

Nama : M. ISA KHALIM ZAMIL

NIM : 2355201038

MATKUL : PENGOLAHAN CITRA

TUGAS UTS

1. Deskripsikan perbedaan citra biner, grayscale, dan RGB

menggunakan satu gambar asli berwarna yang Anda miliki
(misal: foto taman/kendaraan).

- a. Tampilkan hasil konversi ke tiga jenis tersebut menggunakan Octave.

Codenya :

```
%Membaca Citra %
```

```
Img = imread('D:\gambar\becak.webp');
```

```
%Menampilkan Citra %
```

```
subplot(1,3,1), imshow(Img), title('Citra RGB');
```

```
%Menampilkan Ukuran Citra %
```

```
Ukuran = size(Img);
```

```
%Mengkonversi Citra Warna ke Grayscale %
```

```
RGB = imread('D:\gambar\becak.webp');
```

```
Abu = rgb2gray(RGB);
```

```
subplot(1,3,2), imshow(Abu), title('Citra Grayscale');
```

```
%Mengkonversi Citra Grayscale ke Biner %
```

```
Img = imread('D:\gambar\becak.webp');
```

```
[tinggi, lebar] = size(Img);
```

```
ambang = 210; % Nilai ini bisa diubah-ubah
```

```
biner = zeros(tinggi, lebar);
```

```
for baris=1 : tinggi
```

```
for kolom=1 : lebar
```

```
if Img(baris, kolom) >= ambang
```

```
    Biner(baris, kolom) = 0;
```



```

else
Biner(baris, kolom) = 1;
end
end
end
subplot(1,3,3), imshow(Biner), title('Citra Biner');

%Menyimpan Citra %
imwrite(Img, 'citra_RGB.png');
imwrite(Abu, 'citra_grayscale.png');
Biner = logical(Biner);
imwrite(Biner, 'citra_biner.png');

```

b. Dalam kasus kamera keamanan, mengapa pemrosesan awal citra sering dilakukan dalam bentuk grayscale atau biner?

Jawab:

Karena **grayscale atau biner** menyederhanakan data, mengurangi kompleksitas, dan memudahkan deteksi objek (gerakan, tepi) dengan cepat serta efisien untuk kamera keamanan.

- **Grayscale** memudahkan analisis bentuk dan tekstur, serta mengurangi kompleksitas.
- **Biner** cocok untuk deteksi objek (foreground vs background).

2. Lakukan kuantisasi pada sebuah citra grayscale menjadi 2, 4, dan 8 tingkat keabuan.

- a. Tampilkan hasilnya.
- b. Bandingkan kualitas visual tiap hasilnya.

Codenya :

```
% Membaca citra img = imread('D:\gambar\godong.jpg');
```

```

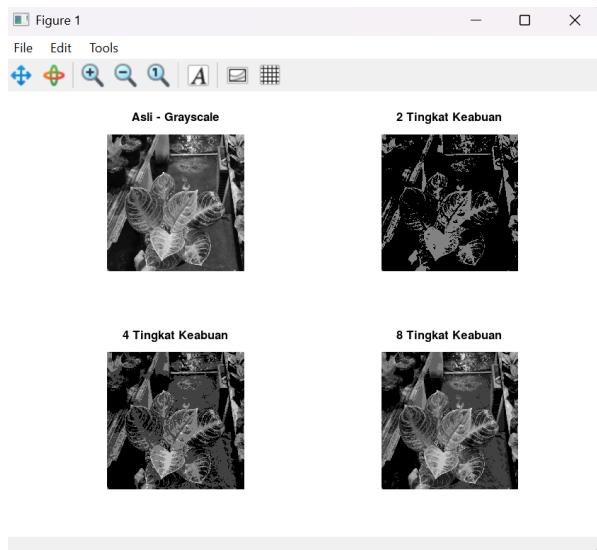
% Konversi ke grayscale jika belum if size(img, 3) == 3 img_gray = rgb2gray(img); else
img_gray = img; end

% Fungsi untuk mengkuantisasi citra ke n tingkat keabuan function quantized_img =
quantize_gray(img_gray, levels) step = 256 / levels; quantized_img =
floor(double(img_gray) / step) * step; quantized_img = uint8(quantized_img); end

% Kuantisasi menjadi 2, 4, dan 8 tingkat keabuan img_2 = quantize_gray(img_gray, 2);
img_4 = quantize_gray(img_gray, 4); img_8 = quantize_gray(img_gray, 8);

% Tampilkan hasil figure; subplot(2,2,1), imshow(img_gray), title('Asli - Grayscale');
subplot(2,2,2), imshow(img_2), title('2 Tingkat Keabuan'); subplot(2,2,3), imshow(img_4),
title('4 Tingkat Keabuan'); subplot(2,2,4), imshow(img_8), title('8 Tingkat Keabuan');

