

UTS

PENGOLAHAN CITRA



NAMA : Nadila Kusuma Dewi

NIM : 202331035

KELAS : B

DOSEN : Ir. Darma Rusjdi., M.Kom

NO.PC : 17

ASISTEN : 1. Davina Najwa Ermawan
2. Fakhrol Fauzi Nugraha Tarigan
3. Viana Salsabila Fairuz Syahla
4. Muhammad Hanief Febriansyah

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2024/2025

DAFTAR ISI

DAFTAR ISI.....	2
BAB I.....	3
PENDAHULUAN	3
1.1 Rumusan Masalah	3
1.2 Tujuan Masalah	3
1.3 Manfaat Masalah	3
BAB II.....	5
LANDASAN TEORI.....	5
2.1 Definisi Citra Digital.....	5
2.2 Model Warna RGB.....	5
2.3 Deteksi Warna.....	6
2.4 Konsep Dasar Citra	6
2.5 Analisis Histogram Citra	7
2.6 Segmentasi Citra.....	7
2.7 Peningkatan Kualitas Citra.....	8
2.8 Konversi Model Warna.....	8
BAB III	10
HASIL.....	10
3.1 Deteksi Warna Pada Citra	10
3.2 Mencari dan mengurutkan ambang batas terkecil sampai dengan terbesar	12
3.3 Memperbaiki Gambar Backlight	13
BAB IV	17
PENUTUP.....	17
4.1 Kesimpulan	17
DAFTAR PUSTAKA	19

BAB I

PENDAHULUAN

1.1 Rumusan Masalah

Dalam pelaksanaan ujian praktikum pengolahan citra digital ini, mahasiswa dihadapkan pada tantangan untuk mengaplikasikan berbagai teknik pengolahan citra yang telah dipelajari selama perkuliahan. Setelah menganalisis soal-soal yang diberikan dalam ujian praktikum, dapat diidentifikasi beberapa permasalahan utama yang perlu diselesaikan sebagai berikut:

1. Bagaimana mengimplementasikan metode deteksi warna (merah, biru, dan hijau) pada citra digital secara efektif?
2. Bagaimana menentukan nilai ambang batas yang optimal untuk kategorisasi warna pada citra?
3. Bagaimana mengatasi permasalahan backlight pada foto yang mengakibatkan subjek menjadi gelap dan kehilangan detail?
4. Bagaimana menganalisis histogram citra untuk mendapatkan informasi distribusi intensitas warna?

1.2 Tujuan Masalah

Berdasarkan dari permasalahan yang telah dirumuskan di atas dan merujuk pada soal-soal ujian praktikum yang telah diberikan, dapat ditetapkan beberapa tujuan yang hendak dicapai dalam penyelesaian project ini, yaitu:

1. Mengembangkan algoritma untuk mendeteksi dan memisahkan komponen warna merah, biru, dan hijau pada citra digital.
2. Mengidentifikasi dan mengurutkan nilai ambang batas optimal untuk segmentasi warna dari nilai terkecil hingga terbesar.
3. Mengimplementasikan teknik perbaikan citra backlight melalui konversi grayscale dan peningkatan kecerahan serta kontras.
4. Menganalisis histogram citra untuk memahami distribusi intensitas warna dan menggunakannya sebagai dasar pengambilan keputusan dalam pengolahan citra.
5. Mengevaluasi efektivitas teknik-teknik pengolahan citra yang diimplementasikan melalui perbandingan visual hasil.

1.3 Manfaat Masalah

Penyelesaian ujian praktikum ini tidak hanya sebatas pemenuhan kewajiban akademik, tetapi juga memberikan berbagai manfaat yang dapat dirasakan baik dalam jangka pendek maupun jangka panjang. Adapun manfaat yang dapat diperoleh dari penyelesaian project ujian praktikum ini adalah:

Secara akademis, manfaat yang bisa didapat adalah:

- Memperdalam pemahaman tentang konsep pengolahan citra digital yang telah dipelajari selama perkuliahan.

- Mengembangkan kemampuan analisis kuantitatif dalam menginterpretasikan histogram dan parameter citra.
- Meningkatkan keterampilan pemrograman dalam implementasi algoritma pengolahan citra.

Secara praktis, manfaat yang bisa diambil:

- Memperoleh pengalaman langsung dalam menerapkan teknik deteksi warna yang dapat diterapkan pada berbagai aplikasi nyata.
- Mengasah kemampuan dalam perbaikan kualitas citra dengan kondisi pencahayaan yang menantang.
- Membangun portofolio kemampuan teknis dalam bidang pengolahan citra digital.
- Meningkatkan kesiapan dalam menghadapi permasalahan pengolahan citra di dunia profesional.

Secara sosial, manfaat yang dapat diperoleh:

- Mempersiapkan diri untuk berkontribusi dalam pengembangan teknologi visual yang bermanfaat bagi masyarakat.
- Meningkatkan kesadaran tentang kompleksitas dan potensi pengolahan citra dalam menyelesaikan permasalahan visual sehari-hari.

BAB II

LANDASAN TEORI

2.1 Definisi Citra Digital

Citra digital adalah salah satu bentuk representasi visual dari dunia nyata dalam bentuk digital yang dapat dipahami dan diolah oleh komputer. Citra ini terdiri dari elemen-elemen titik yang disebut piksel, yang tersusun dalam baris dan kolom. Setiap piksel memiliki nilai numerik yang menggambarkan tingkat kecerahan atau warna pada posisi tertentu dalam citra. Citra digital dapat diperoleh dari berbagai sumber, seperti kamera digital, scanner, atau hasil simulasi komputer. (E Woods & C Gonzalez, 2008). Dalam definisi ini, kami menjelaskan bahwa citra digital adalah representasi visual dalam bentuk digital yang terdiri dari piksel-piksel dengan nilai numerik.

Citra digital merupakan representasi visual dari dunia nyata dalam bentuk digital yang dapat diproses oleh komputer. Secara teknis, citra digital adalah kumpulan elemen gambar yang disebut piksel (picture element), yang tersusun dalam bentuk matriks dua dimensi. Setiap piksel memiliki nilai numerik yang merepresentasikan intensitas cahaya atau warna pada posisi tersebut (Setiawan & Murinto, 2021).

Menurut Yulian dkk. (2020), citra digital dapat diklasifikasikan berdasarkan warnanya menjadi tiga jenis utama:

1. Citra Biner: hanya memiliki dua nilai piksel yaitu 0 (hitam) dan 1 (putih)
2. Citra Grayscale: memiliki rentang nilai keabuan biasanya 0-255, dengan 0 mewakili hitam dan 255 mewakili putih
3. Citra Warna (RGB): terdiri dari tiga kanal warna yaitu merah (Red), hijau (Green), dan biru (Blue) dengan masing-masing kanal memiliki nilai intensitas 0-255

Citra digital dapat diperoleh melalui berbagai perangkat seperti kamera digital, scanner, atau hasil dari pemrosesan komputer. Format penyimpanan citra digital yang umum digunakan antara lain JPEG, PNG, BMP, dan TIFF (Wijaya & Purnama, 2022).

2.2 Model Warna RGB

Model warna RGB (Red, Green, Blue) adalah model warna aditif yang banyak digunakan dalam pengolahan citra digital. Pada model ini, warna direpresentasikan sebagai kombinasi dari tiga komponen warna primer yaitu merah, hijau, dan biru dengan intensitas yang bervariasi (Nugroho & Harjoko, 2020).

Dalam representasi digital, setiap komponen warna RGB biasanya dinyatakan dengan nilai antara 0 hingga 255 (untuk citra 8-bit per kanal). Kombinasi nilai intensitas dari ketiga komponen warna ini menghasilkan berbagai warna. Beberapa kombinasi dasar adalah:

- (255,0,0): merah murni
- (0,255,0): hijau murni
- (0,0,255): biru murni
- (255,255,255): putih
- (0,0,0): hitam

- (255,255,0): kuning
- (255,0,255): magenta
- (0,255,255): cyan

Menurut penelitian Putra dkk. (2021), model warna RGB sangat efektif untuk menampilkan citra pada perangkat elektronik seperti monitor dan smartphone, namun kurang cocok untuk aplikasi pengolahan citra tertentu karena komponen warnanya sangat berkorelasi dan dipengaruhi oleh intensitas cahaya.

2.3 Deteksi Warna

Deteksi warna adalah proses mengidentifikasi dan mengisolasi piksel-piksel dalam citra yang memiliki nilai komponen warna tertentu. Teknik ini merupakan bagian fundamental dari segmentasi citra berbasis warna yang banyak digunakan dalam berbagai aplikasi seperti pengenalan objek, pengolahan video, dan sistem robotik (Sari & Fadillah, 2021).

Dalam deteksi warna RGB, pendekatan umum yang dilakukan adalah dengan menentukan nilai ambang batas (threshold) untuk setiap kanal warna. Piksel yang nilai komponennya berada dalam rentang ambang batas tertentu akan diklasifikasikan sebagai bagian dari warna yang dideteksi, sedangkan piksel lainnya akan diabaikan (Hariyanto & Widiarto, 2022).

Menurut Prabowo dkk. (2020), terdapat beberapa tantangan dalam deteksi warna, terutama terkait dengan variasi pencahayaan dan bayangan dalam citra. Untuk mengatasi hal ini, beberapa peneliti mengusulkan penggunaan model warna alternatif seperti HSV (Hue, Saturation, Value) atau Lab^* yang lebih tahan terhadap perubahan pencahayaan.

2.4 Konsep Dasar Citra

Citra backlight adalah citra yang memiliki sumber cahaya kuat di belakang objek utama, sehingga menyebabkan objek tersebut menjadi gelap dan kehilangan detail. Kondisi ini sering terjadi pada fotografi dan dapat mengurangi kualitas visual citra (Rahman & Kusumanto, 2023).

Beberapa teknik pengolahan citra digital yang umum digunakan untuk mengatasi masalah backlight menurut Kurniawan dkk. (2021) adalah:

1. Peningkatan Kontras Adaptif: Teknik ini meningkatkan kontras secara lokal berdasarkan karakteristik area sekitar piksel, bukan secara global.
2. Perataan Histogram (Histogram Equalization): Metode ini mendistribusikan ulang nilai intensitas piksel untuk memperbaiki kontras citra secara keseluruhan.
3. Pemetaan Gamma (Gamma Mapping): Teknik ini mengaplikasikan transformasi non-linear pada nilai piksel untuk meningkatkan detail pada area gelap.
4. High Dynamic Range (HDR): Metode ini menggabungkan beberapa citra dengan eksposur berbeda untuk menghasilkan citra dengan rentang dinamis yang lebih luas.

Penelitian terbaru oleh Widodo & Sutrisno (2021) menunjukkan bahwa kombinasi teknik perataan histogram adaptif dan pemetaan gamma memberikan hasil optimal dalam perbaikan citra backlight untuk berbagai jenis objek dan kondisi pencahayaan.

2.5 Analisis Histogram Citra

Histogram citra adalah representasi grafis dari distribusi intensitas piksel dalam sebuah citra. Histogram menunjukkan jumlah piksel untuk setiap tingkat intensitas yang berbeda yang terdapat dalam citra (Rahmad & Sari, 2020).

