yang kontinu menjadi gambar melalui proses sampling. Gambar analog dibagi menjadi N baris dan M kolom sehingga menjadi gambar diskrit. Citra digital merupakan citra yang dapat diolah komputer. Yang disimpan dalam komputer hanyalah angka-angka yang menunjukkan besar intensitas pada masing-masing piksel. Karena berbentuk data numerik, maka citra digital dapat diolah dengan komputer.

2. Pengolahan Citra Digital

Pengolahan citra digital adalah ilmu yang mempelajari hal-hal berkaitan dengan perbaikan kualitas terhadap suatu gambar (meningkatkan kontras, perubahan warna, restorasi citra), transformasi gambar (translasi, rotasi transformasi, skala, geometrik), melakukan pemilihan citra ciri (feature images) yang optimal untuk tujuan analisis, melakukan penyimpanan data yang sebelumnya dilakukan reduksi dan kompresi, transmisi data, dan waktu proses data.

3. Prinsip Dasar Pengolahan Citra

- Kecerahan dan kontras Gambar yang diambil oleh lensa kamera terlihat gelap karena kurangnya intensitas cahaya dapat diperbaiki dengan pengolahan citra berupa peningkatan kontras. Cara kerjanya yaitu dengan meningkatkan persepsi objek dalam pemandangan dengan meningkatkan perbedaan kecerahan antara objek dan latar belakangnya. Peningkatan kontras biasanya dilakukan sebagai peregangan kontras diikuti oleh peningkatan nada, meskipun keduanya dapat dilakukan dalam satu langkah.
- Penghilangan noise Gambar yang akan diproses seringkali dalam keadaan terdistorsi atau mengandung derau (noise). Untuk kepentingan tertentu, derau tersebut perlu dibersihkan terlebih dulu dengan cara mengurangi atau menghilangkan visibilitasnya melalui proses menghaluskan seluruh gambar meninggalkan area di dekat batas kontras. Proses ini diperlukan untuk mendapatkan gambar dengan kualitas lebih baik dan bebas dari objek yang mengganggu.

4. Jenis Citra

- Citra berwarna
Citra berwarna dikenal dengan istilah truecolor atau citra RGB, merupakan jenis citra yang menyajikan warna dalam bentuk komponen R

(red/merah), G (green/hijau), dan B (blue/biru) yang jika digabungkan akan membentuk suatu susunan warna yang luas. Setiap komponen warna menggunakan 8 bit (nilainya berkisar antara 0 hingga 255). Oleh karena itu, kemungkinan warna yang bisa disajikan mencapai $255 \times 255 \times 255$ atau 16.581.375 warna. Satu jenis warna, dapat dimisalkan sebagai sebuah vektor di ruang 3 dimensi yang biasanya dipakai dalam matematika. Koordinatnya dinyatakan dalam bentuk x , y , dan z . Misalkan sebuah vektor dituliskan sebagai $r = (x, y, z)$, maka untuk untuk setiap citra, komponen-komponen tersebut digantikan oleh komponen RGB. Dapat dituliskan sebagai berikut untuk citra berwarna = RGB(30,75,255). Putih = RGB (255,255,255), sedangkan untuk hitam= RGB(0,0,0).

- Citra berskala keabuan

Citra ini memiliki gradasi hitam dan putih hingga menghasilkan warna abu. Biasa juga dikenal dengan istilah gray scale atau gray level. Pada jenis gambar ini piksel bergantung dengan intensitasnya. Kelas array dalam citra keabuan bisa terdiri dari single, double, uint8, maupun int16. Dalam hal ini, intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih.

- Citra biner

Citra biner juga dikenal dengan istilah binary bilevel image. Piksel dalam citra biner hanya dapat mengasumsikan dua kemungkinan nilai, 0 atau 1. Dimana nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Gambar biner dinilai paling ‘murah’ karena menggunakan penyimpanan dan juga persyaratan pemrosesan yang paling sedikit. Contoh gambar biner adalah gambar garis, teks tercetak pada halaman putih, atau siluet. Gambar-gambar ini berisi informasi yang cukup tentang objek dalam gambar dan kita dapat mengenalinya dengan mudah. Dalam computer vision citra biner biasa digunakan untuk pengenalan objek, dan pelacakan, Citra tingkat keabuan atau grayscale dapat diubah menjadi citra biner dengan proses thresholding.