```



c. Dalam konteks pengiriman gambar melalui

WhatsApp yang lambat, bagaimana kuantisasi ini bisa membantu?

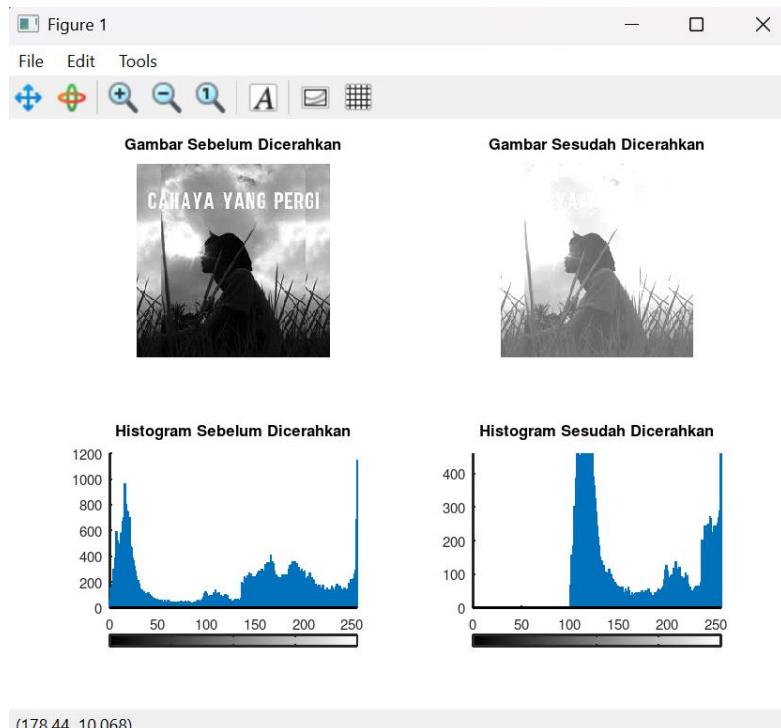
Jawab :

Kuantisasi membantu memperkecil ukuran gambar sehingga lebih cepat dikirim. Dengan mengurangi jumlah level warna atau intensitas piksel, data yang dikirim lebih sedikit, sehingga gambar bisa dikirim lebih cepat meski kualitas sedikit menurun.

3. Ambil satu foto dalam pencahayaan redup (gelap), kemudian:
 - a. Tingkatkan kecerahan citra dengan penyesuaian nilai piksel.
 - b. Tampilkan histogram sebelum dan sesudah.

Codenya :

```
pkg load image
% Membaca gambar
img = imread('D:\gambar\redup.jpg');
% Mengubah gambar RGB ke grayscale
imgray = rgb2gray(img);
% Meningkatkan kecerahan
cerah = imgray + 100;
cerah(cerah > 255) = 255; % Menjaga agar nilai pixel tidak melebihi 255
cerah = uint8(cerah);
% Menampilkan gambar sebelum dan sesudah cerah
figure;
subplot(2,2,1);
imshow(imgray);
title('Gambar Sebelum Dicerahkan');
subplot(2,2,2);
imshow(cerah);
title('Gambar Sesudah Dicerahkan');
% Menampilkan histogram
subplot(2,2,3);
imhist(imgray);
title('Histogram Sebelum Dicerahkan');
subplot(2,2,4);
imhist(cerah);
title('Histogram Sesudah Dicerahkan');
```



c. Apa dampaknya terhadap pengenalan objek di citra gelap jika hanya menggunakan operasi piksel?