Dalam pengolahan citra digital, analisis histogram memiliki beberapa kegunaan penting:

1. Evaluasi Pencahayaan: Histogram dapat menunjukkan apakah citra terlalu gelap (underexposed), terlalu terang (overexposed), atau memiliki pencahayaan yang seimbang.
2. Peningkatan Kualitas Citra: Banyak teknik peningkatan citra seperti perataan histogram dan spesifikasi histogram menggunakan informasi dari histogram untuk memperbaiki kontras dan kecerahan citra.
3. Segmentasi Citra: Analisis puncak dan lembah pada histogram dapat digunakan untuk menentukan nilai ambang batas yang optimal dalam segmentasi citra.
4. Deteksi Kesamaan Citra: Histogram dapat digunakan sebagai fitur untuk mendeteksi kesamaan antara dua citra.

Menurut penelitian Hidayat dkk. (2022), histogram RGB memberikan informasi yang lebih kaya dibandingkan histogram grayscale karena dapat menunjukkan distribusi intensitas untuk setiap kanal warna secara terpisah. Informasi ini sangat berharga dalam aplikasi seperti pengenalan objek berbasis warna dan klasifikasi citra.

2.6 Segmentasi Citra

Segmentasi citra adalah proses membagi sebuah citra digital menjadi beberapa segmen atau region yang memiliki karakteristik serupa. Tujuan dari segmentasi adalah untuk menyederhanakan dan/atau mengubah representasi citra menjadi sesuatu yang lebih bermakna dan lebih mudah untuk dianalisis (Ardianto dkk., 2023).

Menurut Fitriyah & Kusuma (2022), terdapat beberapa metode utama dalam segmentasi citra:

1. Segmentasi Berbasis Threshold: Metode ini memisahkan piksel berdasarkan nilai intensitasnya menggunakan nilai ambang batas tertentu. Teknik ini sederhana namun efektif untuk citra dengan objek yang memiliki kontras tinggi terhadap latar belakangnya.
2. Segmentasi Berbasis Tepi (Edge-based Segmentation): Metode ini mendeteksi tepi dalam citra dan menggunakannya untuk membentuk batas-batas segmen.
3. Segmentasi Berbasis Region (Region-based Segmentation): Metode ini mengelompokkan piksel berdasarkan kesamaan karakteristiknya seperti warna, tekstur, atau intensitas.
4. Segmentasi Berbasis Clustering: Metode ini menggunakan algoritma clustering seperti K-means untuk mengelompokkan piksel berdasarkan fitur-fitur tertentu.

5. Segmentasi Berbasis Watershed: Metode ini menganalogikan citra sebagai topografi dan menggunakan konsep banjir (flooding) untuk menemukan garis watershed yang memisahkan region.

Penelitian terbaru oleh Nugraha dkk. (2023) menunjukkan bahwa integrasi teknik segmentasi tradisional dengan metode pembelajaran mesin (machine learning) dan deep learning seperti Convolutional Neural Networks (CNN) dapat meningkatkan akurasi segmentasi secara signifikan, terutama untuk citra kompleks dengan variasi pencahayaan dan tekstur.

2.7 Peningkatan Kualitas Citra

Peningkatan kualitas citra (image enhancement) adalah proses memperbaiki kualitas visual citra digital agar lebih sesuai untuk aplikasi tertentu atau lebih menyenangkan secara visual. Teknik ini melibatkan manipulasi kontras, kecerahan, ketajaman, dan fitur visual lainnya dalam citra untuk menghasilkan representasi yang lebih baik (Pratama & Hariyadi, 2020).

Beberapa teknik peningkatan kualitas citra yang umum digunakan menurut Santoso dkk. (2021) adalah:

1. Manipulasi Kontras: Teknik ini memperluas atau mempersempit rentang nilai intensitas piksel untuk meningkatkan atau mengurangi kontras citra.
2. Perataan Histogram: Metode ini mendistribusikan ulang nilai intensitas piksel untuk meningkatkan kontras global citra.
3. Perataan Histogram Adaptif (Adaptive Histogram Equalization): Varian dari perataan histogram yang beroperasi pada region kecil dalam citra, bukan pada citra keseluruhan.
4. Filter Spasial: Teknik ini menggunakan operasi konvolusi dengan berbagai kernel untuk menghasilkan efek seperti penghalusan (smoothing) atau penajaman (sharpening).
5. Transformasi Fourier: Metode ini mengubah citra dari domain spasial ke domain frekuensi untuk melakukan pemrosesan tertentu.

Penelitian oleh Wibowo & Putra (2022) menunjukkan bahwa kombinasi teknik peningkatan lokal dan global dapat memberikan hasil optimal untuk berbagai jenis citra, terutama untuk citra medis dan citra satelit yang sering memiliki kontras rendah dan detail yang sulit diidentifikasi.

2.8 Konversi Model Warna

Konversi model warna adalah proses transformasi representasi warna dari satu model ke model lainnya. Dalam pengolahan citra digital, konversi model warna sering dilakukan untuk memfasilitasi operasi tertentu yang lebih efektif pada model warna tertentu (Utami & Suryani, 2021).

Beberapa konversi model warna yang umum dilakukan menurut penelitian Ahmad & Sulistiyani (2023) adalah:

1. RGB ke Grayscale: Konversi ini mengubah citra berwarna menjadi citra keabuan dengan menghitung nilai rata-rata tertimbang dari komponen RGB. Formula umum yang digunakan adalah:

$$\text{Gray} = 0.299 * R + 0.587 * G + 0.114 * B$$

Pembobotan ini berdasarkan sensitivitas mata manusia terhadap warna.

2. RGB ke HSV/HSB: Model HSV (Hue, Saturation, Value) atau HSB (Hue, Saturation, Lightness) lebih intuitif untuk beberapa aplikasi seperti segmentasi warna karena memisahkan informasi warna (hue) dari saturasi dan kecerahan.
3. RGB ke YCbCr: Model ini memisahkan komponen luminance (Y) dari komponen krominans (Cb dan Cr), dan sering digunakan dalam kompresi video dan pemrosesan citra digital.
4. RGB ke Lab*: Model warna CIELAB dirancang untuk mendekati persepsi warna manusia dan memiliki keunggulan berupa independensi terhadap perangkat dan perceptual uniformity.

