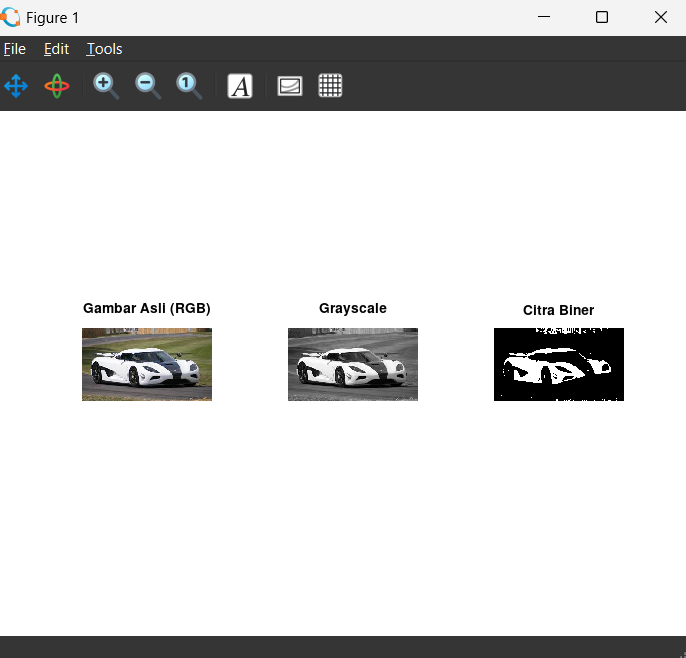
UTS PENGOLAHAN CITRA SEMESTER 4 ILMU KOMPUTER

NAMA : MOHAMMAD RIYAN HARIYANTO

NIM : 2355201036

1. **Biner, GreySCALE, RGB**
2. 

**CODE OCTAV:**

pkg load image;

% Baca gambar

gambar\_rgb = imread('D:\cp\citra2\Biner,Grey,RGB.jpeg');

% Konversi ke grayscale

gambar\_gray = rgb2gray(gambar\_rgb);

% Konversi ke biner dengan ambang otomatis

gambar\_gray = rgb2gray(gambar\_rgb); % konversi grayscale

threshold = graythresh(gambar\_gray); % cari nilai ambang otomatis

gambar\_biner = gambar\_gray > threshold \* 255; % manual thresholding

% Menampilkan ketiganya dalam satu figure

figure;

subplot(1, 3, 1);

imshow(gambar\_rgb);

title('Gambar Asli (RGB)');

subplot(1, 3, 2);

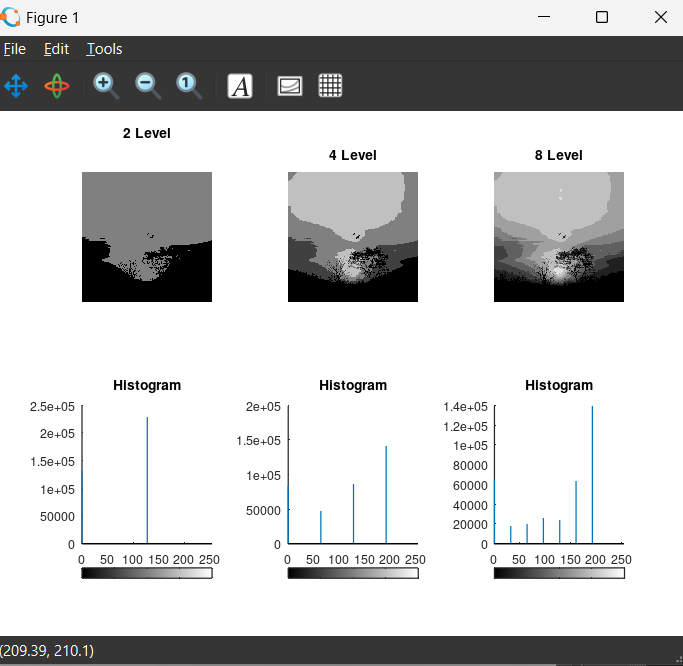
imshow(gambar\_gray);

title('Grayscale');

subplot(1, 3, 3);

imshow(gambar\_biner);

title('Citra Biner');

