

UTS

PENGOLAHAN CITRA DIGITAL



NAMA : Margaret Puji Lestari Saragih

NIM : 202331176

KELAS : E

DOSEN : Dr. Darma Rusjdi, S.T., M.Kom

NO.PC : 13

ASISTEN : 1. Fuazan Arroyan

2. Abdur Rasyid Ridho

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.....	4
BAB II LANDASAN TEORI.....	5
2.1 Citra	5
2.2 Pengolahan Citra	5
2.3 Warna	5
2.4 Citra Keabuan	5
2.5 Peta Keabuan	6
2.6 Pengolahan Citra Digital.....	6
2.7 Citra Warna	7
2..8 Citra Grayscale.....	7
2.9 Citra Biner	8
BAB III HASIL.....	9
BAB IV PENUTUP.....	16
DAFTAR PUSTAKA	17

BAB I

PENDAHULUAN

1.1 Rumusan Masalah

Dalam praktikum UTS Pengolahan Citra Digital ini, mahasiswa diminta untuk memproses sebuah citra hasil potret pribadi yang memuat tulisan tangan dengan tinta warna merah, hijau, dan biru. Berdasarkan ketentuan dan eksperimen yang dilakukan, rumusan masalah dari praktikum ini dapat dijabarkan sebagai berikut:

1. Bagaimana cara mendeteksi warna merah, hijau, dan biru secara terpisah dari sebuah citra menggunakan pemrograman Python?
2. Bagaimana cara menampilkan hasil histogram dari masing-masing kanal warna untuk mengetahui distribusi intensitas piksel pada citra?
3. Bagaimana menentukan nilai ambang batas (threshold) warna yang optimal untuk memisahkan masing-masing warna secara akurat?
4. Bagaimana memperbaiki kualitas gambar backlight agar objek utama (misalnya wajah) menjadi lebih jelas dan menonjol dibandingkan latar belakang terang?

Rumusan masalah ini muncul karena tantangan dalam pemisahan warna, ketepatan dalam memilih nilai threshold, serta teknik yang digunakan untuk peningkatan kualitas gambar backlight agar informasi visual tetap jelas dan tidak overexposed

1.2 Tujuan Masalah

Tujuan dari praktikum ini secara umum adalah untuk mengaplikasikan teori-teori pengolahan citra yang telah dipelajari selama perkuliahan ke dalam bentuk implementasi nyata menggunakan bahasa pemrograman Python dan pustaka OpenCV. Secara khusus, tujuan dari UTS ini meliputi:

1. Mampu memisahkan kanal warna merah, hijau, dan biru dari citra RGB dengan menggunakan teknik ekstraksi kanal.
2. Menganalisis intensitas warna dengan menggunakan histogram untuk melihat distribusi piksel terhadap tiap kanal warna.
3. Menentukan nilai ambang batas terkecil hingga terbesar guna melakukan segmentasi warna yang lebih akurat pada citra.
4. Memperbaiki kualitas citra backlight dengan cara meningkatkan kontras dan kecerahan pada bagian yang gelap agar objek utama lebih terlihat jelas.
5. Meningkatkan pemahaman mahasiswa terhadap konsep dasar pengolahan citra digital seperti konversi warna, thresholding, histogram, serta teknik preprocessing gambar.

1.3 Manfaat Masalah

Melalui praktikum UTS ini, terdapat beberapa manfaat penting yang diperoleh, antara lain:

1. Penerapan Praktis Ilmu – Mahasiswa dapat mengaplikasikan ilmu pengolahan citra digital dalam bentuk praktik langsung melalui coding dan pengolahan data citra yang nyata.
2. Kemampuan Analisis Citra – Mahasiswa menjadi terbiasa menganalisis dan memahami struktur warna dalam sebuah gambar serta bagaimana setiap kanal warna berkontribusi terhadap informasi visual.
3. Pemahaman Teknik Thresholding – Mahasiswa belajar pentingnya memilih ambang batas yang tepat dalam proses segmentasi objek berdasarkan warna, yang merupakan dasar dalam machine vision.
4. Perbaikan Citra Nyata – Praktikum memberikan pemahaman bagaimana memperbaiki foto dengan kondisi pencahayaan yang tidak ideal (backlight), yang sering terjadi dalam kehidupan sehari-hari.
5. Keterampilan Pemrograman – Mahasiswa terasah keterampilannya dalam menggunakan Python, terutama pustaka OpenCV yang merupakan standar dalam industri pemrosesan citra.
6. Persiapan Dunia Kerja – Pengetahuan ini sangat relevan untuk dipakai di dunia kerja, terutama di bidang computer vision, teknologi kamera, augmented reality, hingga sistem pengenalan visual berbasis AI.