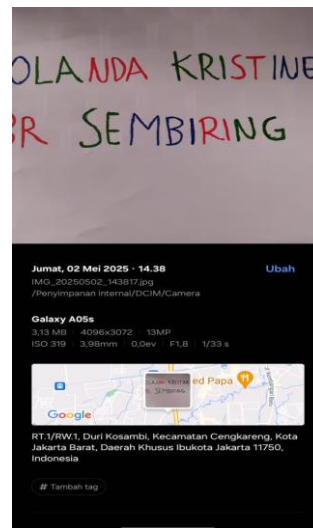
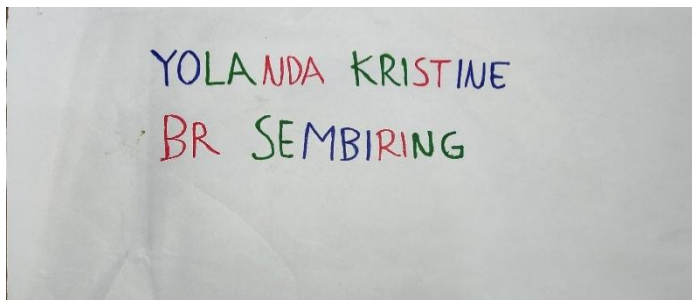
5. Histogram Citra

Histogram citra merupakan grafik yang dapat menggambarkan penyebaran intensitas piksel terhadap jumlah piksel pada intensitas tertentu. Oleh sebab itu, histogram merupakan alat bantu yang sangat penting dalam pengelolaan citra baik secara kualitatif maupun secara kuantitatif. Misalkan citra digital memiliki L derajat keabuan, yaitu dari nilai 0 sampai dengan $L - 1$. Secara matematis, histogram citra dapat dihitung dengan persamaan.

BAB III

HASIL

1. DETEKSI WARNA PADA CITRA

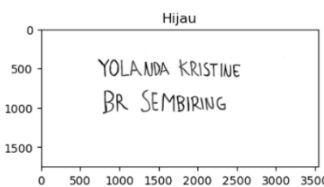


HASIL/OUTPUT

(-0.5, 3539.5, 1751.5, -0.5)

Citra Kontras

YOLANDA KRISTINE
BR SEMBIRING



ANALISIS:

Gambar diatas disusun dari tiga kanal warna:

- Red (Merah): merepresentasikan intensitas warna merah
- Green (Hijau): merepresentasikan intensitas warna hijau
- Blue (Biru): merepresentasikan intensitas warna biru

Ketika kita menampilkan hanya satu kanal Warna yang sesuai dengan kanal tersebut akan tampak pudar atau hilang, karena warnanya “menyatu” dengan latar. Sebaliknya, huruf dengan warna yang berbeda dari kanal tersebut akan tampak lebih kontras dan jelas.

- Pada kanal Merah menampilkan bahwa warna merah yang berada pada gambar tersebut pudar dan warna lain tampak sangat jelas, karena kanal merah menyerap huruf/gambar yang berwarna merah sehingga pada warna tersebut pudar dan warna lain terlihat jelas.

- Pada kanal Hijau menampilkan bahwa warna Hijau yang berada pada gambar tersebut tampak sedikit pudar jika kita sangat memperhatikan/zoom dan warna lain tampak sangat jelas, karena kanal Hijau menyerap huruf/gambar yang berwarna Hijau sehingga pada warna tersebut sedikit pudar pudar dan warna lain terlihat jelas.
- Pada kanal Biru menampilkan bahwa warna biru yang berada pada gambar tersebut pudar bahkan hampir hilang dan warna lain tampak sangat jelas, karena kanal biru menyerap huruf/gambar yang berwarna biru sehingga pada warna tersebut hampir hilang dan warna lain terlihat jelas.