1. **Pemrosesan dalam grayscale atau biner adalah kompromi optimal antara kecepatan, efisiensi, dan efektivitas dalam menangkap fitur penting untuk pengawasan keamanan.**
2. **Kuantisasi citra GREYSCALE**
3. ****

**CODE OCTAV :**

pkg load image;

img = imread('D:\cp\citra2\Twirl.jpg');

gray = rgb2gray(img);

level\_list = [2, 4, 8];

figure;

for i = 1:length(level\_list);

jumlah\_level = level\_list(i);

interval = 256 / jumlah\_level;

kuantisasi = floor(double(gray) / interval) \* interval;

kuantisasi = uint8(kuantisasi);

subplot(2, 3, i);

imshow(kuantisasi);

title([num2str(jumlah\_level), ' Level']);

subplot (2,3,i+3);

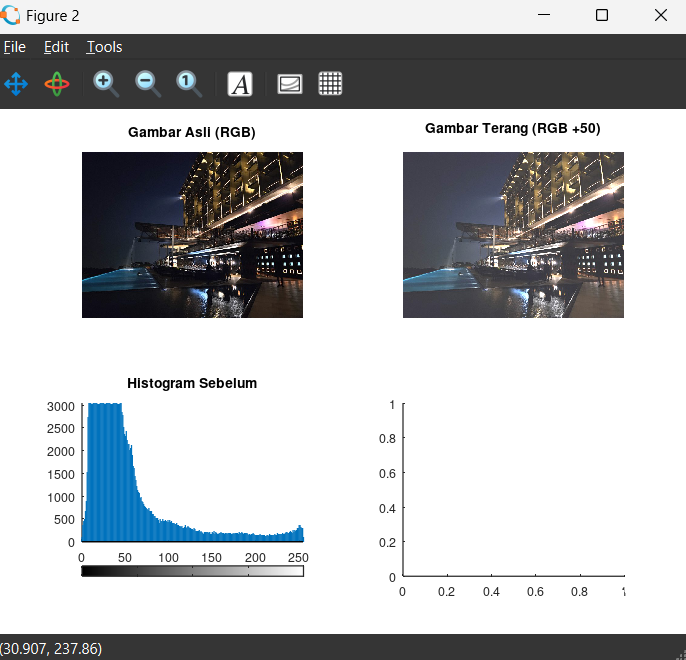
imhist(kuantisasi);

title('Histogram');

end

1. **Kuantisasi 2 Level** Menghasilkan citra yang sangat sederhana dengan dua intensitas saja (hitam dan putih), sehingga detail objek sulit terlihat. **Kuantisasi 4 Level** Memberikan sedikit peningkatan kualitas. Tepi objek lebih jelas, tetapi detail dalam objek masih kurang.

**Kuantisasi 8 Level** Mampu menampilkan lebih banyak gradasi keabuan, sehingga objek terlihat lebih detail dan bentuknya lebih jelas.

1. **Kuantisasi membantu dengan mengurangi detail warna yang tidak penting, sehingga ukuran gambar jadi kecil dan lebih cepat dikirim, sangat cocok untuk kondisi jaringan yang lambat seperti saat menggunakan WhatsApp.**
2. **Peningkatan Cahaya**
3. 

**CODE OCTAV :**

pkg load image;

gambar = imread('D:\cp\citra2\MinPencahaya.jpg');

% Menambah kecerahan pada semua channel RGB

kenaikan = 50;

gambar\_terang = uint8(double(gambar) + kenaikan); % Menambahkan nilai

gambar\_terang(gambar\_terang > 255) = 255; % Membatasi agar tidak melebihi 255

% Menampilkan gambar sebelum dan sesudah

figure;

subplot(2, 2, 1);

imshow(gambar);

title('Gambar Asli (RGB)');

subplot(2, 2, 2);

imshow(gambar\_terang);

title(['Gambar Terang (RGB +', num2str(kenaikan), ')']);

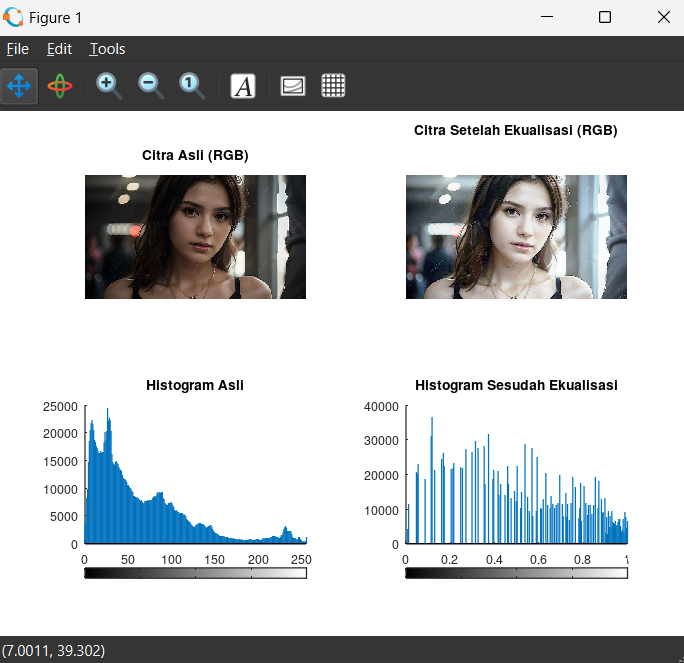
% Menampilkan histogram masing-masing channel sebelum

subplot(2, 2, 3);

imhist(rgb2gray(gambar));

title('Histogram Sebelum');

subplot(2, 2, 4)

1. **Jika hanya menggunakan operasi piksel di citra gelap, maka hasil pengenalan objek sangat tidak akurat, karena kurangnya kontras, dominasi noise, dan tidak adanya konteks spasial**
2. **Ekualisasi Histogram**
3. ****

**CODE OCTAV :**

pkg load image;

gambar = imread('D:\cp\citra2\FotoDlmRuang.jpg');

% Memisahkan channel warna

R = gambar(:, :, 1);

G = gambar(:, :, 2);

B = gambar(:, :, 3);

% Ekualisasi histogram pada setiap channel

R\_eq = histeq(R);

G\_eq = histeq(G);

B\_eq = histeq(B);

% Menggabungkan kembali jadi citra RGB

gambar\_eq\_rgb = cat(3, R\_eq, G\_eq, B\_eq);

% Menampilkan hasil sebelum dan sesudah

figure;

subplot(2, 2, 1);

imshow(gambar);

title('Citra Asli (RGB)');

subplot(2, 2, 2);

imshow(gambar\_eq\_rgb);

title('Citra Setelah Ekualisasi (RGB)');

% Menampilkan histogram

subplot(2, 2, 3);

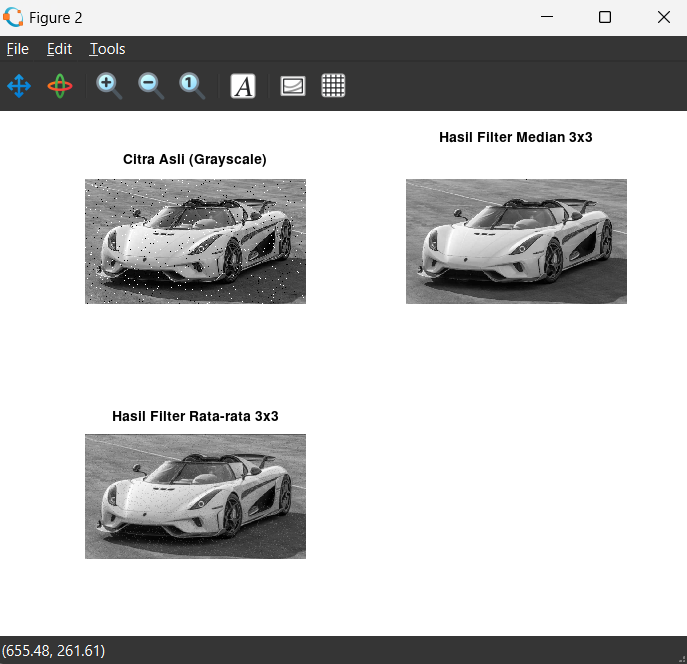
imhist(gambar);

title('Histogram Asli');

subplot(2, 2, 4);

imhist(gambar\_eq\_rgb);

title('Histogram Sesudah Ekualisasi');

1. **Ekualisasi histogram membuat fitur wajah lebih kontras dan konsisten, meningkatkan keakuratan dan ketahanan sistem pengenalan wajah terhadap pencahayaan yang bervariasi. Maka itu, teknik ini sangat umum digunakan sebagai pra-pemrosesan standar.**
2. **Filter Median dan Mean**
3. ****

**CODE OCTAV :**

pkg load image;

gambar = imread('D:\cp\citra2\Salt\_and\_Pepper\_Noise\_koenigsegg.png'); % Ganti dengan nama file kamu

% Mengubah ke grayscale jika masih RGB

if ndims(gambar) == 3

gambar\_gray = rgb2gray(gambar);

else

gambar\_gray = gambar;

end

% Filter median 3x3

hasil\_median = medfilt2(gambar\_gray, [3 3]);

% Filter rata-rata (mean) 3x3

kernel\_mean = ones(3, 3) / 9;

hasil\_mean = imfilter(gambar\_gray, kernel\_mean);

% Menampilkan hasil perbandingan dalam satu figure

figure;

subplot(2, 2, 1);

imshow(gambar\_gray);

title('Citra Asli (Grayscale)');

subplot(2, 2, 2);

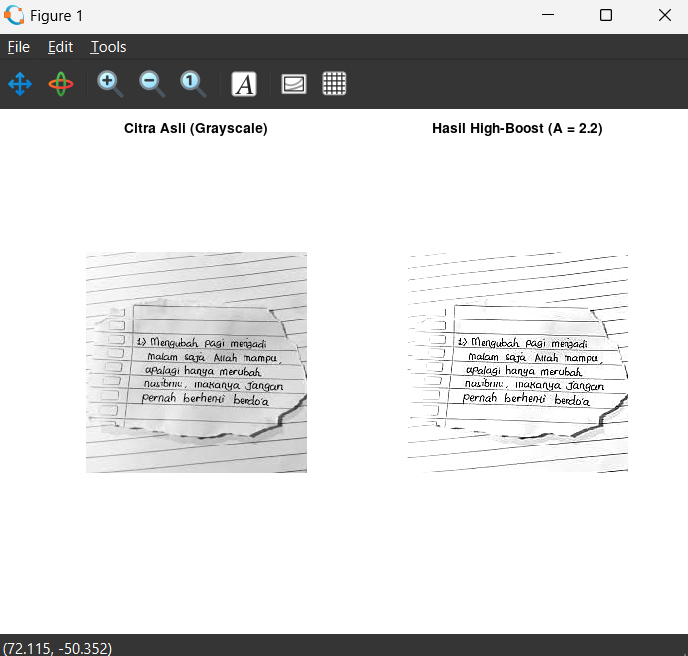
imshow(hasil\_median);

title('Hasil Filter Median 3x3');

subplot(2, 2, 3);

imshow(hasil\_mean);

title('Hasil Filter Rata-rata 3x3');

1. **Filter median cocok untuk mempertahankan bentuk dan struktur objek, sedangkan filter mean lebih baik untuk mengurangi noise secara menyeluruh tapi mengorbankan detail.**
2. **Filter median sangat disukai dalam pemrosesan rontgen karena kemampuannya menghilangkan noise secara efektif tanpa mengorbankan detail anatomi penting, yang sangat dibutuhkan dalam analisis medis yang akurat.**
3. **High-Boost**
4. ****

**CODE OCTAV :**

gambar = imread('D:\cp\citra2\Kertas.jpeg');

% Mengubah ke grayscale jika berwarna

if ndims(gambar) == 3

gambar\_gray = rgb2gray(gambar);

else

gambar\_gray = gambar;

end

% Menerapkan filter rata-rata (Low-Pass) 3x3

h = ones(3, 3) / 9;

blurred = imfilter(double(gambar\_gray), h);

% Menentukan faktor boosting (A > 1, misal A = 1.5 atau 2)

A = 2.2;

% High-Boost Filtering: A\*original - blurred

high\_boost = A \* double(gambar\_gray) - blurred;

% Konversi hasil ke uint8 agar bisa ditampilkan

high\_boost\_uint8 = uint8(high\_boost);

% Menampilkan hasil

figure;

subplot(1, 2, 1);

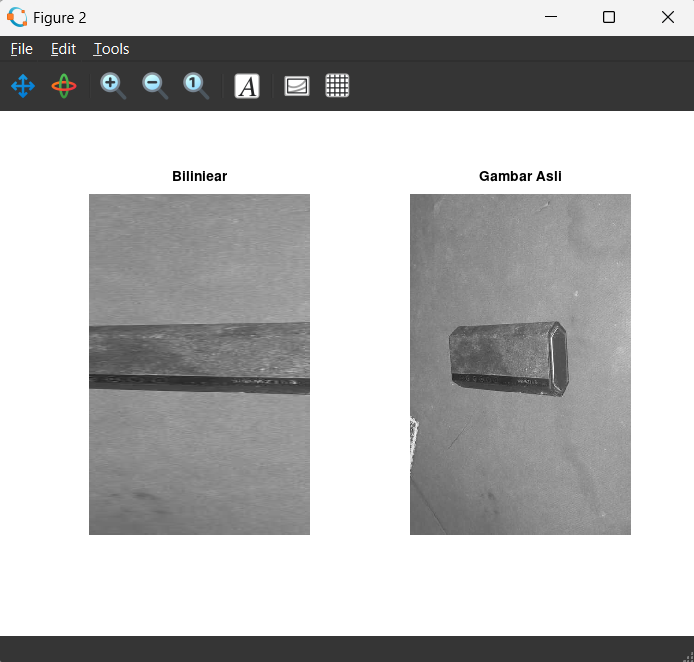
imshow(gambar\_gray);

title('Citra Asli (Grayscale)');

subplot(1, 2, 2);

imshow(high\_boost\_uint8);

title(['Hasil High-Boost (A = ' num2str(A) ')']);

1. **Filter median membantu OCR dengan membersihkan noise tanpa mengaburkan bentuk huruf, sehingga proses ekstraksi karakter menjadi lebih akurat dan andal.**
2. **Interpolasi Bilinier dan Perbesar citra**
3. ****

**CODE OCTAV :**

clc;

berkas =('D:\cp\citra2\MiringAngel.jpg');

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 = melengkkung horizontal

##a4 = geser kanan kiri (pixel)

##b1 = miring atas bawah

##b2 = memanjang keatas

##b3 = melengkung vertikal

##b4 = geser atas bawah (pixel)

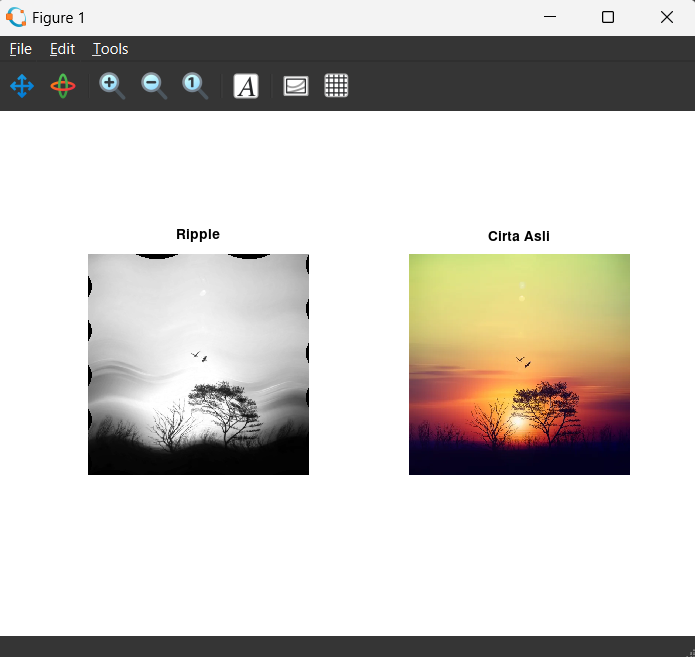
% fungsi bilinier

G = tbilin(F, 0.3,0,0,140, 0,1,0,0); % fungsi biliniear

figure;

subplot (1,2,1); imshow (G); title ('Biliniear');

subplot (1,2,2); imshow (img); title ('Gambar Asli');

1. **Rotasi penting untuk memastikan orientasi dokumen benar agar sistem bisa mengenali teks dan layout dengan akurat, sedangkan perbesaran penting untuk meningkatkan keterbacaan dan akurasi pengenalan, terutama pada bagian yang kecil atau kabur.**
2. **Penerapan efek Ripple dan Twirl**
3. ****

**CODE OCTAV :**

clc;

F = imread('D:\cp\citra2\Twirl.jpg');

function G = ripple(F, ax, ay, tx, ty)

% RIPPLE Berfungsi untuk melakukan transformasi 'ripple'.

dimensi = size(F);

tinggi = dimensi(1);

lebar = dimensi(2);

for y=1 : tinggi

for x=1 : lebar

x2 = x + ax \* sin(2 \* pi \* y / tx);

y2 = y + ay \* sin(2 \* pi \* x / ty);

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 (floor(x2)==lebar) || ...

(floor(y2) == tinggi)

G(y, x) = F(floor(y2), floor(x2));

else

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;

end

else

G(y, x) = 0;

end

end

end

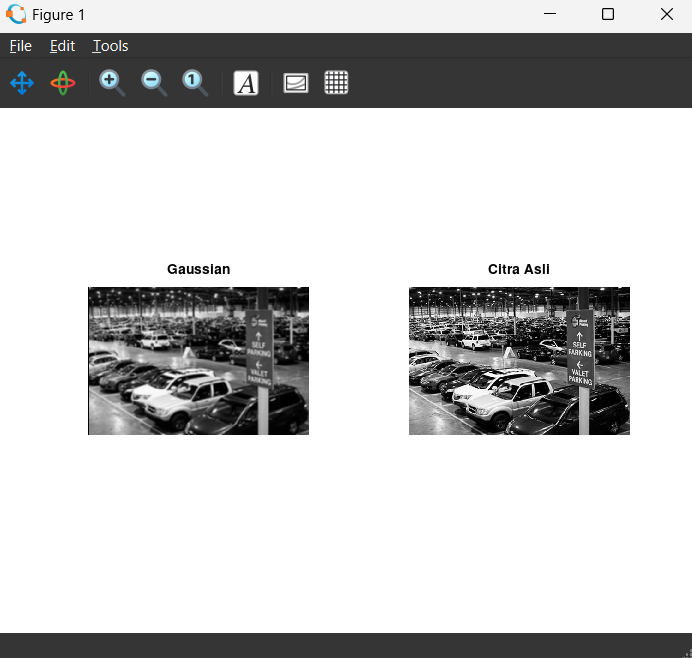
G = uint8(G);

end

G = ripple(F,10,15,120, 250);

subplot(1,2,1); imshow(G); title('Ripple');

subplot(1,2,2); imshow(F); title('Cirta Asli');

1. **Ripple dan twirl punya potensi tinggi dalam seni digital dan media sosial karena mereka mampu menciptakan efek visual yang ekspresif, simbolik, dan menarik perhatian, baik untuk tujuan artistik maupun viralitas konten**.
2. **Peneraoan filter Gaussian**
3. ****

**CODE OCTAV :**

clc;

F = imread('D:\cp\citra2\Parkiran.jpg');

F = rgb2gray(F);

kernel\_size = 5; % ukuran kernel Gaussian harus ganjil

sigma = 2.0; % standar deviasi Gaussian

function [G] = konvolusi\_gaussian(F, kernel\_size, sigma)

% kernel Gaussian 1D

m2 = floor(kernel\_size / 2);

x = -m2:m2;

H = exp(-(x.^2) / (2\*sigma^2));

H = H / sum(H); % Normalisasi

Hkol = H; % Filter vertikal (1D horizontal)

Hbrs = H; % Filter horizontal (1D horizontal)

[tinggi\_f, lebar\_f] = size(F);

F2 = double(F);

T = F2;

% Konvolusi vertikal dengan Hkol

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 dengan Hbrs

G = zeros(size(F2)); % Inisialisasi output

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

G(y, x) = jum;

end

end

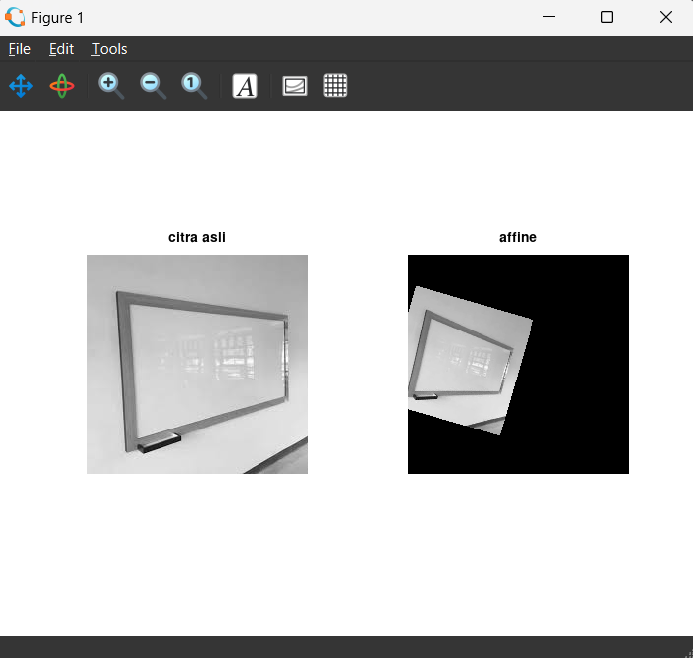
G = uint8(G);

end

G = konvolusi\_gaussian(F, kernel\_size, sigma);

subplot(1,2,1); imshow(G); title('Gaussian');

subplot(1,2,2); imshow(F); title('Citra Asli');

1. **Gaussian filter memberikan solusi yang efektif, halus, dan profesional untuk menjaga privasi individu dalam video pengawasan — baik untuk melindungi identitas, memenuhi regulasi, maupun untuk menampilkan rekaman secara publik tanpa pelanggaran hak privasi.**
2. **Penerapan transformasi Affine**
3. ****

**CODE OCTAV :**

F = rgb2gray(imread('D:\cp\citra2\papan.jpeg'));

function G = taffine(F, a11, a12, a21, a22, tx, ty)

[tinggi, lebar] = size(F);

for y=1 : tinggi

for x=1 : lebar

x2 = a11 \* x + a12 \* y + tx;

y2 = a21 \* x + a22 \* y + ty;

if (x2>=1) && (x2<=lebar) && ...

(y2>=1) && (y2<=tinggi)

% interpolasi bilinear

p = floor(y2);

q = floor(x2);

a = y2-p;

b = x2-q;

if (floor(x2)==lebar) || ...

(floor(y2) == tinggi)

G(y, x) = F(floor(y2), floor(x2));

else

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;

end

else

G(y, x) = 0;

end

end

end

G = uint8(G);

end

##parameter : G = taffine(F, a11, a12, a21, a22, tx, ty)

##a11, a22 = skala dan rotasi

##a12,a21 = rotasi dan shear

##tx, ty = translasi

rad = pi/6;

G = taffine(F, 2 \* cos(rad) ,sin(rad) ,-sin(rad), 2 \* cos(rad), -30,-

50);

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

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

1. Transformasi affine **menjaga struktur geometris** sambil memungkinkan perubahan posisi, orientasi, dan skala citra. Ini sangat penting untuk:

**Membuat peta yang akurat** dari citra drone

**Mengarahkan drone** ke lokasi yang benar meskipun ada perubahan sudut pandang

**Menyesuaikan data visual dengan dunia nyata**