Jawab :

Jika hanya menggunakan operasi piksel pada citra gelap, objek mungkin sulit dikenali karena **kontras rendah** (perbedaan antara objek dan latar belakang kecil) sehingga detail penting bisa hilang, membuat deteksi objek jadi kurang akurat.

4. Gunakan teknik ekualisasi histogram pada sebuah foto

wajah yang diambil di dalam ruangan.

a. Tampilkan citra sebelum dan sesudah.

Codenya :

```
pkg load image

% Baca dan konversi ke grayscale
Img = imread('D:\gambar\volly (1).jpg');
Img =
rgb2gray(Img); % Konversi ke grayscale

[jum_baris, jum_kolom] = size(Img);
L = 256; % Jumlah level intensitas
Histog = zeros(L, 1);

% Hitung histogram citra asli
for baris = 1 : jum_baris
    for kolom = 1 : jum_kolom
        nilai =
        Img(baris, kolom);
        Histog(nilai + 1) =
        Histog(nilai + 1) + 1;
    end
end

% Hitung distribusi kumulatif (CDF)
alpha = (L - 1) / (jumlah_baris * jumlah_kolom);
C(1) = alpha *
Histog(1);
for i = 2 : L
    C(i) =
    C(i - 1) + round(alpha * Histog(i));
end

% Transformasi piksel berdasarkan CDF
Hasil = Img;
for baris = 1 : jumlah_baris
    for kolom = 1 : jumlah_kolom
        Hasil(baris, kolom) =
        C(Img(baris, kolom) + 1);
    end
end

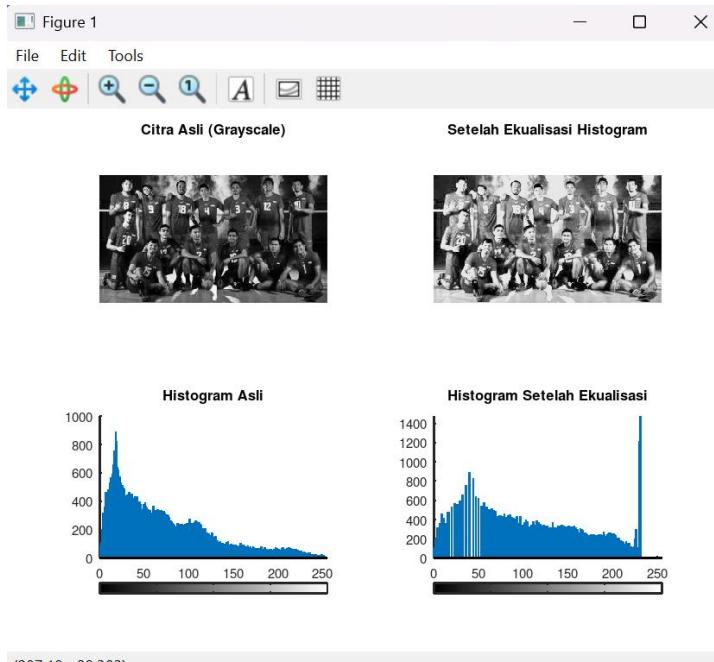
Hasil = uint8(Hasil);

% Tampilkan citra sebelum dan sesudah ekualisasi
figure;
subplot(2,2,1);
imshow(Img);
title('Citra Asli (Grayscale)');

subplot(2,2,2);
imshow(Hasil);
title('Setelah Ekualisasi Histogram');

% Tampilkan histogram sebelum dan sesudah
subplot(2,2,3);
imhist(Img);
title('Histogram Asli');
```

```
subplot(2,2,4); imhist(Hasil); title('Histogram Setelah Ekualisasi');
```



dalam sistem pengenalan wajah?