Source Code

```

IMPORT LIBRARY

[44]: import cv2
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as img

[98]: img = cv2.imread('coloring.jpg')

[88]: (baris, kolom) = img_rgb.shape[:2]

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

f, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, figsize=(20, 10))
ax1.set_title('Citra Kontras')
ax1.imshow(color_image)
ax1.axis('off')

ax2.set_title('Merah')
ax2.imshow(r, cmap="gray")
ax2.axis('on')

ax3.set_title('Hijau')
ax3.imshow(g, cmap="gray")
ax3.axis('on')

ax4.set_title('Biru')
ax4.imshow(b, cmap="gray")
ax4.axis('on')

```

Codingan pada gambar tersebut digunakan untuk menampilkan gambar asli dan memisahkan gambar tersebut ke dalam tiga komponen warna utama RGB (Merah, Hijau, Biru), lalu menampilkannya dalam satu baris dengan empat subplot menggunakan library Python seperti OpenCV, NumPy, dan Matplotlib. Gambar di-load dari file dengan OpenCV, lalu masing-masing kanal warna (Red, Green, Blue) diambil secara terpisah. Setelah itu, gambar asli dan masing-masing kanal warna ditampilkan dalam satu figure berisi empat gambar: satu gambar asli berwarna penuh (color image), dan tiga gambar kanal warna dalam skala abu-abu untuk mempermudah visualisasi intensitas masing-masing warna. Tujuan program ini adalah untuk membantu memahami bagaimana setiap warna penyusun (R, G, B) berkontribusi terhadap gambar berwarna secara visual.

HISTOGRAM

Source code

```

[126]: b, g, r = cv2.split(img)

plt.figure(figsize=(20, 5))

plt.subplot(141)
plt.hist(r.ravel(), bins=256, color='red', range=[0,256], alpha=0.7)
plt.title('Histogram Merah')
plt.xlabel('Intensitas')
plt.ylabel('Jumlah Pikel')

plt.subplot(142)
plt.hist(g.ravel(), bins=256, color='green', range=[0,256], alpha=0.7)
plt.title('Histogram Hijau')
plt.xlabel('Intensitas')

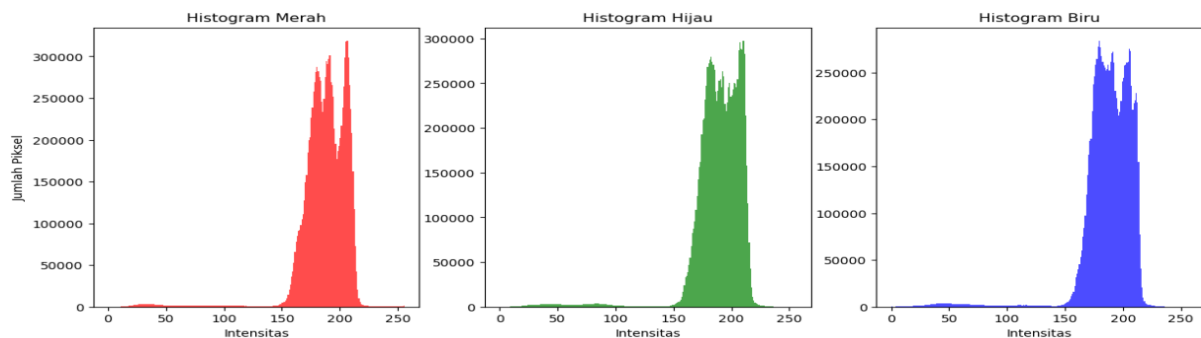
plt.subplot(143)
plt.hist(b.ravel(), bins=256, color='blue', range=[0,256], alpha=0.7)
plt.title('Histogram Biru')
plt.xlabel('Intensitas')

```

kode di atas digunakan untuk menampilkan histogram dari masing-masing kanal warna (Merah, Hijau, dan Biru) dari sebuah gambar berwarna. Histogram ini menggambarkan sebaran intensitas piksel untuk tiap warna, sehingga kita bisa tahu seberapa banyak piksel yang memiliki nilai terang atau gelap dalam setiap kanal. Proses dimulai dengan memisahkan gambar ke dalam tiga warna utama (R, G, B), lalu masing-masing warna dihitung distribusi nilainya dari 0 sampai 255 (intensitas piksel). Hasilnya ditampilkan dalam bentuk grafik batang (histogram) untuk setiap warna secara berdampingan. Ini berguna buat analisis visual, misalnya untuk cek apakah gambar terlalu gelap, terlalu terang, atau didominasi warna tertentu.