BAB II

LANDASAN TEORI

2.1 Citra

Citra adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus dan intensitas Cahaya pada bidang dwimatra (Munir, 2020)

2.2 Pengolahan Citra

Pengolahan Citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra yang berkualitas lebih baik daripada citra masukan (Munir, 2004).

2.3 Warna

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu mempunyai panjang gelombang paling rendah

Warna-warna yang diterima oleh mata merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Kombinasi warna yang memberikan rentang warna yang paling lebar adalah red(R), green(G) dan blue(B) dan warna bukan merupakan besaran fisik, tetapi warna merupakan suatu sensasi yang dihubungkan dengan sistem saraf kita, seperti halnya rasa maupun bau. Sensasi warna diperoleh dengan adanya interaksi antara warna dengan sistem saraf semutif warna kita

2.4 Citra Keabuan

Citra beratus keabuan adalah citra yang hanya menggunakan warna yang merupakan tingkatan warna abu-abu. Warna abu-abu adalah satu-satunya warna pada ruang RGB dengan komponen merah, hijau, dan biru mempunyai intensitas yang sama. Pada citra beratus keabuan hanya perlu menyatakan nilai intensitas untuk tiap pixel sebagai nilai tunggal, sedangkan pada citra berwarna perlu tiga nilai intensitas untuk tiap pixel-nya.

Intensitas citra, beratus keabuan disimpan sebagai integer 8 bit sehingga memberikan $2^8 = 256$ tingkat keabuan dari warna hitam sampai warna putih. Dengan menggunakan pola ini citra beratus keabuan membutuhkan ruang memori, disk, dan waktu pengolahan yang lebih sedikit daripada citra berwarna

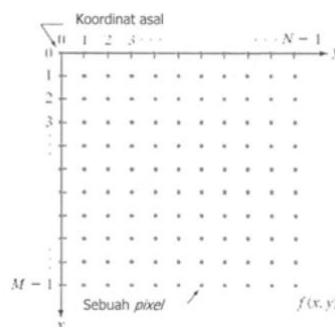
2.5 Peta Keabuan

pada suatu pixel keluaran terhadap intensitas pixel masukan. Pixel dengan intensitas terendah adalah hitam, dan pixel dengan intensitas tertinggi adalah putih. Sebuah pixel dengan intensitas sedang mungkin berwarna abu-abu atau memiliki tingkat keabuan tertentu.

2.6 Pengolahan Citra Digital

Citra atau Gambar (Image) adalah sebuah informasi yang memiliki bentuk visual. Suatu citra yang diperoleh dari hasil kamera dan dapat diolah menggunakan komputer atau laptop merupakan citra digital, dapat juga diartikan bahwa citra adalah kumpulan dari piksel-piksel yang disusun dalam dua dimensi. Piksel merupakan elemen terkecil yang menyusun citra pada suatu titik tertentu mengandung nilai yang mewakili kecerahan dari sebuah warna, Umumnya Citra digital berbentuk persegi panjang yang memiliki lebar dan tingi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya piksel sehingga ukuran citra selalu bernilai bulat.

Dalam komputer, setiap piksel diwakili oleh dua buah bilangan bulat (integer) untuk menunjukkan lokasinya dalam bidang citra dan sebuah nilai dalam bilangan bulat untuk menunjukkan cahaya atau terang-gelapnya piksel tersebut. URAB Untuk menunjukkan lokasi pada suatu piksel, koordinat (0,0) digunakan sebagai posisi kiri atas dalam bidang citra, dan koordinat (m-1,n-1) digunakan sebagai posisi kanan bawah dalam citra berukuran (mxn) piksel, dapat dilihat pada gambar dibawah



Gambar Koordinat Citra Digital

Sedangkan untuk menunjukkan tingkat pencahayaan suatu piksel, seringkali menggunakan bilangan bulat yang besarnya 8-bit, dengan lebar selang nilai 0 hingga 255, di mana nilai 0 merupakan warna hitam, nilai 255 merupakan warna putih dan tingkat abu-abu berada diantara nilai-nilai 0 dan 255

Pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya digunakan untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Secara umum, istilah pengolahan citra digital merupakan pemrosesan gambar berdua dimensi melalui komputer digital (Efford, 2000). Suatu citra dapat juga didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y yang merupakan koordinat spasial, dan amplitudo / pada titik koordinat (x,y) dinamakan sebagai intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai dari x, y, dan nilai dari amplitudo / bernilai diskrit, maka dapat dikatakan bahwa citra tersebut merupakan citra digital,

Matrik pada citra digital dapat dituliskan sebagai berikut:

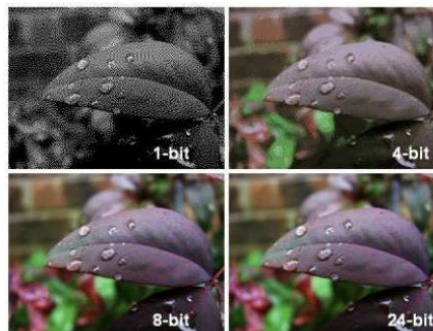
$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \dots\dots\dots (1)$$

Nilai pada suatu irisan antara baris dengan kolom (pada posisi x, y) disebut dengan picture elements, image elements, atau pixels. Istilah terakhir (pixel) merupakan yang paling sering digunakan pada citra digital. Nilai pada suatu pixel memiliki nilai rentang tertentu, dari nilai minimumnya hingga nilai maksimumnya. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0 hingga 255. Citra dengan penggambaran T dalam seperti ini tergolong ke dalam citra integer. Berikut adalah jenis-jenis citra berdasarkan nilai pixelnya

2.7 Citra Warna

Setiap pixel dari citra warna 24 bit diwakili dengan 24 bit sehingga memiliki total 16.777.216 variasi warna. Variasi warna ini untuk memvisualisasikan seluruh Warna yang dapat dilihat oleh penglihatan manusia sudah lebih dari cukup. Karena penglihatan manusia hanya dapat membedakan hingga 10 juta warna.

Setiap poin informasi yang dimiliki pixel (RGB) disimpan ke dalam 1 byte data. Pada 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai Hijau ORAB pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah



Gambar Citra Warna

2.8 Citra Grayscale

Menurut Basuki, Proses yang banyak dilakukan dalam image processing pada proses awal adalah merubah citra berwarna menjadi citra grayscale. Hal ini bertujuan untuk menyederhanakan model citra. Terdapat 3 layer matrix pada citra berwarna, yaitu Red-layer, Green-layer dan Blue-layer maka untuk melakukan proses-proses berikutnya tetap diperhatikan pada tiga layer diatas. Bila setiap proses perhitungan dilakukan menggunakan tiga layer, maka dilakukan tiga perhitungan yang sama. Sehingga konsep itu dirubah dengan mengubah tiga layer diatas menjadi 1 layer matrix grayscale dan akan menghasilkan citra grayscale. Dalam citra grayscale tidak ada lagi warna, yang

ada adalah derajat keabuabuan. Untuk mengubah citra berwarna yang mempunyai nilai matrix masing-masing red, green, dan blue menjadi citra grayscale dengan tilai S . maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai red, green, dan blue.

2.9 Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai posel yaitu hitam dan putih, seperti pada gambar 2.4 Citra Biner, Cira B&W (black and white) atau citra monokrom disebut juga sebagai Citra biner. Hanya membutuhkan. 1 bit untuk dapat mewakili nilai setiap pixel dari citra biner. Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti pengambangan (thresholding)



Operasi pengambangan (thresholding) mengelompokkan nilai derajat kesbuannya pada setiap putel ke dalam 2 kelas, hitam dan putih. Dua pendekatan yang digunakan dalam melakukan operasi pengambangan adalah erasi pengambangan adalah pengambangan secara global dan pengambangan secara lokal. Dua pendekatan ini memiliki kelebihan dan kekurangan salah satunya adalah pada pengambangan secara local prosesnya lebih lambat bila dibandingkan dengan pengembangan secara global. Ada banyam algoritma yang digunakan untuk citra biner, salah satu diantaranya adalah menggunakan metode threshold otsu