Jawab :

Karena ekualisasi histogram meningkatkan kontras citra, membuat detail wajah lebih jelas dan mempermudah deteksi fitur wajah (misalnya mata, hidung, mulut), sehingga meningkatkan akurasi sistem pengenalan wajah.

5. Ambil gambar yang memiliki banyak noise (misal: hasil screenshot dengan banyak bintik).

- Terapkan filter median menggunakan kernel 3x3.
- Bandingkan hasilnya dengan filter rata-rata (mean).

Codenya :

```
pkg load image; % Membaca gambar gambar = imread('D:\gambar\nois.webp');

% Konversi ke grayscale jika perlu if size(gambar, 3) == 3 gambar_gray = rgb2gray(gambar);
else gambar_gray = gambar; end

% Terapkan median filter 3x3 gambar_median = medfilt2(gambar_gray, [3 3]);

% Terapkan mean filter 3x3 (menggunakan konvolusi) kernel_mean = ones(3,3) / 9;
gambar_mean = conv2(double(gambar_gray), kernel_mean, 'same'); gambar_mean =
uint8(gambar_mean);
```

```
% Tampilkan hasil figure; subplot(1,3,1), imshow(gambar_gray), title('Asli (Grayscale)');
subplot(1,3,2), imshow(gambar_median), title('Median Filter 3x3'); subplot(1,3,3),
imshow(gambar_mean), title('Mean Filter 3x3');
```



ngapa

filter median lebih disukai?

Jawab :

Karena filter median efektif menghilangkan **noise impulsif** (seperti bintik hitam atau putih) tanpa mengaburkan tepi objek penting, sehingga struktur anatomi tetap jelas dan mendukung diagnosis yang lebih akurat.

6. Ambil foto benda bertekstur (misal: daun, kertas bergaris,

atau baju bermotif).

a. Terapkan filter High-Boost.

b. Tampilkan perbandingan sebelum dan sesudah.

Codenya :

```
F = imread('D:\gambar\godong.jpg'); F = rgb2gray(F); % Kernel Gaussian separabel Hkol =
[1 2 1]/4; Htrs = [1 2 1]/4;
```

```
% Faktor penguat high-boost A = 2.5;
```

```
%Fungsi High-Boost [tinggi_f, lebar_f] = size(F); [~, lebar_h] = size(Htrs); m2 =
floor(lebar_h/2); F2 = double(F); T = F2;
```

```
% Konvolusi vertikal for y = m2+1 : tinggi_f - m2 for x = 1 : lebar_f jum = 0; for p = -m2 : m2
jum = jum + Hkol(p + m2 + 1) * F2(y - p, x); end T(y, x) = jum; end end

% Konvolusi horizontal (low-pass) LPF = zeros(size(F2)); for y = 1 : tinggi_f for x = m2+1 :
lebar_f - m2 jum = 0; for p = -m2 : m2 jum = jum + Hbrs(p + m2 + 1) * T(y, x - p); end LPF(y, x)
= jum; end end % rumus High-Boost G = A * F2 - LPF; G = uint8(max(min(G, 255), 0));

% Tampilkan hasil subplot(1,2,1); imshow(F); title('Citra Asli'); subplot(1,2,2); imshow(G);
title('High-Boost');
```



ya bintik hitam atau putih) pada teks
|s, yang membuat **pengenalan**

karakter oleh OCR lebih akurat.

7. Ambil foto papan tulis atau spanduk dari sudut miring.

a. Rotasikan gambar agar tegak lurus dengan
interpolasi bilinear.

b. Perbesar citra agar teks mudah dibaca.