HASIL

```
[126]: Text(0.5, 0, 'Intensitas')
```



ANALISIS

Interpretasi histogram secara umum:

1. Distribusi Warna Merah, Hijau, dan Biru

Ketiga grafik histogram menunjukkan bahwa sebagian besar nilai intensitas warna berada pada rentang sekitar 150 hingga 230 untuk ketiga kanal (Merah, Hijau, dan Biru). Ini berarti bahwa gambar memiliki dominasi warna-warna yang terang (karena intensitas tinggi berarti lebih dekat ke putih).

2. Bentuk Histogram

Ketiga histogram menunjukkan pola puncak-puncak tajam (peaks), menandakan bahwa banyak piksel yang memiliki nilai intensitas spesifik tertentu, kemungkinan karena objek dalam gambar memiliki warna yang homogen atau berulang (seperti area besar dengan warna yang seragam).

3. Kesamaan Pola

Bentuk ketiga histogram sangat mirip dan selaras satu sama lain. Hal ini menunjukkan bahwa gambar cenderung berwarna netral atau seimbang, tanpa dominasi warna tertentu. Ini bisa terjadi misalnya jika gambar merupakan objek yang penuh warna atau hasil edit yang warnanya telah disesuaikan agar seimbang.

2. NILAI AMBANG BATAS CITRA

SOURCE CODE

```

IMPORT LIBRARY

[2]: import cv2
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as img

MEMBACA GAMBAR

[22]: img = cv2.imread('coloring.jpg')
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

UBAH KE HSV

[24]: img_hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)

AMBANG BATAS BIRU, MERAH, HIJAU

[26]: lower_blue = np.array([100, 150, 0])
upper_blue = np.array([140, 255, 255])
mask_blue = cv2.inRange(img_hsv, lower_blue, upper_blue)

lower_red1 = np.array([0, 120, 70])
upper_red1 = np.array([10, 255, 255])
mask_red1 = cv2.inRange(img_hsv, lower_red1, upper_red1)

lower_red2 = np.array([170, 120, 70])
upper_red2 = np.array([180, 255, 255])
mask_red2 = cv2.inRange(img_hsv, lower_red2, upper_red2)

mask_red = mask_red1 | mask_red2
mask_red_blue = mask_red | mask_blue

lower_green = np.array([40, 40, 40])
upper_green = np.array([90, 255, 255])
mask_green = cv2.inRange(img_hsv, lower_green, upper_green)

mask_rgb = mask_red | mask_green | mask_blue

MENAMPILKAN HASIL

fig, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, figsize=(20, 10))
ax1.imshow(np.zeros_like(mask_blue), cmap='gray')
ax1.set_title("NONE")
ax1.axis('off')

ax2.imshow(mask_blue, cmap='gray')
ax2.set_title("BLUE")
ax2.axis('off')

ax3.imshow(mask_red_blue, cmap='gray')
ax3.set_title("RED-BLUE")
ax3.axis('off')

ax4.imshow(mask_rgb, cmap='gray')
ax4.set_title("RED-GREEN-BLUE")
ax4.axis('off')

plt.tight_layout()
plt.show()

```

Kode di atas digunakan untuk mendeteksi dan menampilkan area tertentu pada gambar yang mengandung warna-warna dasar seperti merah, biru, dan hijau menggunakan pendekatan berbasis warna di ruang warna HSV. Prosesnya dimulai dari konversi gambar ke format HSV karena HSV lebih cocok untuk segmentasi warna dibanding RGB. Setelah itu, dibuat beberapa *masking*, yaitu filter untuk menangkap bagian-bagian gambar yang warnanya berada dalam rentang tertentu (misalnya biru terang atau merah pekat). Untuk warna merah, karena letaknya terpisah di spektrum HSV, perlu dua rentang sekaligus. Setelah masking untuk masing-masing warna dibuat, hasilnya digabungkan untuk menampilkan kombinasi warna (misalnya gabungan merah dan biru, atau semua warna utama RGB). Kemudian semua hasil tersebut divisualisasikan dalam satu jendela yang terdiri dari 4 gambar hitam-putih (mask), yang menunjukkan di mana warna-warna itu muncul pada gambar asli. Program ini digunakan untuk mengecek di mana letak warna-warna utama tertentu (merah, hijau, biru) dalam gambar, dan hasilnya divisualisasikan dengan sangat jelas dalam bentuk grayscale mask.

HASIL



ANALISIS:

- **NONE:**
Ini adalah gambar kosong (hitam seluruhnya), sebagai pembandingan untuk menunjukkan bahwa tidak ada warna yang disegmentasi. Bisa dibilang ini "baseline"-nya.
- **BLUE:**
Bagian ini hanya menampilkan teks yang memiliki warna biru pada gambar asli. Terlihat sebagian tulisan muncul, seperti "YO", "INE", dan "MBI", karena bagian teks tersebut terdeteksi berada dalam rentang warna biru.
- **RED-BLUE:**
Ini adalah gabungan antara mask merah dan biru. Lebih banyak teks muncul dibanding BLUE saja, karena sekarang teks berwarna merah juga disertakan, misalnya: "YO", "NDA", "STINE", "BR", "MBIRI". Ini menunjukkan bahwa beberapa huruf atau kata memang ditulis dengan warna merah atau biru.
- **RED-GREEN-BLUE:**
Ini adalah gabungan dari semua warna utama (merah, hijau, dan biru). Pada bagian ini, seluruh teks terlihat jelas dan utuh, menampilkan "YOLANDA KRISTINE BR SEMBIRING". Artinya, warna tulisan pada gambar asli memang menggunakan kombinasi RGB dan berhasil dideteksi semua.

3. MEMPERBAIKI GAMBAR BACKLIGHT

Source code

IMPORT LIBRARY

```
[24]: from PIL import Image, ImageEnhance
import matplotlib.pyplot as plt
import cv2
import numpy as np
import os
```

MEMBACA GAMBAR dan MEMBUKA GAMBAR ASLI

```
[27]: image_path = "backlight.jpg"

[33]: if not os.path.exists(image_path):
    print(f"❌ File tidak ditemukan: {image_path}")
else:
    original = Image.open(image_path)
```

KONVERSI KE GRAYSCALE

```
[37]: gray = original.convert("L")
```

Mencerahkan grayscale

```
[40]: gray_np = np.array(gray)
bright_gray_np = cv2.convertScaleAbs(gray_np, alpha=1.0, beta=50) # beta: tambah kecerahan
bright_gray = Image.fromarray(bright_gray_np)
```

MENINGKATKAN KONTRAS GRASYSSCALE

```
[43]: contrast_enhancer = ImageEnhance.Contrast(gray)
      contrast_gray = contrast_enhancer.enhance(2.0)
```

GABUNGAN CERAH DENGAN KONTRAS

```
[46]: bright_then_contrast_np = cv2.convertScaleAbs(gray_np, alpha=2.0, beta=50)
      bright_then_contrast = Image.fromarray(bright_then_contrast_np)
```

MENAMPILKAN HASIL

```
[54]: fig, axes = plt.subplots(1, 5, figsize=(20, 5))

      axes[0].imshow(original)
      axes[0].set_title("Gambar Asli")

      axes[1].imshow(gray, cmap='gray')
      axes[1].set_title("Grayscale")

      axes[2].imshow(bright_gray, cmap='gray')
      axes[2].set_title("Grayscale Dicerahkan")

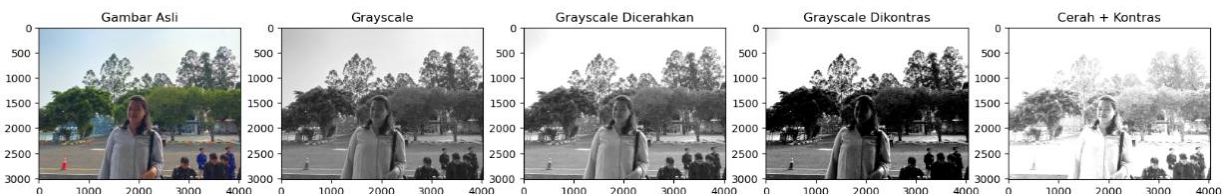
      axes[3].imshow(contrast_gray, cmap='gray')
      axes[3].set_title("Grayscale Dikontras")

      axes[4].imshow(bright_then_contrast, cmap='gray')
      axes[4].set_title("Cerah + Kontras")
```

