**LAPORAN IMAGE ENHANCEMENT**

**IMAGE PROCESSING**

****

**DISUSUN OLEH:**

1. **Muhammad Al Hafiz 2211533010**
2. **Naufal Dira Agustian 2211533011**
3. **Najwa Azka Thalita Mehdi 2211533012**
4. **Rafano Alesander 2211533016**

**Dosen Pengampu:**

**Afdhal**

**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**PADANG**

**2025**

1. Pendahuluan

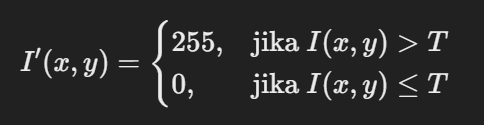
Image enhancement merupakan teknik untuk meningkatkan kualitas gambar agar lebih mudah dianalisis oleh manusia maupun sistem komputer. Proses ini bertujuan untuk memperjelas detail, meningkatkan kontras, serta mengurangi gangguan atau noise dalam gambar.

Terdapat beberapa metode yang akan diterapkan untuk meningkatkan kualitas gambar, yaitu:

1. Segmentasi gambar dengan image thresholding

Teknik segmentasi digunakan untuk memisahkan objek dari latar belakang dalam sebuah gambar. Image thresholding bekerja dengan menetapkan nilai ambang (threshold) tertentu, sehingga piksel dengan nilai intensitas di atas atau di bawah ambang batas tersebut akan dikategorikan ke dalam kelompok yang berbeda.

Rumus Dasar:



1. **Mengubah gambar menjadi** image negative

Teknik ini bekerja dengan membalik nilai intensitas setiap piksel dalam gambar. Tujuan dari metode ini adalah untuk meningkatkan keterbacaan detail yang mungkin kurang terlihat pada area terang atau gelap.

Rumus Dasar;

1. Meningkatkan kontras dengan contrast stretching

Contrast stretching merupakan metode yang digunakan untuk memperluas rentang intensitas piksel dalam gambar, sehingga kontras menjadi lebih tinggi dan detail lebih mudah terlihat.

Rumus dasar:

Di mana dan adalah nilai intensitas minimum dan maksimum dalam citra asli.

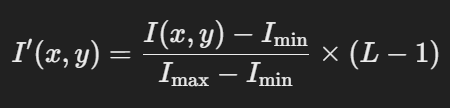
1. Memotong bagian gambar (cropping) menggunakan image subtraction

Metode ini digunakan untuk menyeleksi bagian tertentu dalam gambar dengan cara menghilangkan atau mengurangi bagian yang tidak diperlukan menggunakan teknik image subtraction.

1. Normalisasi histogram gambar

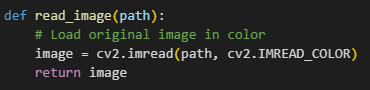
Normalisasi histogram bertujuan untuk menyebarkan distribusi intensitas piksel dalam gambar agar lebih merata. Dengan teknik ini, gambar yang terlalu terang atau terlalu gelap dapat diperbaiki agar memiliki keseimbangan warna yang lebih baik.

Rumus dasar:



di mana L adalah jumlah tingkat intensitas dalam gambar (biasanya 256 untuk gambar 8-bit).

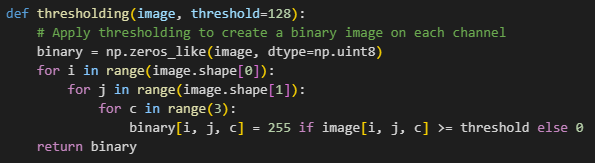
1. Penjelasan kode program
2. Fungsi read\_image(path)



Penjelasan:

* Fungsi ini digunakan untuk membaca gambar dari file dengan menggunakan OpenCV (cv2.imread()).
* Parameter:
  + path: Lokasi atau path file gambar yang ingin dibaca.
* OpenCV membaca gambar dalam format BGR (bukan RGB).
* cv2.IMREAD\_COLOR memastikan bahwa gambar dibaca dalam mode berwarna (bukan grayscale atau transparan).