Codenya :

```
pkg load image;
clc; berkas ='D:\gambar\papan tulis.webp';
F = imread(berkas); img = rgb2gray(F); F = double(img);
```

```

function G = tbilin(F, a1, a2, a3, a4, b1, b2, b3, b4) [tinggi, lebar] = size(F); G = zeros(tinggi, lebar); % Inisialisasi output

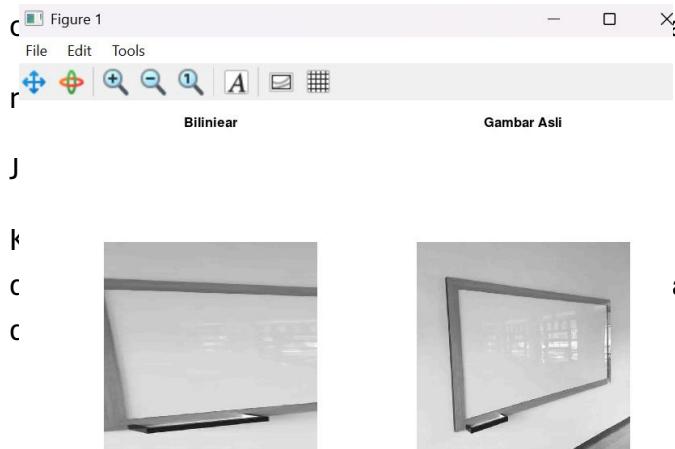
for y = 1 : tinggi for x = 1 : lebar x2 = a1 * x + a2 * y + a3 * x * y + a4; y2 = b1 * x + b2 * y + b3 * x * y + b4;

if (x2 >= 1) && (x2 <= lebar - 1) && (y2 >= 1) && (y2 <= tinggi - 1) p = floor(y2); q = floor(x2); a = y2 - p; b = x2 - q;

% Interpolasi bilinear intensitas = (1 - a) * ((1 - b) * F(p, q) + b * F(p, q + 1)) + a * ((1 - b) * F(p + 1, q) + b * F(p + 1, q + 1)); G(y, x) = intensitas; else G(y, x) = 0; end end end

G = uint8(G); end %acuan setting ##a1 = besarin gambar ##a2 = miring kanan kiri ##a3 = melengkung horizontal ##a4 = geser kanan kiri (pixel) ##b1 = miring atas bawah ##b2 = memanjang keatas ##b3 = melengkung vertikal ##b4 = geser atas bawah (pixel) % fungsi bilinier G = tbilin(F, 0.3,0,0,140, 0,1,0,0); % fungsi bilinear figure; subplot (1,2,1); imshow (G); title ('Bilinear'); subplot (1,2,2); imshow (img); title ('Gambar Asli');


```



suaikan dokumen agar teks **tegak lurus** au algoritma lainnya dapat **membaca**

8. T

wajah, pemandangan).

a. Bandingkan dengan citra aslinya.

Codenya :

```

% Baca gambar img = imread('D:\gambar\bromo.jpg'); img = im2double(img); % Ubah ke tipe double untuk manipulasi

% Ukuran gambar [rows, cols, ch] = size(img);

% Parameter ripple x_amplitude = 5; y_amplitude = 5; x_frequency = 0.05; y_frequency = 0.05;

% Buat meshgrid untuk koordinat asli [x, y] = meshgrid(1:cols, 1:rows);

```

```
% Hitung koordinat baru dengan efek ripple x_disp = x + x_amplitude * sin(2 * pi *  
y_frequency * y); y_disp = y + y_amplitude * sin(2 * pi * x_frequency * x);  
  
% Interpolasi dan hasil akhir ripple_img = zeros(size(img)); for c = 1:ch ripple_img(:,:,c) =  
interp2(x, y, img(:,:,c), x_disp, y_disp, 'linear', 0); end  
  
% Tampilkan hasil sebelum dan sesudah figure; subplot(1,2,1); imshow(img);  
title('Original Image'); subplot(1,2,2); imshow(ripple_img); title('Ripple Effect Applied');
```



di

Ja

Ef
di
sc
de

Original Image Ripple Effect Applied

embuat karya seni digital lebih
an atau perspektif baru. Di media
perhatian audiens, dan menonjolkan
estetis.