Kode di atas adalah program untuk memperbaiki kualitas gambar yang kurang pencahayaan (backlight) dengan mengubah tampilannya ke mode grayscale, lalu meningkatkan kecerahan dan kontrasnya. Proses ini bertujuan agar detail dalam gambar yang tadinya gelap bisa terlihat lebih jelas. Secara garis besar, program ini pertama-tama mengecek apakah gambar yang dimaksud tersedia. Kalau ada, gambar dibuka dan dikonversi ke grayscale agar lebih fokus ke intensitas cahaya, bukan warna. Setelah itu, ada beberapa metode yang diterapkan: menambah kecerahan, memperbesar kontras, dan gabungan keduanya. Semua hasilnya lalu ditampilkan secara berdampingan dalam satu tampilan agar bisa dibandingkan secara visual, mana yang paling efektif meningkatkan visibilitas gambar.

HASIL:

```
[54]: Text(0.5, 1.0, 'Cerah + Kontras')
```



ANALISIS:

1. Gambar Asli: Terlihat bahwa objek utama tampak gelap karena sumber cahaya dari belakangnya sangat kuat. Ini adalah masalah umum pada kondisi backlight.
2. Grayscale: Gambar dikonversi ke hitam putih. Di sini kita bisa melihat bahwa meskipun warna dihilangkan, masalah pencahayaan masih sama, objek utama tetap kurang terlihat jelas.
3. Grayscale Dicerahkan: Gambar grayscale ditingkatkan kecerahannya. Hasilnya, wajah dan tubuh orang jadi lebih kelihatan, tapi detail latar belakang juga jadi makin terang, yang kadang bisa membuat hasilnya terlalu "pucat".
4. Grayscale Dikontras: Fokusnya pada meningkatkan perbedaan terang-gelap. Bagian yang gelap jadi lebih gelap, yang terang jadi lebih terang. Ini bikin siluet orang jadi lebih jelas, tapi beberapa bagian bisa terlihat terlalu pekat.
5. Cerah + Kontras: Ini gabungan dari dua metode sebelumnya. Objek utama jadi paling kelihatan jelas, dengan latar belakang yang tetap terang. Bisa dibilang ini hasil terbaik untuk memperjelas objek tanpa mengorbankan terlalu banyak detail.

BAB IV

PENUTUP

Kesimpulan

- **Pengolahan citra digital** merupakan teknik penting dalam memanipulasi dan menganalisis gambar menggunakan komputer. Dengan memahami konsep dasar seperti citra RGB, grayscale, biner, serta histogram, kita dapat mengevaluasi dan meningkatkan kualitas gambar sesuai kebutuhan.
- **Pemisahan kanal warna RGB** memungkinkan kita untuk mengetahui kontribusi masing-masing warna terhadap gambar keseluruhan. Setiap kanal (merah, hijau, biru) dapat memperlihatkan bagian-bagian tertentu dari gambar dengan lebih jelas, tergantung pada dominasi warnanya.
- **Histogram warna** sangat berguna untuk menganalisis sebaran intensitas piksel dalam gambar. Dari histogram, dapat diketahui apakah gambar memiliki pencahayaan yang cukup, terlalu terang, terlalu gelap, atau seimbang.
- **Segmentasi warna dengan HSV** terbukti efektif dalam mendeteksi bagian-bagian tertentu dari gambar yang memiliki warna spesifik. Teknik thresholding berdasarkan rentang HSV sangat membantu untuk mengekstrak informasi berbasis warna secara akurat.
- **Perbaikan citra backlight** dapat dilakukan melalui peningkatan kecerahan dan kontras pada gambar grayscale. Kombinasi kedua teknik tersebut mampu membuat objek utama yang awalnya gelap menjadi lebih jelas dan mudah dikenali.

DAFTAR PUSTAKA

Dr. Arnita, S. M. (2022). COMPUTER VISION DAN PENGOLAHAN. *PUSTAKA AKSARA*.

Irene Devi Damayanti, A. M. (Juli 2023). Klasifikasi Citra Daging Babi dan Daging Kerbau Menggunakan Histogram Citra dan GLCM. *Journal of System and Computer Engineering (JSCE)*.

nadzir Zaid Munantri, H. S. (2021). APLIKASI PENGOLAHAN CITRA DIGITAL UNTUK IDENTIFIKASI. *JURNAL TEKNIK INFORMATIKA*.