BAB III

HASIL

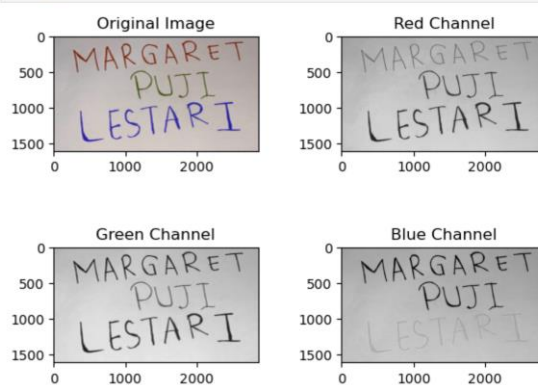
```
1]: plt.subplot(2, 2, 1)
plt.imshow(cv2.cvtColor(rgb, cv2.COLOR_BGR2RGB))
plt.title('Original Image')

plt.subplot(2, 2, 2)
plt.imshow(cv2.cvtColor(rgb, cv2.COLOR_BGR2RGB)[: , :, 0], cmap="gray")
plt.title('Red Channel')

plt.subplot(2, 2, 3)
plt.imshow(cv2.cvtColor(rgb, cv2.COLOR_BGR2RGB)[: , :, 1], cmap="gray")
plt.title('Green Channel')

plt.subplot(2, 2, 4)
plt.imshow(cv2.cvtColor(rgb, cv2.COLOR_BGR2RGB)[: , :, 2], cmap="gray")
plt.title('Blue Channel')

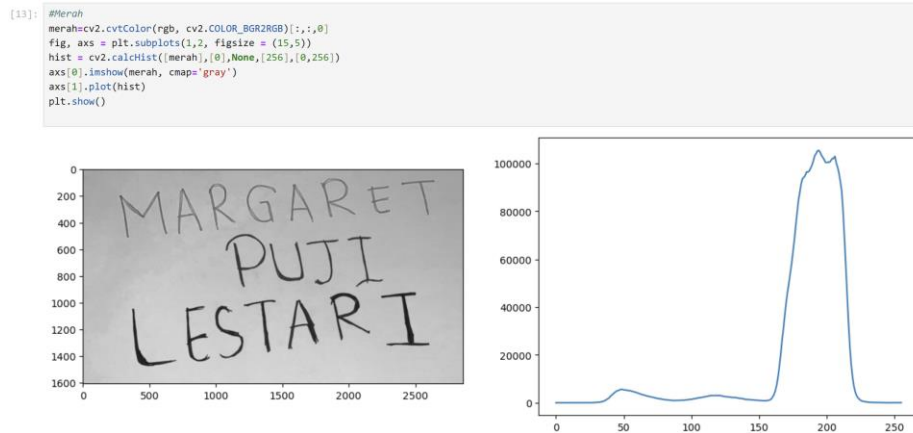
plt.tight_layout(pad=3.0) # Tambahkan ini untuk memberi jarak antar subplot
plt.show()
```



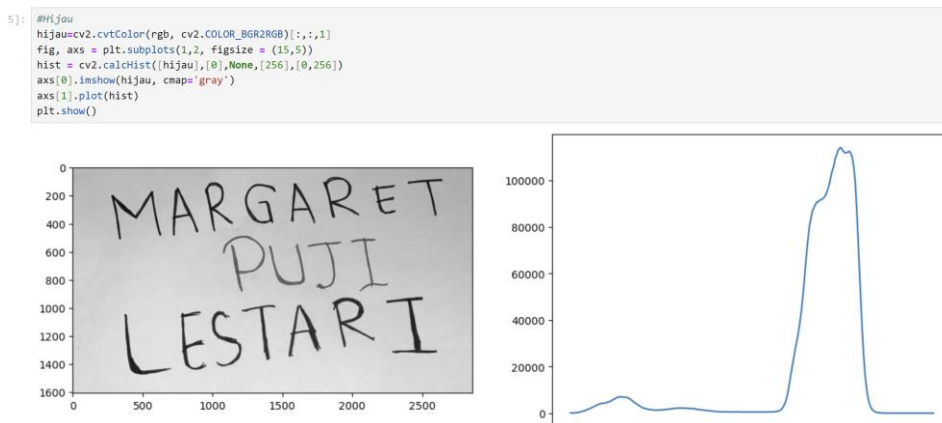
Pada praktikum ini, saya menjalankan kode Python yang tujuannya untuk menampilkan gambar asli dan memecah gambar tersebut ke dalam tiga channel warna utama, yaitu Red (merah), Green (hijau), dan Blue (biru). Pertama-tama, gambar asli yang ditampilkan adalah gambar bertuliskan “MARGARET PUJI LESTARI” dengan warna tulisan yang berbeda-beda—kata “MARGARET” ditulis dengan warna merah, “PUJI” dengan hijau, dan “LESTARI” dengan biru. Untuk bisa menampilkan warnanya dengan benar, gambar diubah dulu dari format BGR (yang biasanya digunakan di OpenCV) ke RGB.