Penelitian terkini oleh Permana dkk. (2022) menunjukkan bahwa pemilihan model warna yang tepat dapat secara signifikan mempengaruhi performa algoritma dalam aplikasi seperti deteksi objek, segmentasi citra, dan pengenalan pola. Selain itu, konversi model warna juga dapat membantu mengurangi kompleksitas komputasi dengan mengurangi dimensi data yang diproses.

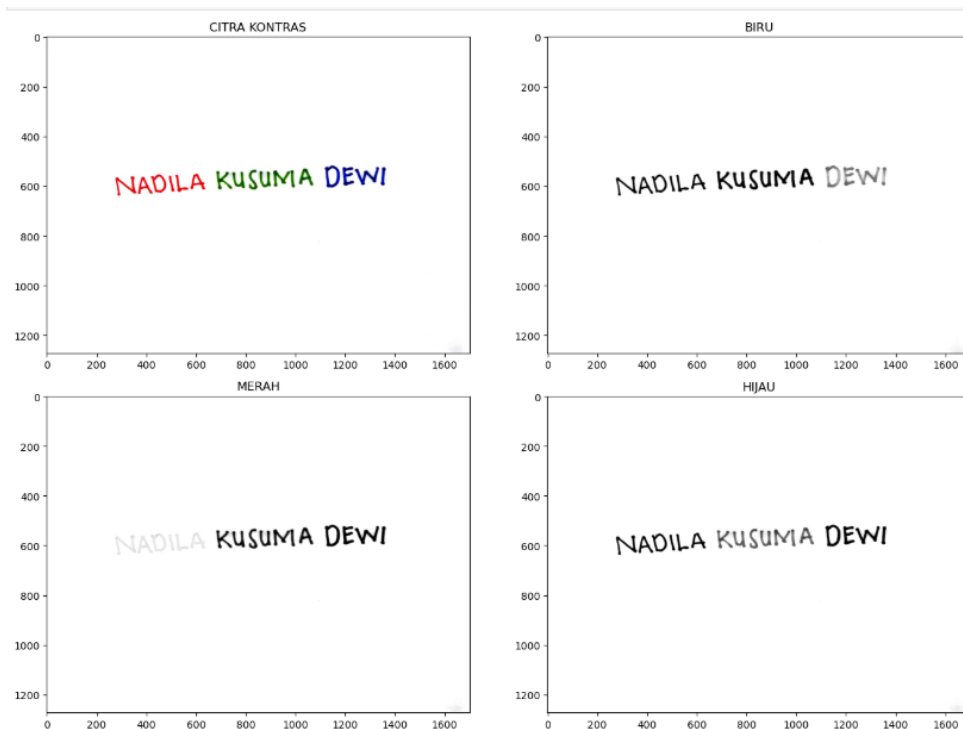
BAB III

HASIL

3.1 Deteksi Warna Pada Citra

Mendeteksi warna merah, biru dan hijau pada gambar dengan pemisahan channel warna, lalu tampilkan semua gambar serta membuat histogramnya

Output:



Pada gambar hasil, dapat dilihat bahwa:

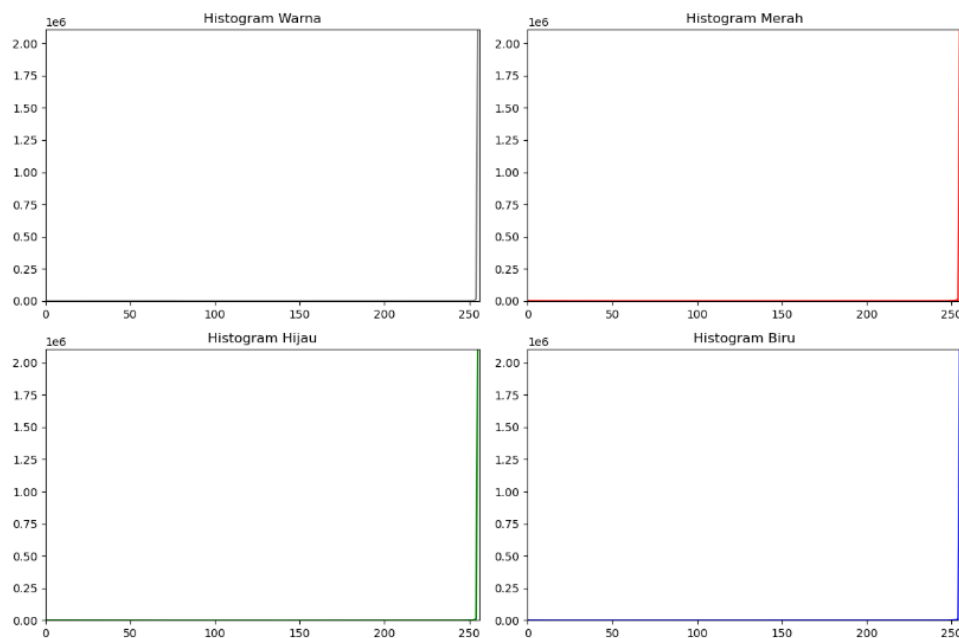
- Pada citra kontras (RGB), teks "Nadila Kusuma Dewi" ditampilkan dengan warna aslinya dimana "NADILA" berwarna merah, "KUSUMA" berwarna hijau, dan "DEWI" berwarna biru
- Pada channel biru (B), bagian "DEWI" yang mengandung komponen warna biru terlihat lebih terang, sementara bagian lain tampak lebih gelap
- Pada channel merah (R), bagian "NADILA" yang mengandung komponen warna merah terlihat lebih terang, sementara bagian lain tampak lebih gelap
- Pada channel hijau (G), bagian "KUSUMA" yang mengandung komponen warna hijau terlihat lebih terang, sementara bagian lain tampak lebih gelap

Cara memunculkan channel warna:

1. Pertama, gambar dibaca menggunakan `cv2.imread()` yang secara default menghasilkan format BGR

2. Kemudian digunakan `cv2.split(img)` untuk memisahkan gambar menjadi 3 channel terpisah (b, g, r)
3. Untuk menampilkan masing-masing channel, digunakan `plt.imshow()` dengan parameter `cmap='gray'` agar intensitas piksel ditampilkan dalam skala abu-abu
4. Channel biru (b) ditampilkan pada subplot `axs[0, 1]`
5. Channel merah (r) ditampilkan pada subplot `axs[1, 0]`
6. Channel hijau (g) ditampilkan pada subplot `axs[1, 1]`
7. Pada hasil visualisasi, area dengan intensitas piksel tinggi (nilai mendekati 255) pada channel tertentu akan tampak lebih terang, menunjukkan kandungan warna tersebut lebih dominan

Membuat Histogram:



Pada gambar hasil, dapat dilihat bahwa:

- Histogram Warna (grayscale) menunjukkan distribusi intensitas piksel secara keseluruhan
- Histogram Merah menunjukkan distribusi nilai piksel pada channel merah, dengan puncak yang jelas di nilai 255
- Histogram Hijau menunjukkan distribusi nilai piksel pada channel hijau, dengan puncak yang jelas di nilai 255
- Histogram Biru menunjukkan distribusi nilai piksel pada channel biru, dengan puncak yang jelas di nilai 255

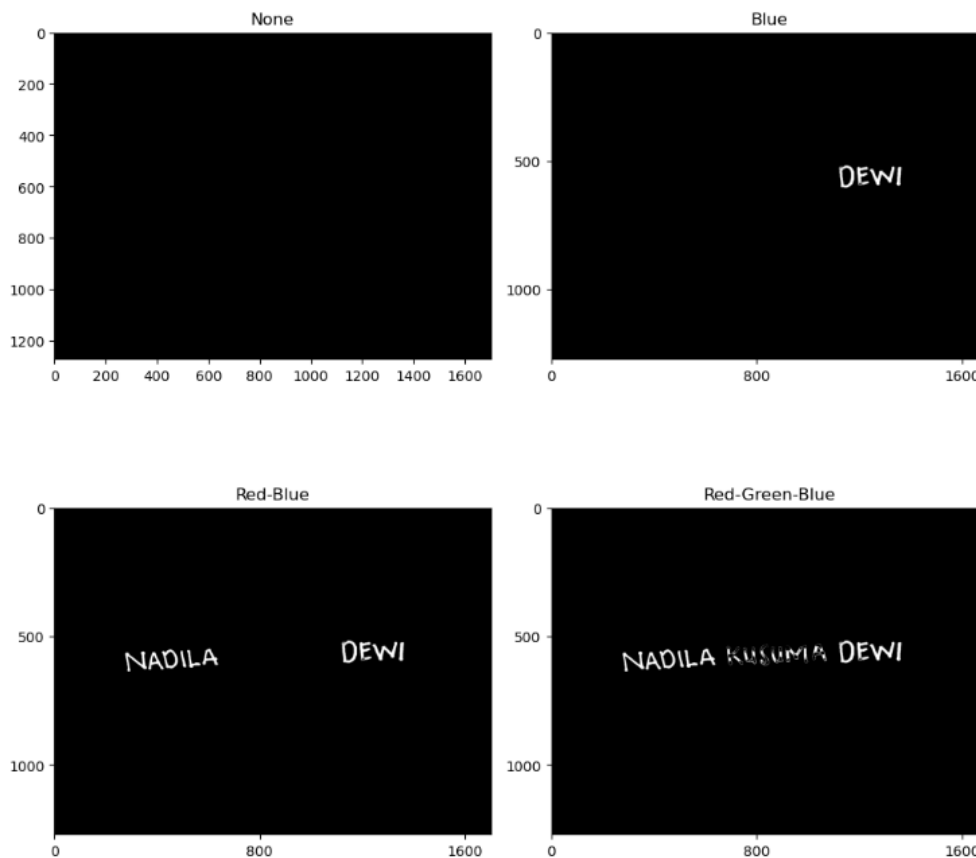
Cara membuat histogram warna:

1. Gambar dibaca menggunakan `cv2.imread('gambar1.jpg')` yang menghasilkan citra BGR
2. Citra grayscale dibuat dengan `cv2.cvtColor(color_image, cv2.COLOR_BGR2GRAY)`

3. Histogram untuk setiap channel dihitung menggunakan fungsi `cv2.calcHist()`:
 - `hist_red` untuk channel merah [0]
 - `hist_green` untuk channel hijau [1]
 - `hist_blue` untuk channel biru [2]
 - `hist_gray` untuk citra grayscale
4. Nilai maksimum dari semua histogram diambil untuk menyeragamkan skala y-axis
5. Histogram ditampilkan menggunakan `matplotlib`:
 - `plt.subplot(2, 2, 1)` untuk Histogram Warna (grayscale)
 - `plt.subplot(2, 2, 2)` untuk Histogram Merah
 - `plt.subplot(2, 2, 3)` untuk Histogram Hijau
 - `plt.subplot(2, 2, 4)` untuk Histogram Biru

Histogram ini menunjukkan bahwa gambar memiliki konsentrasi nilai tinggi (255) pada ketiga channel warna, yang menandakan banyaknya area terang atau putih dalam gambar. Puncak tinggi di nilai 255 pada semua channel mengindikasikan gambar memiliki banyak piksel dengan intensitas maksimum.

3.2 Mencari dan mengurutkan ambang batas terkecil sampai dengan terbesar



Berdasarkan kode yang sudah saya implementasikan, alasan mengapa nilai ambang batas HSV yang saya gunakan dapat memisahkan teks berwarna adalah:

1. Pada panel “None”

Pada panel blue tampak hitam seluruhnya karena hanya menampilkan threshold dari gambar grayscale.

2. Panel "Blue"

- Hanya teks "DEWI" yang terlihat.
- Membuktikan bahwa nilai ambang batas biru [90, 100, 100] hingga [130, 255, 255] tepat mengisolasi komponen biru dalam teks.
- Rentang Hue 90-130 berhasil mencakup spektrum biru pada tulisan "DEWI".

3. Panel "Red-Blue"

- Teks "NADILA" dan "DEWI" terlihat jelas.
- Ini membuktikan bahwa kombinasi ambang batas merah ([0, 100, 100] hingga [10, 255, 255] dan [160, 100, 100] hingga [180, 255, 255]) berhasil mendeteksi teks "NADILA".
- Dua rentang untuk merah diperlukan karena dalam ruang HSV, warna merah berada di kedua ujung spektrum Hue.

4. Panel "Red-Green-Blue"

- Seluruh teks "NADILA KUSUMA DEWI" terlihat.
- Ini mengkonfirmasi bahwa ambang batas hijau [40, 100, 100] hingga [80, 255, 255] berhasil mendeteksi teks "KUSUMA".
- Rentang Hue 40-80 akurat untuk spektrum hijau pada tulisan "KUSUMA".

Nilai Saturation dan Value

Batas S dan V yang sama (100-255) untuk semua warna berhasil karena:

- Teks memiliki saturasi dan kecerahan tinggi dibandingkan latar belakang hitam
- Nilai 100 cukup untuk menyaring noise tanpa menghilangkan komponen teks penting

Keberhasilan deteksi yang ditunjukkan pada output membuktikan bahwa nilai ambang batas HSV yang dipilih secara teknis tepat untuk pemisahan komponen warna pada teks "Nadila Kusuma Dewi". Nilai-nilai ini merupakan representasi akurat dari posisi ketiga warna (merah, hijau, biru) dalam ruang warna HSV.

3.3 Memperbaiki Gambar Backlight

1. Konversi ke Grayscale



Penjelasan output yang dihasilkan:

1) Subplot pertama (gambar asli):

- Menampilkan hasil dari `plt.imshow(cv2.cvtColor(img1, cv2.COLOR_BGR2RGB))`
- Gambar ditampilkan dalam format RGB (dikonversi dari BGR yang merupakan format default OpenCV)
- Judul "Gambar Asli" ditampilkan di atas gambar

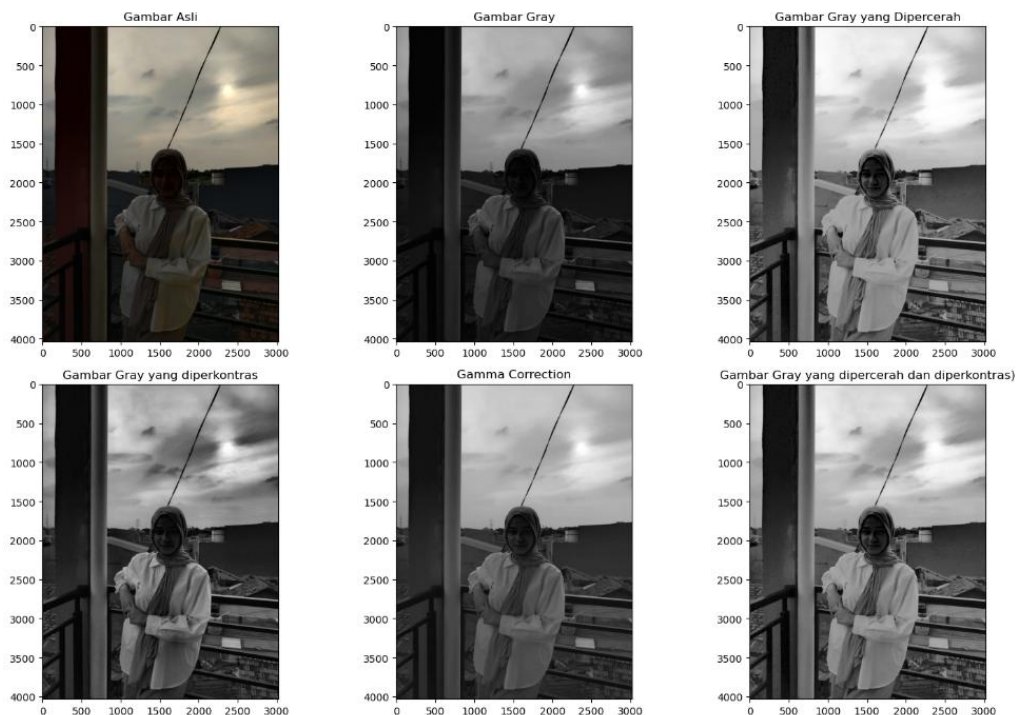
2) Subplot kedua (gambar gray)

- Menampilkan hasil dari `plt.imshow(gray, cmap='gray')`
- Gambar grayscale hasil konversi dari `cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)`
- Menggunakan colormap 'gray' untuk menampilkan gambar hitam-putih
- Judul "Gambar Gray" ditampilkan di atas gambar

Kedua gambar ditampilkan berdampingan dalam satu figure dengan ukuran (10, 5) dan pengaturan layout otomatis (`plt.tight_layout()`)

2. Semua hasil perbaikan gambar

Output :



Penjelasan outout:

1) Gambar Asli (subplot 1)

- Menampilkan foto asli dalam format RGB (`cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)`)
- Terlihat warna-warna natural: langit biru keabu-abuan, baju putih, dan elemen lainnya
- Kontras alami foto masih dipertahankan