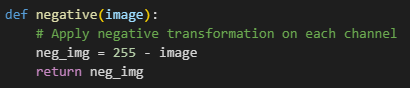
1. Fungsi thresholding(image, threshold=128)



Penjelasan:

* Fungsi ini menerapkan image thresholding untuk mengubah gambar menjadi biner.
* Prosesnya:
* Membuat gambar baru binary dengan ukuran yang sama tetapi diisi dengan nol (np.zeros\_like(image)).
* Melakukan iterasi ke seluruh piksel gambar:
  + Jika nilai intensitas piksel lebih besar dari atau sama dengan threshold (default: 128), maka diubah menjadi 255 (putih).
  + Jika lebih kecil, diubah menjadi 0 (hitam).
* Hasilnya adalah gambar biner di setiap kanal warna.

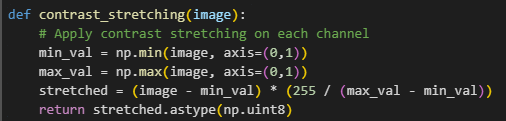
1. Fungsi negative(image)



Penjelasan:

* Fungsi ini mengubah gambar menjadi gambar negatif.
* Prosesnya:
  + - Setiap piksel
    - 𝐼(x,y) dikonversi menjadi 𝐼′(x,y)=255−I(x,y).
    - Piksel terang akan menjadi gelap, dan sebaliknya.
* Hasilnya adalah inversi warna dari gambar asli.

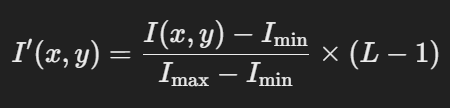
1. Fungsi contrast\_stretching(image)



Penjelasan:

* Fungsi ini menerapkan contrast stretching untuk memperluas rentang intensitas warna dalam gambar.
* Prosesnya:

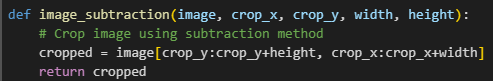
1. Menentukan nilai minimum (min\_val) dan maksimum (max\_val) dari setiap kanal warna.
2. Menggunakan rumus:



untuk merentangkan nilai intensitas dari [min, max] ke [0, 255].

* Hasilnya adalah gambar dengan kontras yang lebih baik.

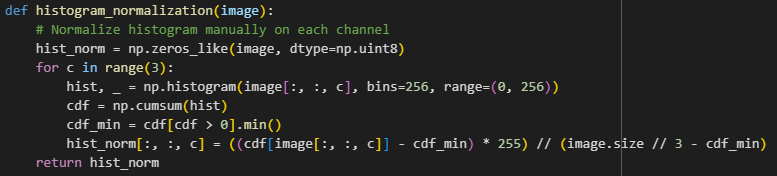
1. Fungsi image\_subtraction(image, crop\_x, crop\_y, width, height)



Penjelasan:

* Fungsi ini digunakan untuk memotong (cropping) bagian tertentu dari gambar.
* Prosesnya:
* Parameter crop\_x dan crop\_y menentukan titik awal pemotongan (koordinat kiri atas).
* width dan height menentukan ukuran potongan gambar.
* Menggunakan slicing pada array gambar untuk mengambil bagian yang diinginkan: cropped = image
* Hasilnya adalah gambar yang telah dipotong.

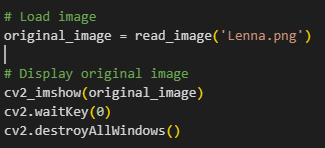
1. Fungsi histogram\_normalization(image)



Penjelasan:

* Fungsi ini melakukan normalisasi histogram untuk menyesuaikan distribusi intensitas warna dalam gambar.
* Prosesnya:
* Menghitung histogram dari setiap kanal warna menggunakan np.histogram().
* Menghitung CDF (Cumulative Distribution Function) untuk mengetahui penyebaran intensitas.
* Mengonversi hasil ke format uint8 agar bisa digunakan sebagai gambar.

1. Load dan Tampilkan Gambar Asli



Penjelasan:

* Gambar "Lenna.png" dimuat menggunakan fungsi read\_image(), yang membaca gambar dalam mode berwarna (BGR).
* cv2\_imshow(original\_image): Menampilkan gambar menggunakan OpenCV.
* cv2.waitKey(0): Menunggu input tombol sebelum menutup jendela.
* cv2.destroyAllWindows(): Menutup semua jendela yang terbuka.