Setelah gambar aslinya ditampilkan, langkah berikutnya adalah menampilkan masing-masing channel warna secara terpisah. Di bagian Red Channel, kita bisa lihat kalau tulisan “MARGARET” terlihat paling terang, itu karena warna merah memang paling dominan di bagian tersebut. Lalu di Green Channel, tulisan “PUJI” jadi yang paling kelihatan terang, karena memang warnanya hijau. Dan di Blue Channel, bagian tulisan “LESTARI” yang paling menonjol karena ditulis dengan warna biru.

Dari hasil visualisasi ini, saya jadi bisa lebih paham gimana sebuah gambar digital itu tersusun dari tiga channel warna utama. Setiap channel ini membawa informasi intensitas warna masing-masing, dan kita bisa manfaatin informasi ini buat analisis gambar lebih lanjut, misalnya untuk deteksi warna tertentu, segmentasi objek, atau pengolahan citra lainnya. Praktikum ini cukup menarik karena membantu saya memahami konsep dasar RGB dalam pengolahan citra digital secara langsung lewat visual.



Pada gambar di atas, dilakukan proses analisis citra menggunakan Python dan OpenCV untuk mengambil kanal merah (Red Channel) dari sebuah gambar berformat RGB. Proses ini dimulai dengan mengubah urutan warna dari BGR ke RGB menggunakan fungsi `cv2.cvtColor`, lalu diambil kanal merahnya dengan `[:, :, 0]`. Setelah itu, histogram dari kanal merah dihitung menggunakan `cv2.calcHist`, yang berfungsi untuk mengetahui sebaran intensitas warna merah dari 0 sampai 255. Gambar sebelah kiri menunjukkan tampilan grayscale dari kanal merah, sedangkan grafik di sebelah kanan adalah histogram yang menunjukkan jumlah piksel untuk setiap level intensitas merah. Dari hasil histogram, tampak bahwa sebagian besar piksel berada pada rentang intensitas tinggi (sekitar nilai 200), yang menunjukkan bahwa gambar memiliki dominasi warna merah terang. Ini bisa terjadi karena latar belakang gambar lebih terang dibandingkan dengan tulisan yang berwarna gelap. Analisis semacam ini penting dalam pemrosesan citra digital, terutama untuk segmentasi objek, deteksi tepi, atau peningkatan kualitas gambar.

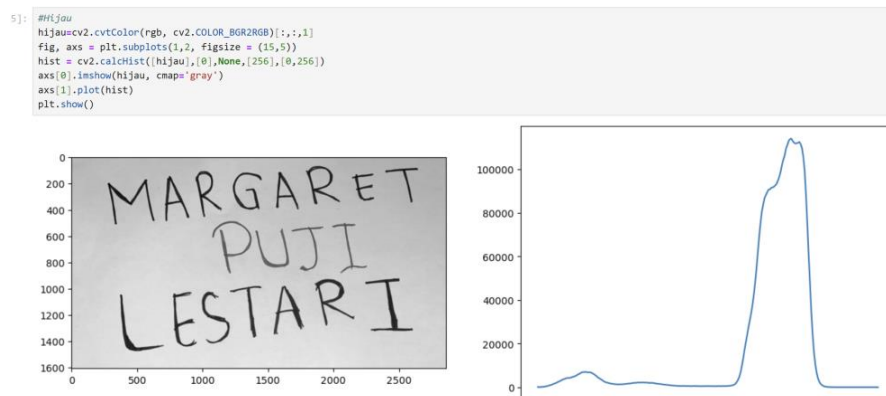


Pada bagian praktikum ini, saya memfokuskan analisis pada channel warna hijau (green channel) dari gambar yang sebelumnya digunakan. Kode yang dijalankan bertujuan untuk menampilkan citra hasil ekstraksi channel hijau serta menampilkan histogram dari channel tersebut. Pertama-tama, gambar diubah dulu dari format BGR ke RGB, kemudian dipilih channel ke-1 (karena indeks 0 untuk merah, 1 untuk hijau, dan 2 untuk biru).

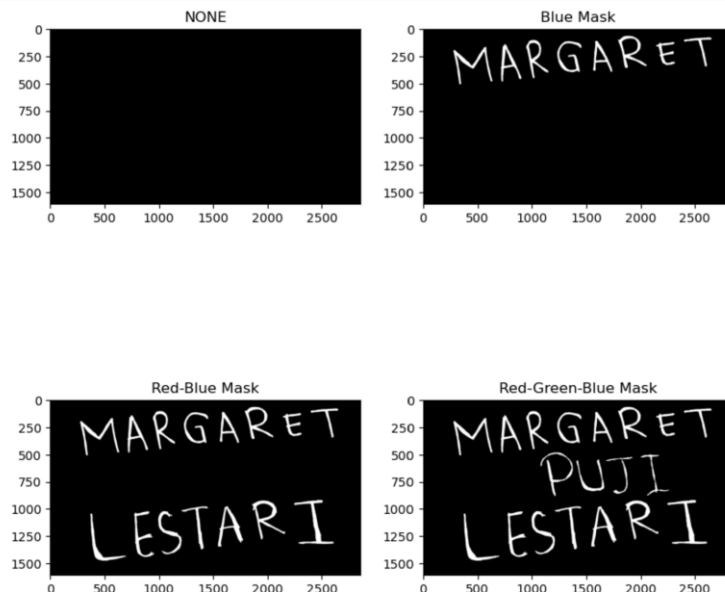
Gambar di sebelah kiri menunjukkan hasil channel hijau dalam bentuk grayscale, di mana tingkat kecerahan pada gambar menunjukkan intensitas warna hijau di setiap piksel. Semakin terang suatu bagian, berarti semakin tinggi kandungan warna hijaunya. Dari hasil tersebut, bisa terlihat bahwa bagian tulisan "PUJI" terlihat lebih terang dibandingkan bagian lainnya, karena memang huruf tersebut ditulis dengan warna hijau.

Sementara itu, grafik di sebelah kanan adalah histogram dari channel hijau. Histogram ini menggambarkan distribusi intensitas piksel hijau dalam gambar dari nilai 0 (gelap) sampai 255 (terang). Dari grafiknya terlihat ada puncak di area intensitas tinggi, yang menandakan bahwa sebagian besar piksel di gambar ini memiliki tingkat kecerahan hijau yang cukup terang. Hal ini sesuai dengan gambar aslinya yang latarnya cukup terang dan hanya sebagian kecil (pada huruf-huruf) yang berwarna lebih gelap.

Melalui praktikum ini, saya jadi lebih memahami bagaimana citra digital bisa dipecah ke dalam channel warna dan bagaimana histogram bisa memberikan informasi kuantitatif tentang distribusi intensitas warna. Analisis seperti ini penting dalam pengolahan citra, apalagi kalau kita ingin melakukan segmentasi objek, peningkatan kualitas gambar, atau pendeteksian objek berdasarkan warna tertentu.



Gambar di atas menunjukkan hasil analisis kanal biru dari citra menggunakan OpenCV. Proses dimulai dengan mengubah format warna gambar dari BGR ke RGB, lalu mengambil kanal birunya menggunakan `[:, :, 2]`. Gambar grayscale di sebelah kiri adalah representasi dari intensitas warna biru di setiap piksel. Dari citra tersebut bisa dilihat bahwa tulisan "LESTARI" tampak lebih terang dibanding huruf lainnya, menunjukkan bahwa tulisan tersebut mengandung lebih banyak komponen biru. Sedangkan grafik di sebelah kanan adalah histogram yang menunjukkan distribusi intensitas warna biru dari 0 sampai 255. Histogram memperlihatkan bahwa mayoritas piksel berada di kisaran intensitas tinggi (sekitar 180-200), menandakan latar belakang gambar memiliki dominasi warna biru terang. Sementara itu, terdapat juga lonjakan kecil di nilai intensitas rendah (sekitar 0–30), yang berasal dari bagian huruf yang gelap karena minim komponen biru. Analisis seperti ini berguna untuk memahami komposisi warna dalam citra, terutama dalam aplikasi pemrosesan citra digital seperti deteksi objek berbasis warna atau segmentasi.