9. A) a dari

(2124.9, 30.541)

kamera HP).

a. Terapkan filter Gaussian untuk mengaburkan citra.

Codenya :

pkg load image;

```
% Membaca gambar img = imread('D:\gambar\dalan.jpg');
```

```
% Konversi ke grayscale jika RGB if ndims(img) == 3 img = rgb2gray(img); endif
```

```
% Buat filter Gaussian sigma = 2; % Standar deviasi Gaussian (semakin besar, semakin  
blur) kernel_size = 2 * ceil(3 * sigma) + 1; % Ukuran kernel berdasarkan sigma h =  
fspecial('gaussian', kernel_size, sigma); % Buat kernel Gaussian
```

```
% Terapkan filter ke gambar blurred_img = imfilter(img, h, 'replicate');
```

```
% Tampilkan hasil figure; subplot(1,2,1), imshow(img), title('Asli'); subplot(1,2,2),  
imshow(blurred_img), title('Gaussian Blur');
```



privasi

Jawab

Teknik
objek
atau o
untuk
relevan



ngaburkan detail wajah atau objektif sehingga **identitas orang** jaga privasi saat citra digunakan pengenalan individu yang tidak dikenal.

(409.42, 62.232)

10. Gunakan citra papan kotak atau bangunan, lalu terapkan

transformasi affine (rotasi, skala, translasi sekaligus).

a. Tampilkan hasil transformasi.

Codenya :

```
F = rgb2gray(imread('D:\gambar\bangunan.jpeg'));

% Definisikan fungsi taffine function G = taffine(F, a11, a12, a21, a22, tx, ty) % TAFFINE
Digunakan untuk melakukan transformasi affine. % Masukan: F = Citra berskala keabuan
% a11, a12, a21, a22, tx, ty = parameter transformasi affine

% Inisialisasi matriks G dengan ukuran yang sama dengan F [tinggi, lebar] = size(F); G =
zeros(tinggi, lebar);

% Iterasi untuk setiap piksel for y = 1:tinggi for x = 1:lebar % Hitung koordinat baru x2 =
round(a11 * x + a12 * y + tx); y2 = round(a21 * x + a22 * y + ty);

% Periksa apakah koordinat baru berada dalam batas citra
if (x2 >= 1) && (x2 <= lebar) && (y2 >= 1) && (y2 <= tinggi)
    % Lakukan interpolasi bilinear
    p = floor(y2);
    q = floor(x2);
    a = y2 - p;
    b = x2 - q;

    if (p == tinggi) || (q == lebar)
```

```

G(y, x) = F(min(p, tinggi), min(q, lebar));
else
    intensitas = (1-a)*(1-b)*double(F(p,q)) + b*double(F(p, q+1))
+
... .
    a *((1-b)* double(F(p+1, q)) + b * double(F(p+1,
q+1)));
    G(y, x) = uint8(intensitas);
end
else
    G(y, x) = 0;
end
end

end

% Pastikan G dalam format uint8 G = uint8(G); endfunction

% Tentukan sudut rotasi dalam radian rad = pi/4; % Contoh sudut 45 derajat

% Penggunaan fungsi G = taffine(F, 2 * cos(rad), sin(rad), -sin(rad), 2 * cos(rad), -30, -50);

% Tampilkan hasil

subplot(1,2,1); imshow (F); title ('citra asli'); subplot(1,2,2); imshow (G); title ('affine');

```



an bangunan atau navigasi drone dengan:

(65.968, 580.48) a akibat sudut kamera) sehingga peta atau gambar lebih akurat.

- Menjaga hubungan geometris** (misalnya garis tetap lurus) agar drone dapat memahami tata letak bangunan secara konsisten.

- ✓ **Mempermudah integrasi peta digital** (misalnya untuk overlay citra dari berbagai sudut pandang).