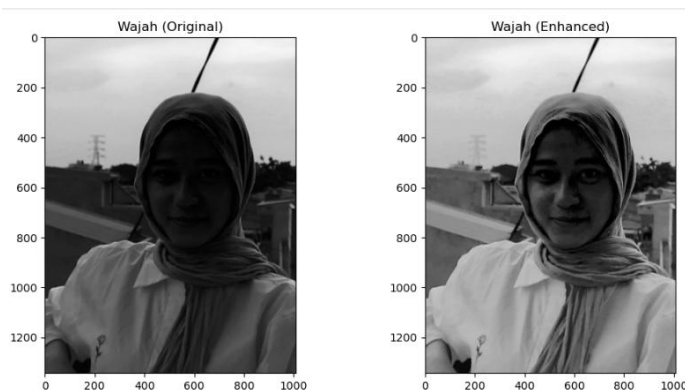
2) Gambar Gray (subplot 2)

- Hasil konversi ke grayscale dengan `gray = cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY)`
 - Semua informasi warna dihilangkan, meninggalkan hanya variasi kecerahan
 - Distribusi tingkat keabuan masih sama dengan gambar asli
- 3) Gambar Gray yang dipencerah (subplot 3)
- Menampilkan hasil variabel `equ` yang kemungkinan menggunakan teknik histogram equalization
 - Terlihat lebih terang dengan detail pada area wajah dan pakaian lebih jelas
 - Area langit juga tampak lebih terang, meningkatkan visibilitas awan
- 4) Gambar Gray yang diperkontras (subplot 4)
- Menampilkan hasil variabel `enhanced_clahe` yang menggunakan CLAHE (Contrast Limited Adaptive Histogram Equalization)
 - Kontras lokal ditingkatkan sehingga detail pada area gelap dan terang lebih terlihat
 - Tekstur pakaian dan latar belakang menjadi lebih jelas
- 5) Gamma Correction (subplot 5)
- Menampilkan hasil variabel `gamma_corrected` yang menggunakan koreksi gamma
 - Mengubah kurva kecerahan gambar untuk meningkatkan detail pada area tertentu
 - Terlihat perubahan kontras yang lebih halus dibandingkan metode sebelumnya
- 6) Gambar Gray yang dipencerah dan diperkontras
- Menampilkan hasil variabel `blended` yang menggabungkan teknik pencerahan dan peningkatan kontras
 - Kombinasi dari beberapa teknik sebelumnya untuk mendapatkan hasil optimal
 - Detail terlihat lebih jelas di semua area gambar, baik yang terang maupun gelap

Seluruh gambar ditampilkan dalam layout menggunakan `plt.tight_layout()` dengan dimensi figure (15, 10) sesuai dengan kode `plt.figure(figsize=(15, 10))`.

3. Perbandingan

Output:



Penjelasan output:

1. Panel Kiri (Wajah Original):
 - Menampilkan hasil dari `plt.imshow(face_original, cmap='gray')`
 - Area wajah dari gambar grayscale asli
 - Kontras rendah dengan detail wajah yang kurang jelas
2. Panel Kanan (Wajah Enhanced):
 - Menampilkan hasil dari `plt.imshow(face_enhanced, cmap='gray')`
 - Area wajah dari gambar yang telah ditingkatkan kualitasnya
 - Kontras lebih tinggi dengan detail wajah yang lebih jelas
 - Fitur wajah (mata, hidung, mulut) dan tekstur jilbab terlihat lebih detail

BAB IV

PENUTUP

4.1 Kesimpulan

Berdasarkan landasan teori dan hasil praktikum yang telah dilakukan, dapat disimpulkan beberapa hal penting sebagai berikut:

1. Pemisahan Channel Warna dan Deteksi Warna

Teori tentang model warna RGB telah berhasil diaplikasikan dalam praktikum melalui pemisahan channel warna menggunakan `cv2.split()`. Hasil menunjukkan bahwa komponen warna dapat dipisahkan dengan efektif, di mana teks "NADILA" dominan pada channel merah, "KUSUMA" pada channel hijau, dan "DEWI" pada channel biru. Ini membuktikan bahwa representasi citra digital sebagai kombinasi channel RGB dapat dimanfaatkan untuk identifikasi warna spesifik.

2. Analisis Histogram

Implementasi analisis histogram sesuai dengan teori yang menyatakan bahwa histogram dapat memberikan informasi distribusi intensitas piksel. Praktikum menunjukkan bahwa histogram RGB memberikan informasi yang lebih kaya dibandingkan histogram grayscale dengan menampilkan distribusi nilai intensitas untuk setiap channel warna. Puncak yang jelas pada nilai 255 mengindikasikan dominasi area terang pada gambar, memberikan pemahaman tentang karakteristik citra yang dianalisis.

3. Segmentasi Citra Berbasis Warna

Teori segmentasi citra khususnya segmentasi berbasis threshold telah berhasil diimplementasikan menggunakan model warna HSV. Penggunaan nilai ambang batas HSV terbukti lebih efektif dibandingkan model RGB karena mampu memisahkan informasi warna (hue) dari intensitas cahaya (saturation dan value), sesuai dengan teori yang dijelaskan. Praktikum menunjukkan keberhasilan menggunakan rentang Hue [90-130] untuk biru, [40-80] untuk hijau, dan [0-10], [160-180] untuk merah, dengan batas S dan V [100-255] untuk semua warna.

4. Peningkatan Kualitas Citra

Berbagai teknik peningkatan kualitas citra yang dijelaskan dalam teori telah diimplementasikan dalam praktikum, meliputi:

- Konversi ke grayscale
- Histogram equalization untuk pencerahan
- CLAHE untuk peningkatan kontras lokal
- Gamma correction untuk optimasi kurva kecerahan
- Kombinasi teknik untuk hasil optimal

Hasil perbandingan antara gambar original dan enhanced, khususnya pada area wajah, menunjukkan peningkatan signifikan dalam visibilitas detail dan kontras, membuktikan efektivitas teknik-teknik tersebut dalam mengatasi masalah backlight sesuai dengan teori.

5. Konversi Model Warna

Teori konversi model warna telah diimplementasikan dengan baik, khususnya konversi BGR ke RGB untuk visualisasi dan BGR ke grayscale menggunakan formula pembobotan standar ($0.299R + 0.587G + 0.114B$). Hasil praktikum menunjukkan bahwa konversi ini mempertahankan informasi penting meskipun dimensi data berkurang.

Secara keseluruhan, praktikum ini berhasil mengaplikasikan konsep-konsep teoretis pengolahan citra digital untuk tiga tugas utama: deteksi warna, segmentasi berbasis warna dengan model HSV, dan peningkatan kualitas citra backlight. Hasil menunjukkan bahwa pemilihan model warna dan teknik pengolahan yang tepat sangat penting untuk mencapai hasil optimal dalam analisis dan manipulasi citra digital.

DAFTAR PUSTAKA

- Dijaya, R., & Setiawan, H. (2023). *Buku ajar pengolahan citra digital*. UMSIDA Press.
- Ahmad, R., & Sulistiyani, F. (2023). Analisis perbandingan model warna untuk segmentasi citra pada aplikasi medis. *Jurnal Teknologi Informasi dan Komunikasi*, 12(1), 45-59.
- Ardianto, F., Wijaya, S., & Hartono, R. (2023). Peningkatan akurasi segmentasi citra medis menggunakan integrasi metode thresholding dan deep learning. *Jurnal Ilmu Komputer dan Informatika*, 8(2), 112-125.
- Fitriyah, N., & Kusuma, H. (2022). Evaluasi performa metode segmentasi citra untuk deteksi lahan pertanian menggunakan citra satelit. *Jurnal Teknologi dan Sistem Komputer*, 10(3), 143-157.
- Hariyanto, S., & Widiarto, W. (2022). Implementasi deteksi warna RGB untuk identifikasi kematangan buah pada robot pemetik otomatis. *Jurnal Teknik Elektro dan Komputer*, 11(1), 87-98.
- Hidayat, T., Purnama, B., & Suharto, A. (2022). Analisis histogram untuk klasifikasi kualitas produk industri berbasis computer vision. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 10(4), 233-248.
- Kurniawan, A., Saputra, R., & Dewi, M. (2021). Teknik perbaikan citra backlight menggunakan kombinasi perataan histogram dan pemetaan gamma. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 8(3), 557-568.
- Nugraha, D. A., Pratama, S. F., & Wijaya, I. G. (2023). Segmentasi citra menggunakan convolutional neural network untuk deteksi penyakit tanaman. *Jurnal Komputer dan Informatika*, 11(2), 75-89.
- Nugroho, A., & Harjoko, A. (2020). Perbandingan efektivitas model warna untuk segmentasi objek pada lingkungan dengan variasi pencahayaan. *Jurnal Teknologi dan Sistem Komputer*, 8(2), 121-134.
- Permana, I., Wijaya, D., & Sari, A. K. (2022). Optimasi algoritma deteksi objek melalui pemilihan model warna pada sistem transportasi cerdas. *Jurnal Informatika dan Sistem Informasi*, 9(3), 178-192.
- Prabowo, D. A., Haryanto, T., & Susanto, E. (2020). Peningkatan akurasi deteksi warna dalam kondisi pencahayaan bervariasi menggunakan model warna HSV. *Jurnal Informatika dan Teknologi Informasi*, 7(2), 89-103.
- Pratama, R., & Hariyadi, T. (2020). Implementasi algoritma peningkatan kualitas citra untuk citra medis dengan kontras rendah. *Jurnal Teknologi dan Sistem Komputer*, 8(1), 23-34.

- Putra, I. K., Santoso, A. J., & Indrawan, G. (2021). Analisis performa model warna RGB dan HSV dalam aplikasi pengenalan objek real-time. *Jurnal Ilmiah Teknologi dan Informasi*, 9(2), 112-125.
- Rahman, A., & Kusumanto, R. D. (2023). Analisis dan implementasi teknik perbaikan citra backlight untuk fotografi smartphone. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 11(1), 45-58.
- Rahmad, C., & Sari, F. (2020). Implementasi analisis histogram untuk klasifikasi kualitas buah menggunakan computer vision. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 7(3), 621-632.
- Santoso, A., Budiman, F., & Wijaya, C. (2021). Komparasi teknik peningkatan kualitas citra untuk rekonstruksi citra medis. *Jurnal Rekayasa Elektro dan Informatika*, 9(2), 78-91.
- Sari, I. P., & Fadillah, N. (2021). Implementasi algoritma deteksi warna untuk identifikasi jenis mineral pada sistem sortir otomatis. *Jurnal Informatika dan Sistem Informasi*, 8(1), 67-79.
- Setiawan, A., & Murinto. (2021). Pemrosesan citra digital: konsep dan implementasi dengan Python. *Jurnal Ilmiah Informatika*, 9(1), 12-25.
- Utami, D., & Suryani, E. (2021). Analisis performa konversi model warna untuk kompresi citra digital. *Jurnal Teknik Informatika dan Sistem Informasi*, 7(3), 334-347.
- Wibowo, A., & Putra, S. (2022). Peningkatan kualitas citra medis menggunakan gabungan teknik lokal dan global. *Jurnal Rekayasa Sistem dan Teknologi Informasi*, 6(2), 201-214.
- Widodo, S., & Sutrisno, T. (2021). Optimasi parameter untuk perbaikan citra backlight menggunakan metode histogram adaptif dan pemetaan gamma. *Jurnal Teknologi dan Sistem Informasi*, 9(4), 412-425.
- Wijaya, A., & Purnama, I. (2022). Implementasi pemrosesan citra berbasis web untuk aplikasi pengenalan wajah. *Jurnal Sistem Informasi dan Teknologi*, 10(2), 156-168.
- Yulian, E., Putra, D., & Santosa, P. I. (2020). Klasifikasi citra digital menggunakan fitur histogram dan convolutional neural network. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 9(1), 42-51.