Gambar dan kode di atas menunjukkan proses segmentasi warna pada citra menggunakan ruang warna HSV untuk mendeteksi tiga warna utama: merah, hijau, dan biru. Pertama, gambar diubah ke grayscale dan dilakukan thresholding sehingga menghasilkan citra biner sebagai pembanding tanpa segmentasi warna. Selanjutnya, dibuat mask untuk warna biru saja, lalu digabungkan dengan mask merah untuk menghasilkan mask kombinasi merah-biru. Setelah itu, ditambahkan juga mask hijau sehingga menghasilkan kombinasi dari ketiga warna utama. Hasil akhirnya adalah citra yang menunjukkan area-area yang mengandung warna mencolok (merah, hijau, biru), yang berguna dalam proses analisis citra seperti pelacakan objek atau klasifikasi berdasarkan warna.

```
[45]: img2 = cv2.imread("gambar3.jpg")
      rgb2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
      plt.imshow(rgb2)

[45]: <matplotlib.image.AxesImage at 0x21d018c5b88>
```

Perintah tersebut digunakan untuk menampilkan gambar kedua yang disimpan dengan nama "gambar2.jpg". Gambar awalnya dibaca dalam format BGR (default-nya OpenCV), lalu diubah ke format RGB agar warnanya bisa ditampilkan dengan benar di Matplotlib. Setelah dikonversi, gambar ditampilkan menggunakan plt.imshow(). Hasil akhirnya adalah tampilan visual dari gambar tersebut dengan warna yang sesuai aslinya, bukan kebalik seperti kalau langsung ditampilkan tanpa konversi warna. Proses ini penting kalau kita ingin melakukan analisis citra dengan tampilan warna yang akurat.

```
[49]: gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

plt.imshow(gray, cmap='gray')
plt.title("Gambar Grayscale")
plt.axis("off")
plt.show()
```

Gambar Grayscale



Perintah tersebut digunakan untuk mengubah gambar berwarna menjadi gambar grayscale atau hitam putih. Gambar yang awalnya dalam format BGR diubah ke grayscale menggunakan OpenCV. Setelah itu, gambar grayscale ditampilkan dengan `plt.imshow()` menggunakan colormap abu-abu (`cmap='gray'`) agar tampak sesuai. Hasil akhirnya adalah citra tanpa warna yang hanya menunjukkan intensitas terang-gelap, di mana nilai piksel rendah terlihat gelap dan nilai piksel tinggi terlihat terang. Konversi ini penting sebagai tahap awal dalam banyak proses pengolahan citra seperti deteksi tepi, segmentasi, atau thresholding.

```
[51]: bright = cv2.convertScaleAbs(gray, alpha=1, beta=50)

plt.imshow(bright, cmap='gray')
plt.title("Grayscale yang Dicerahkan")
plt.axis("off")
plt.show()
```

Grayscale yang Dicerahkan



Perintah di atas digunakan untuk mencerahkan gambar grayscale. Proses ini dilakukan dengan menambahkan nilai konstan ($\beta = 50$) ke setiap piksel pada gambar, sementara nilai $\alpha = 1$ berarti tidak ada perubahan kontras. Hasilnya adalah gambar yang tampak lebih terang karena seluruh intensitas pikselnya dinaikkan. Gambar yang sudah dicerahkan kemudian ditampilkan menggunakan colormap abu-abu. Teknik seperti ini berguna untuk meningkatkan visibilitas detail pada gambar yang awalnya terlalu gelap, terutama saat melakukan analisis visual atau tahap pra-pemrosesan sebelum pengolahan citra lebih lanjut.

```
[53]: contrast = cv2.convertScaleAbs(gray, alpha=2.0, beta=0)
plt.imshow(contrast, cmap='gray')
plt.title("Grayscale yang Diperkontras")
plt.axis("off")
plt.show()
```

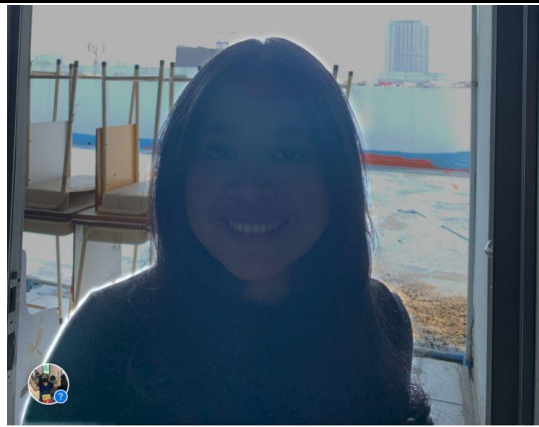


Perintah ini digunakan untuk meningkatkan kontras pada gambar grayscale. Nilai $\alpha = 2.0$ berfungsi untuk mengalikan setiap nilai piksel, sehingga perbedaan antara piksel terang dan gelap jadi lebih mencolok. Sementara $\beta = 0$ artinya tidak ada penambahan kecerahan. Hasil akhirnya adalah gambar dengan tampilan yang lebih tajam—bagian terang jadi lebih terang dan bagian gelap jadi lebih gelap. Ini bermanfaat untuk menonjolkan detail pada gambar, terutama saat gambar aslinya tampak datar atau kurang jelas. Peningkatan kontras sering digunakan dalam tahap awal pengolahan citra agar fitur penting lebih mudah dikenali.

```
[55]: bright_contrast = cv2.convertScaleAbs(bright, alpha=2.0, beta=0)
plt.imshow(bright_contrast, cmap='gray')
plt.title("Grayscale Dicerah + Kontras")
plt.axis("off")
plt.show()
```



Perintah ini digunakan untuk menerapkan dua proses sekaligus pada gambar grayscale: pencerahan dan peningkatan kontras. Pertama, gambar sudah dicerahkan sebelumnya dengan menambahkan nilai β (jadi tampil lebih terang), lalu hasilnya diproses lagi dengan $\alpha = 2.0$ untuk memperbesar kontras. Artinya, gambar awal yang sudah terang kemudian ditajamkan lagi perbedaan gelap-terangnya. Hasil akhirnya adalah gambar yang tampak lebih cerah dan lebih tajam, dengan detail yang lebih menonjol. Proses seperti ini cocok digunakan untuk menonjolkan objek atau teks dalam gambar yang sebelumnya tampak kusam atau tidak terlalu jelas.



Tambah Teks

Kamis, 08 Mei 2025 pukul 13.52

Sesuaikan

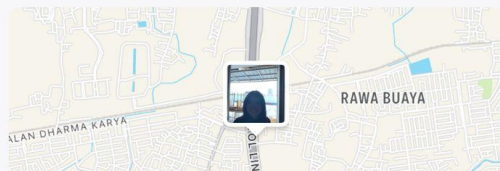
IMG_5743

Apple iPhone XR

HEIF

Kamera Belakang — 26 mm f1.8
12 MP • 3024 × 4032 • 1,4 MB

ISO 25 | 26 mm | 0 ev | f1.8 | 1/577 s



Kota Administrasi...arat - Rawa Buaya > Sesuaikan



Add a Caption

Wednesday • 7 May 2025 • 20.38

Adjust

IMG_5210

Apple iPhone 13

HEIF



Wide Camera — 26 mm f1.6
4 MP • 2858 × 1608 • 3,5 MB

COOL

ISO 32 | 26 mm | 0 ev | f1.6 | 1/75 s



BAB IV

PENUTUP

Setelah saya melakukan seluruh rangkaian praktikum UTS Pengolahan Citra Digital ini, saya jadi lebih memahami bagaimana gambar digital itu bisa diolah menggunakan bahasa pemrograman Python dan library OpenCV. Melalui proses ekstraksi warna, saya bisa melihat bahwa setiap warna (merah, hijau, dan biru) memiliki kanal tersendiri yang bisa dianalisis secara terpisah. Dari situ, saya bisa menampilkan histogram yang menunjukkan sebaran intensitas masing-masing warna dalam gambar. Analisis ini membuka wawasan saya tentang pentingnya mengenali struktur warna di balik gambar digital.

Selain itu, saya juga belajar bagaimana mencari ambang batas (threshold) yang tepat untuk membedakan bagian gambar berdasarkan warna, serta bagaimana memperbaiki gambar backlight dengan meningkatkan kontras dan kecerahannya. Proses tersebut sangat membantu dalam situasi nyata di mana gambar sering kali tidak diambil dalam kondisi pencahayaan yang ideal.

Secara keseluruhan, praktikum ini menurut saya cukup menantang tapi juga seru karena tidak cuma belajar teori, tapi juga langsung praktik dan melihat hasilnya. Dengan praktik ini, saya jadi paham bahwa pengolahan citra bukan cuma sekadar ubah-ubah gambar, tapi juga bisa sangat berguna untuk mendeteksi objek, meningkatkan kualitas visual, sampai jadi dasar dalam pengembangan sistem cerdas berbasis komputer.

Semoga ilmu yang saya pelajari di praktikum ini bisa terus dikembangkan di mata kuliah selanjutnya, bahkan mungkin juga bisa diterapkan di dunia kerja nanti, terutama di bidang teknologi visual dan komputerisasi gambar

DAFTAR PUSTAKA

https://repository.dinamika.ac.id/id/eprint/2029/4/BAB_II.pdf

https://repository.dinamika.ac.id/id/eprint/2610/4/BAB_II.pdf

<https://repository.uin-suska.ac.id/3829/3/BAB%20II%20LANDASAN%20TEORI.pdf>