1. Step 1: Thresholding



Step 1: ThresholdingPenjelasan:

* Fungsi thresholding(image, threshold=128) mengubah gambar menjadi biner (hitam-putih).
* Setiap piksel dibandingkan dengan nilai ambang (threshold), jika lebih besar atau sama dengan ambang, diubah menjadi 255 (putih), jika lebih kecil, diubah menjadi 0 (hitam).
* Hasilnya adalah gambar biner berdasarkan intensitas warna.

1. Step 2: Transformasi Negatif



Penjelasan:

* Fungsi negative(image) menerapkan transformasi negatif dengan rumus:
* I′ (x,y)=255−I(x,y)
* Semua warna terang akan menjadi gelap dan sebaliknya.
* Hasilnya adalah inversi warna dari gambar thresholding.

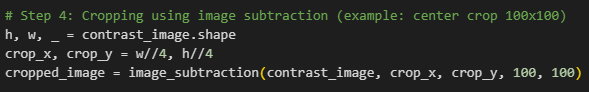
1. Step 3: Contrast Stretching



Penjelasan:

* Fungsi contrast\_stretching(image) meningkatkan kontras gambar dengan merentangkan nilai intensitas piksel dari rentang [min, max] ke [0, 255].
* Proses ini memperjelas detail pada gambar dengan memperbesar perbedaan intensitas warna.
* Hasilnya adalah gambar dengan kontras yang lebih tajam.

1. Step 4: Cropping menggunakan Image Subtraction



Penjelasan:

* Mengambil ukuran tinggi (h) dan lebar (w) dari gambar hasil kontras.
* Menentukan koordinat awal pemotongan dengan:
* crop\_x = w//4 (seperempat dari lebar gambar).
* crop\_y = h//4 (seperempat dari tinggi gambar).
* Fungsi image\_subtraction(image, crop\_x, crop\_y, 100, 100) memotong gambar dengan ukuran 100×100 piksel dari titik awal (crop\_x, crop\_y).
* Hasilnya adalah bagian tengah dari gambar yang telah ditingkatkan kontrasnya.

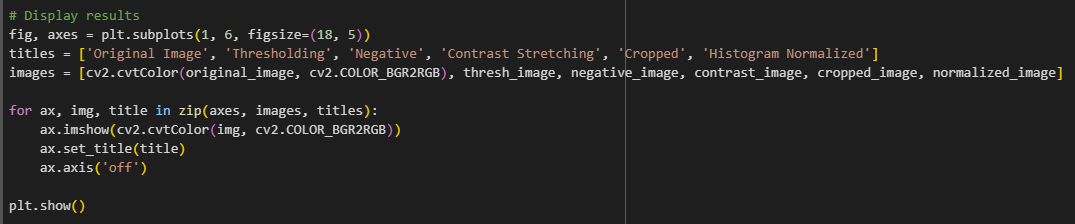
1. Step 5: Normalisasi Histogram



Penjelasan:

* Fungsi histogram\_normalization(image) menyesuaikan distribusi intensitas warna agar lebih merata.
* Histogram gambar asli dihitung, kemudian Cumulative Distribution Function (CDF) digunakan untuk menormalisasi nilai piksel.
* Tujuan: Meningkatkan kualitas gambar dengan meratakan penyebaran warna.
* Hasilnya adalah gambar dengan distribusi warna yang lebih seimbang.

1. Menampilkan Hasil dari Setiap Langkah



Penjelasan:

* plt.subplots(1, 6, figsize=(18, 5)): Membuat 6 subplot untuk menampilkan setiap tahapan gambar.
* Judul setiap gambar:
* Original Image – Gambar asli.
* Thresholding – Gambar biner setelah segmentasi.
* Negative – Inversi warna dari gambar biner.
* Contrast Stretching – Gambar dengan peningkatan kontras.
* Cropped – Bagian gambar yang telah dipotong.
* Histogram Normalized – Gambar setelah normalisasi histogram.
* cv2.cvtColor(img, cv2.COLOR\_BGR2RGB): Mengubah format BGR ke RGB agar sesuai dengan matplotlib.
* Loop for digunakan untuk menampilkan semua gambar dalam satu baris.

Output:

