Nama: Nikolas Adi Kurniatmaja Sijabat

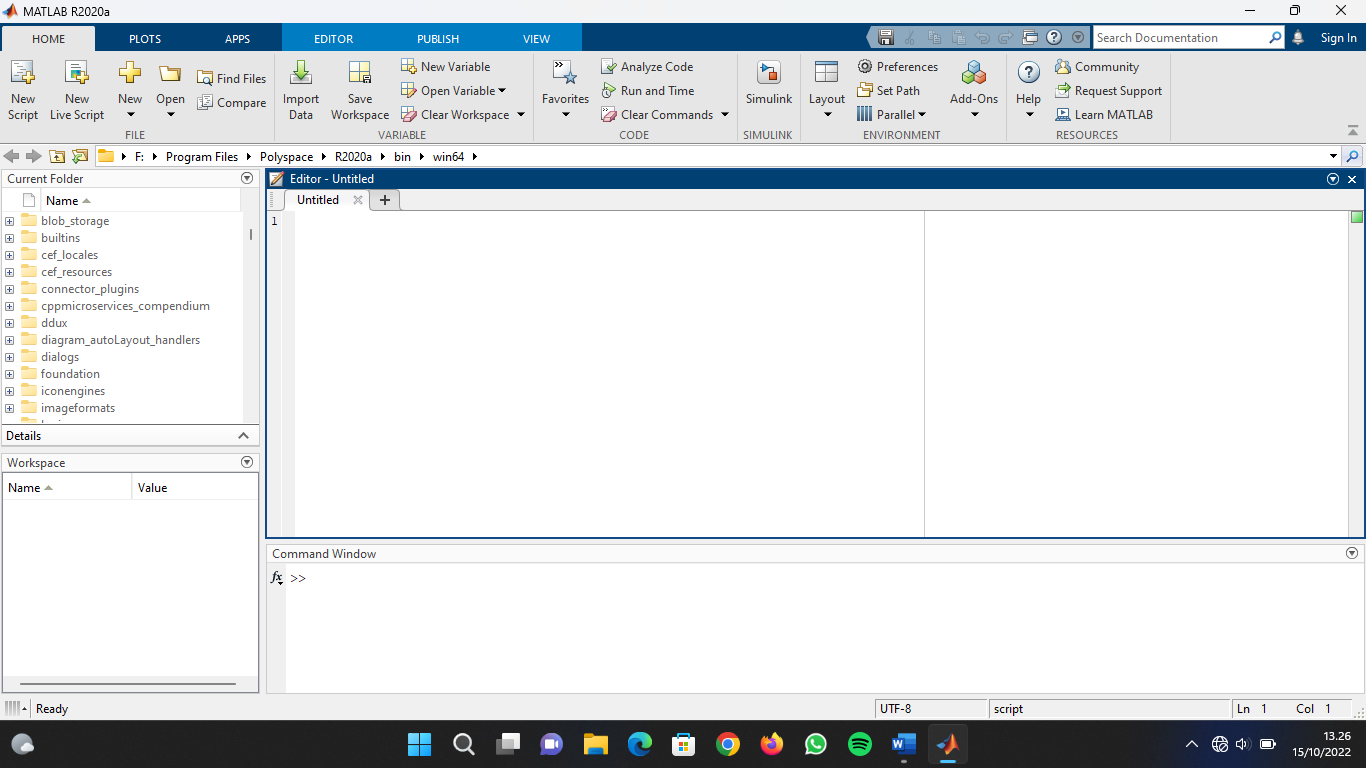
NIM: A11.2021.13800

Kelompok kelas: A11.43SC

**Laporan Program Deteksi Nominal Uang Logam dengan Metode Outsu Thresholding**

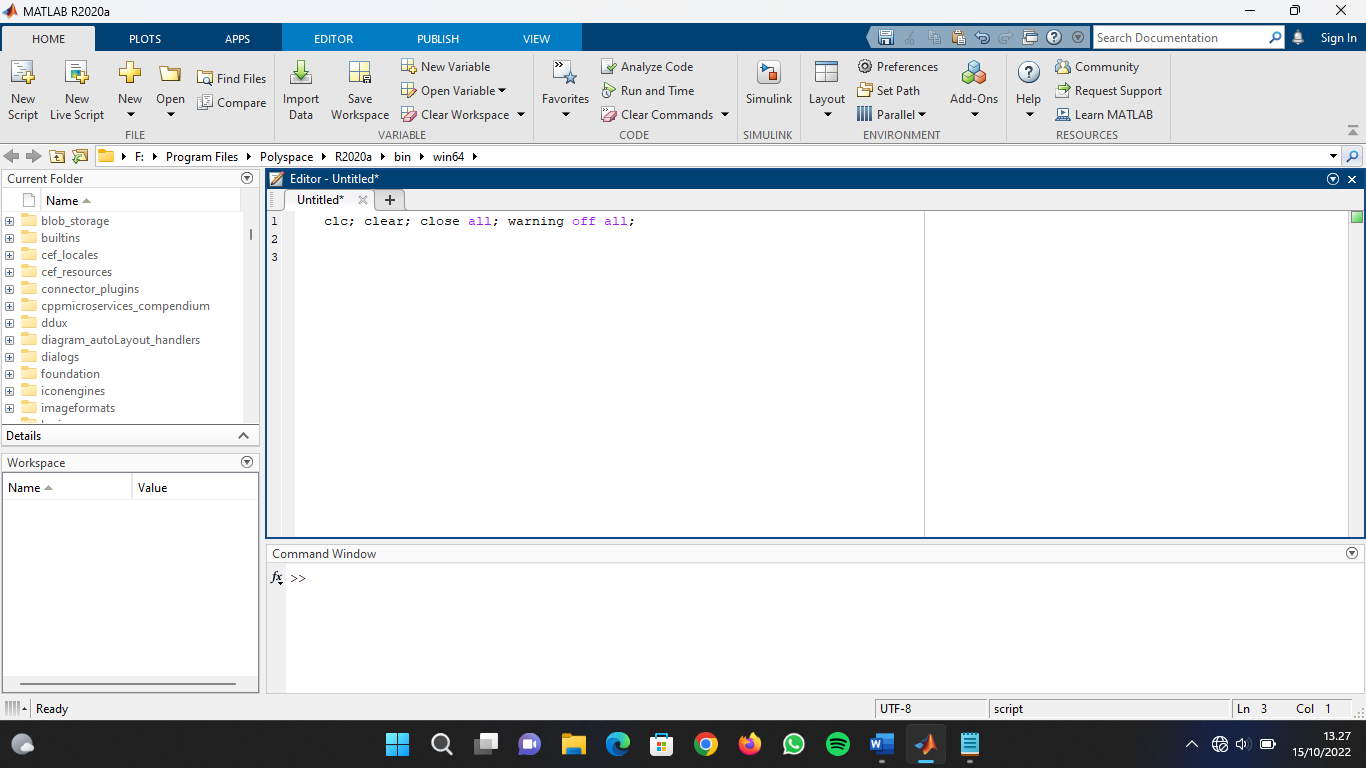
**Tahap Pembuatan Program**

1. Buat file script baru



1. Tuliskan syntax:

clc; clear; close all; warning off all;

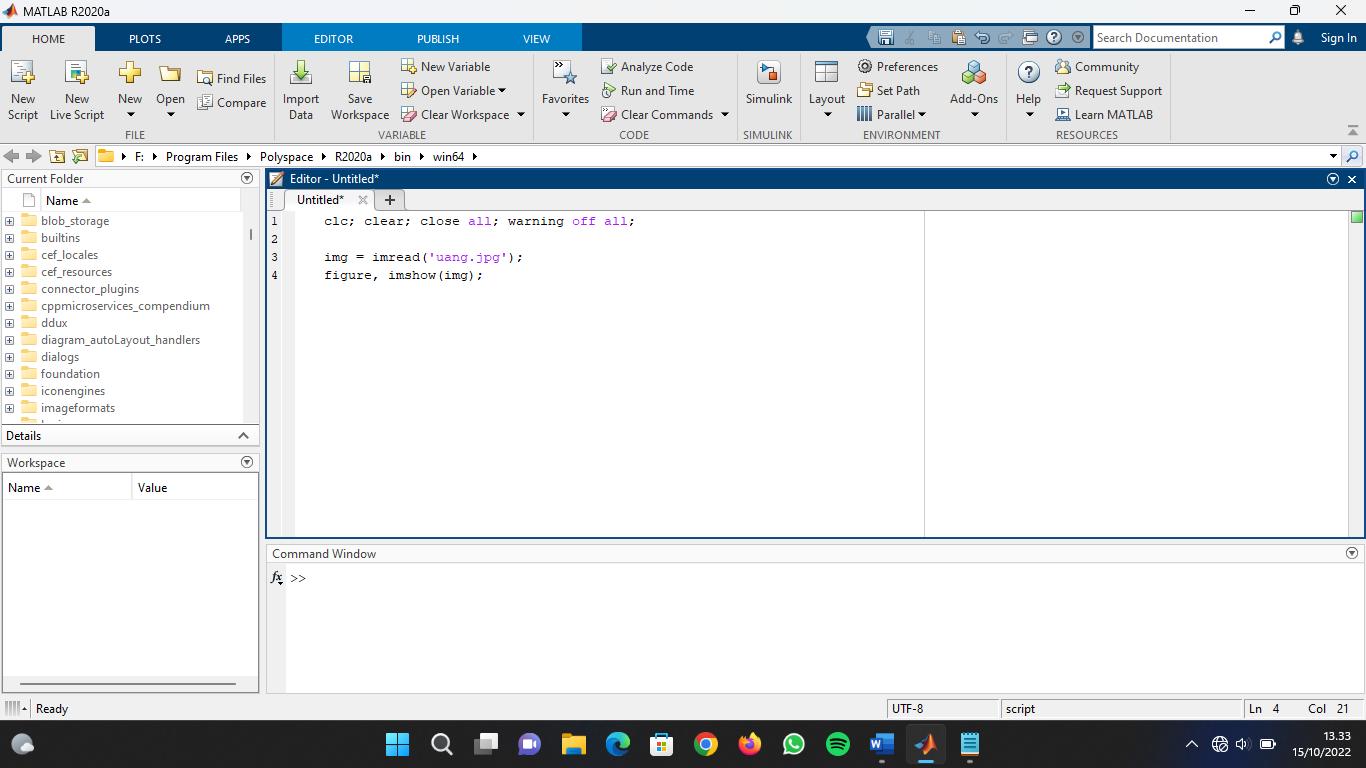


1. Tuliskan syntax:

img = imread('uang.jpg');

figure, imshow(img);

→ Untuk membaca citra gambar warna RGB, kemudian isikan nama file yang ingin di load pada bagian img = imread(‘nama file’).

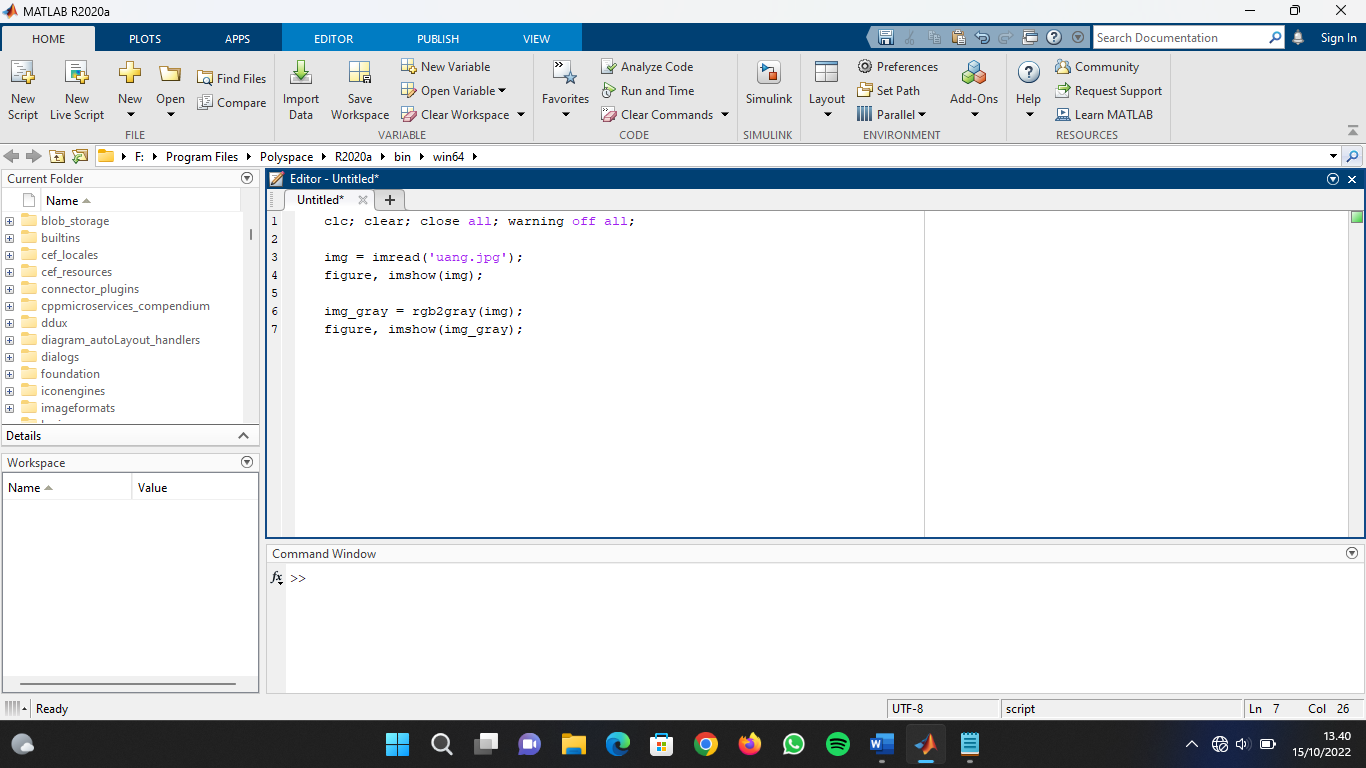


1. Tuliskan syntax:

img\_gray = rgb2gray(img);

figure, imshow(img\_gray);

→ Untuk mengonversi citra gambar warna RGB menjadi citra gambar warna Grayscale.



1. Tuliskan syntax:

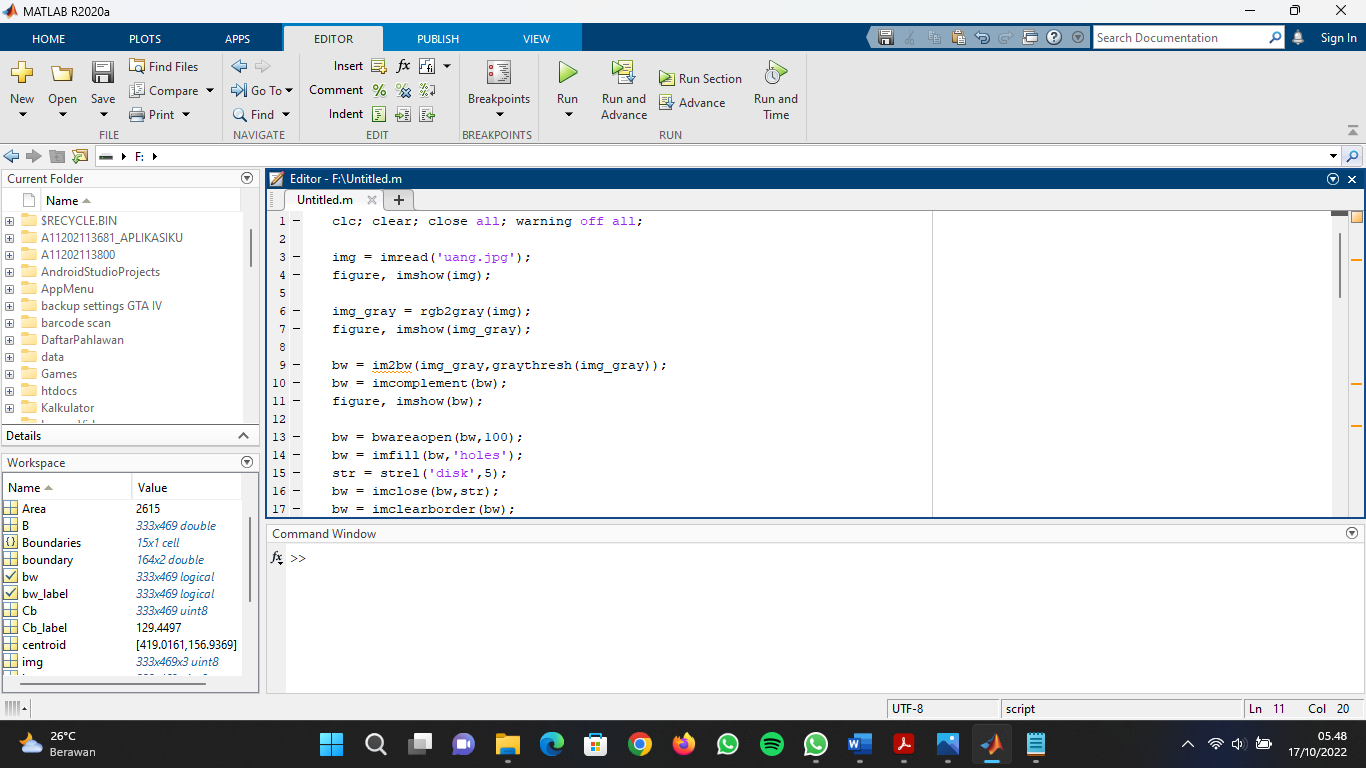
bw = im2bw(img\_gray,graythresh(img\_gray));

→ Untuk mengonversi citra gambar warna Grayscale menjadi bilangan biner.

bw = imcomplement(bw);

figure, imshow(bw);

→ Setelah citra gambar dikonversi menjadi bilangan biner, maka melalui syntax berikut akan dilakukan operasi komplemen citra agar objek bernilai satu dan background bernilai nol.



1. Setelah melakukan operasi komplemen citra, akan dilakukan operasi morfologi pada citra gambar untuk menyempurnakan hasil segmentasi. Syntax:

bw = bwareaopen(bw,100);

→ area opening untuk menghilangkan noise.

bw = imfill(bw,'holes');

→ filling holes untuk mengisi objek yang berlubang.

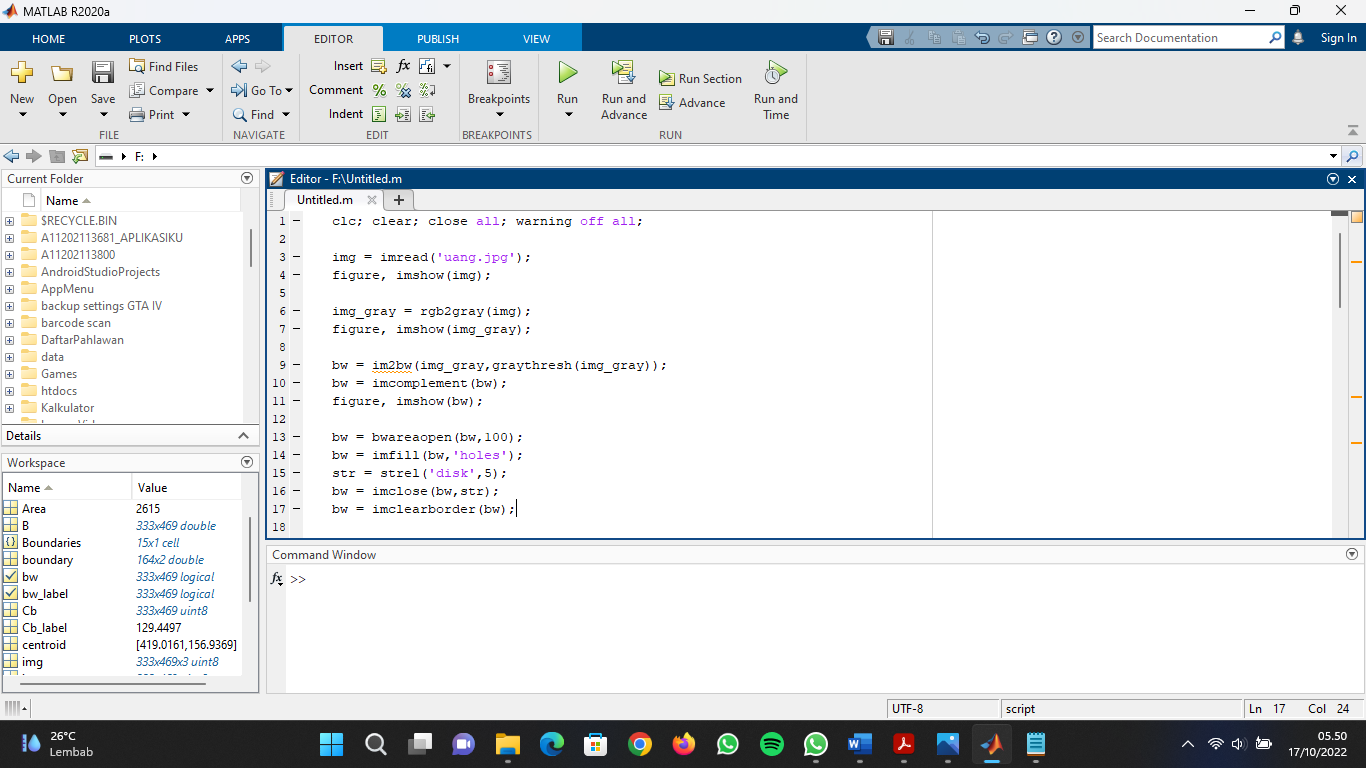
str = strel('disk',5);

bw = imclose(bw,str);

→ closing untuk membuat bentuk objek lebih smooth.

bw = imclearborder(bw);

→ menghilangkan objek yang menempel pada border (tepian citra).



1. Melakukan pelabelan terhadap masing masing objek. Syntax:

[B,L] = bwlabel(bw);

Setelah diberikan label, dilakukan penghitungan luas dan centroid objek. Syntax:

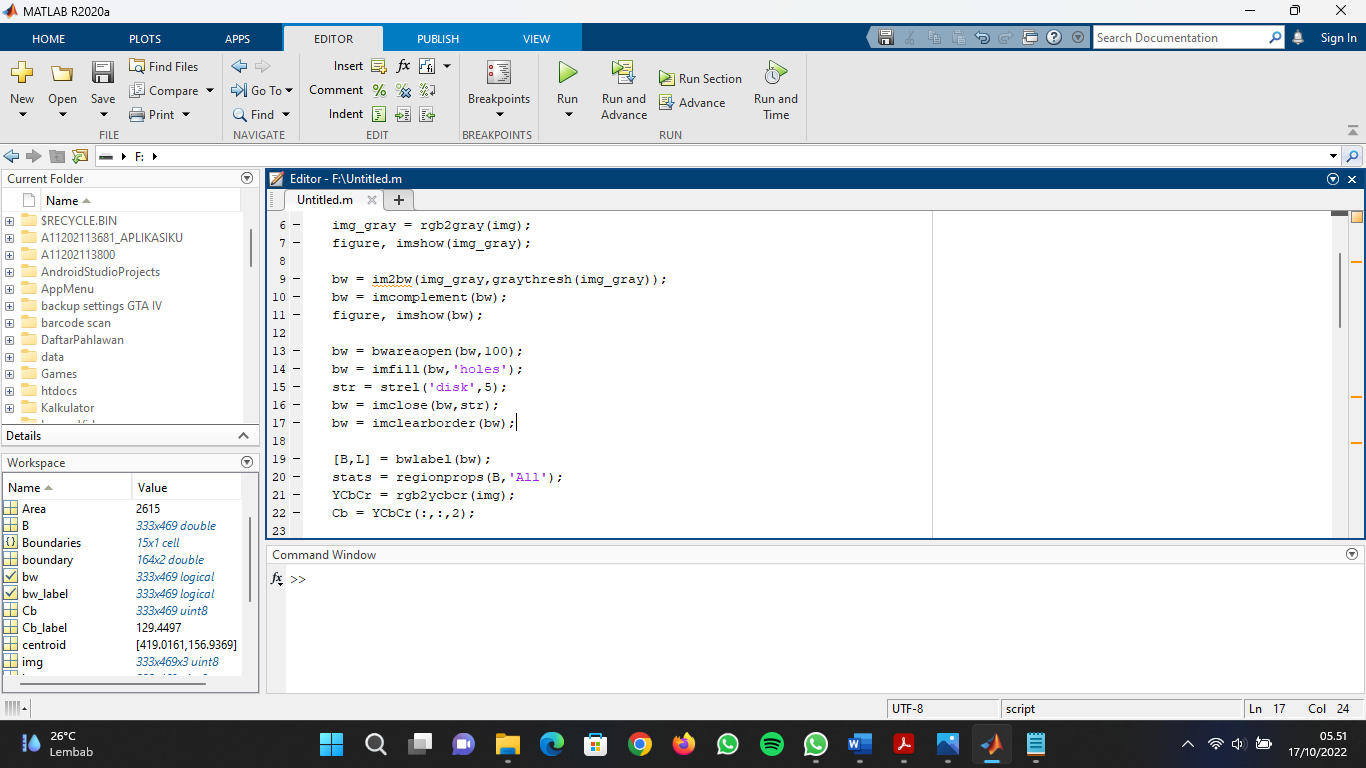
stats = regionprops(B,'All');

Kemudian citra gambar warna RGB dikonversi menjadi YcbCr. Syntax:

YCbCr = rgb2ycbcr(img);

Mengekstrak komponen Cb (Chrominance-blue). Syntax:

Cb = YCbCr(:,:,2);



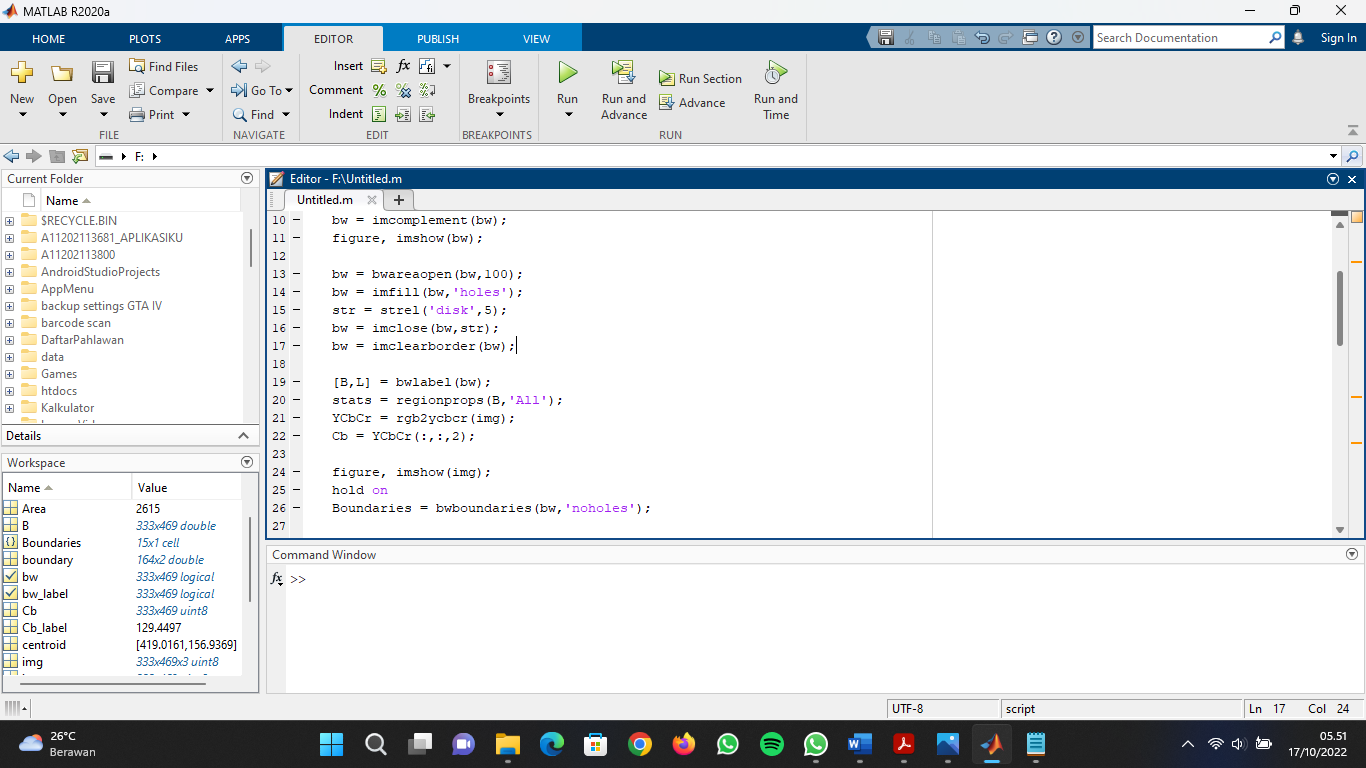
1. Menampilkan citra gambar warna RGB. Syntax:

figure, imshow(img);

hold on

Kemudian membuat boundary pada koin yang terdeteksi. Syntax:

Boundaries = bwboundaries(bw,'noholes');



1. Melakukan perhitungan pada masing-masing objek dalam citra untuk menentukan jenis koin. Syntax:

% untuk n = 1 s.d n = jumlah objek

for n = 1:L

boundary = Boundaries{n};

% menghitung nilai Cb dari masing2 objek

bw\_label = (B==n);

Cb\_label = immultiply(Cb,bw\_label);

Cb\_label = (sum(sum(Cb\_label)))/(sum(sum(bw\_label)));

% menghitung luas dan centroid masing2 objek

Area = stats(n).Area;

centroid = stats(n).Centroid;

% jika nilai Cb > 120 maka dikenali sebagai koin silver

if Cb\_label>120

% jika luas < 70000 maka dikenali sebagai koin 100

if Area<70000

nilai = 100;

% jika luas < 80000 maka dikenali sebagai koin 200

elseif Area<80000 nilai = 200; % jika luas > 80000 maka dikenali sebagai koin 500

else

nilai = 500;

end

% menampilkan boundary pada objek

plot(boundary(:,2), boundary(:,1), 'y', 'LineWidth', 4)

% menampilkan nilai koin pada centroid objek

text(centroid(1)-50,centroid(2),num2str(nilai),...

'Color','y','FontSize',20,'FontWeight','bold');

% jika nilai Cb < 120 maka dikenali sebagai koin kuning

else

% jika luas < 70000 maka dikenali sebagai koin 500

if Area<70000 nilai = 500; % jika luas > 70000 maka dikenali sebagai koin 1000

else

nilai = 1000;

end

% menampilkan boundary pada objek

plot(boundary(:,2), boundary(:,1), 'c', 'LineWidth', 4)

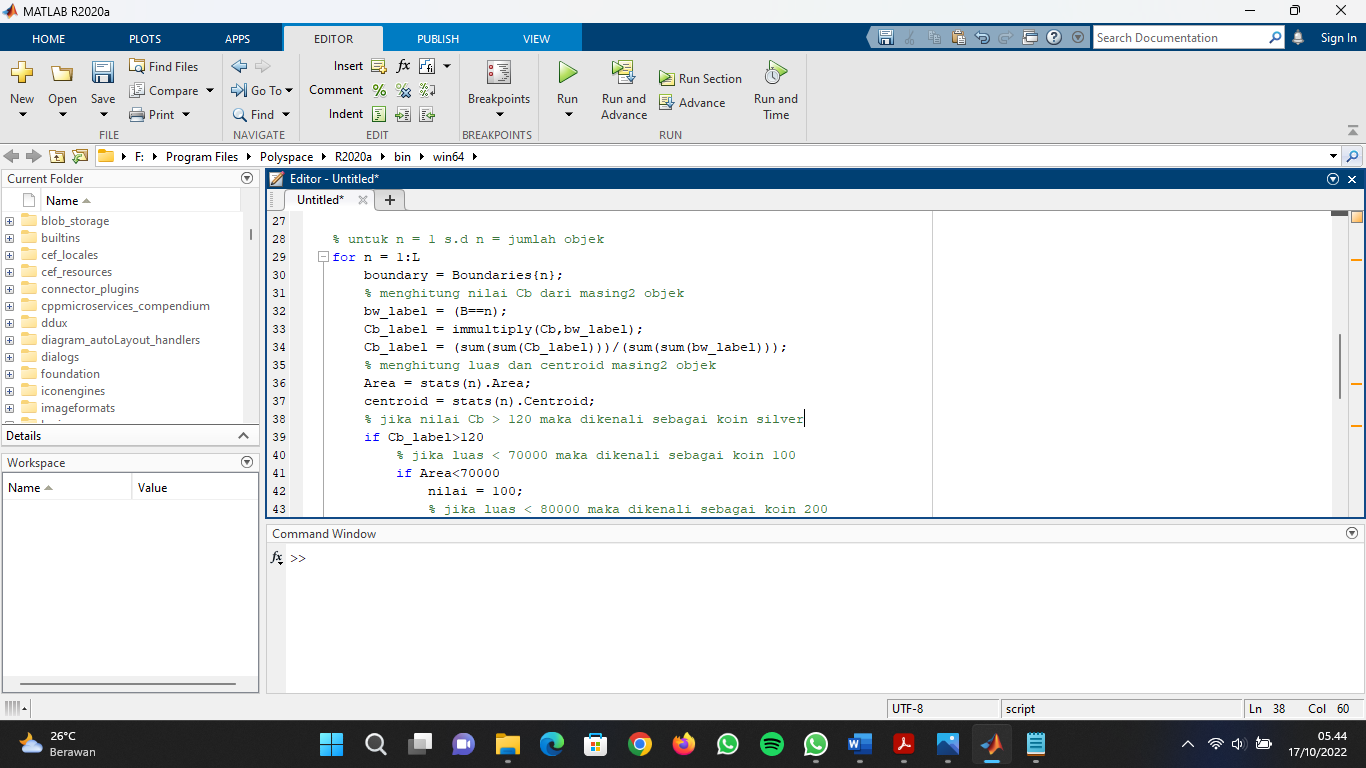
% menampilkan nilai koin pada centroid objek

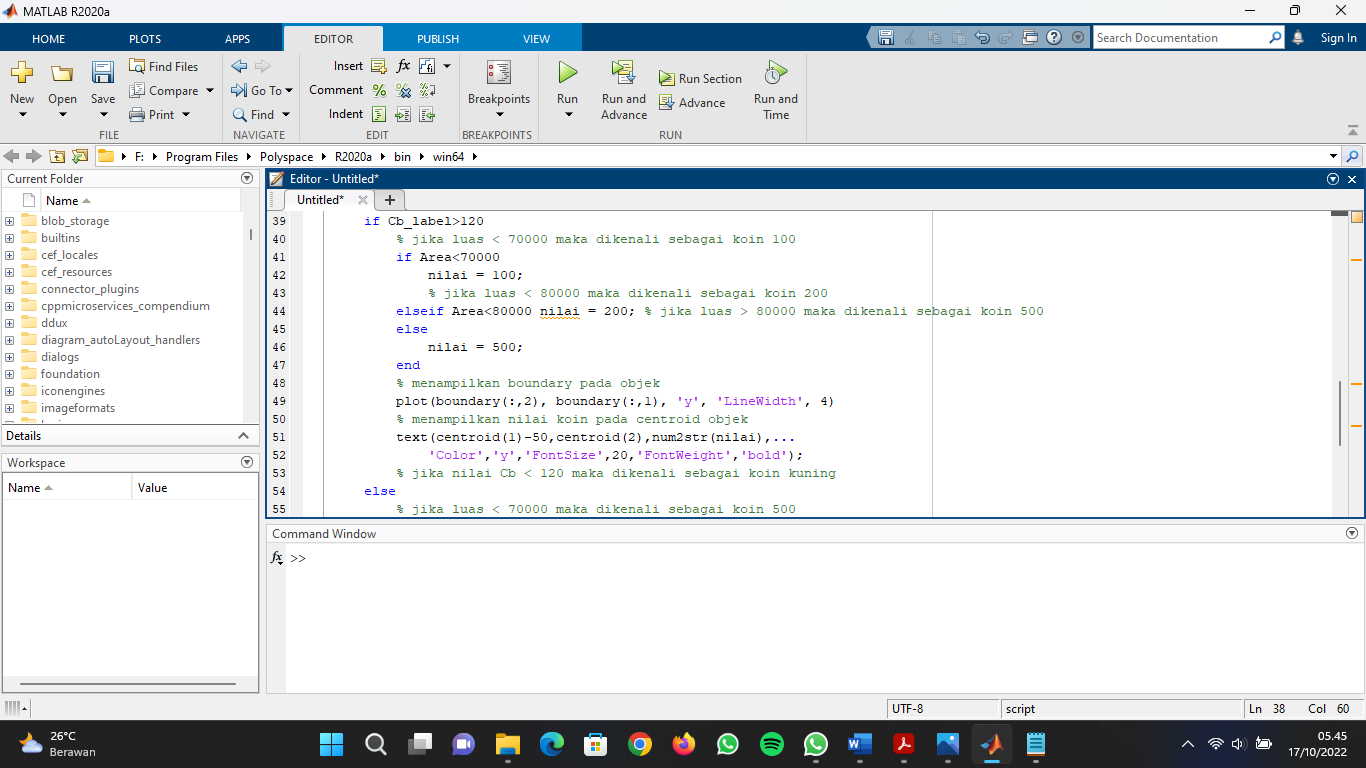
text(centroid(1)-50,centroid(2),num2str(nilai),...

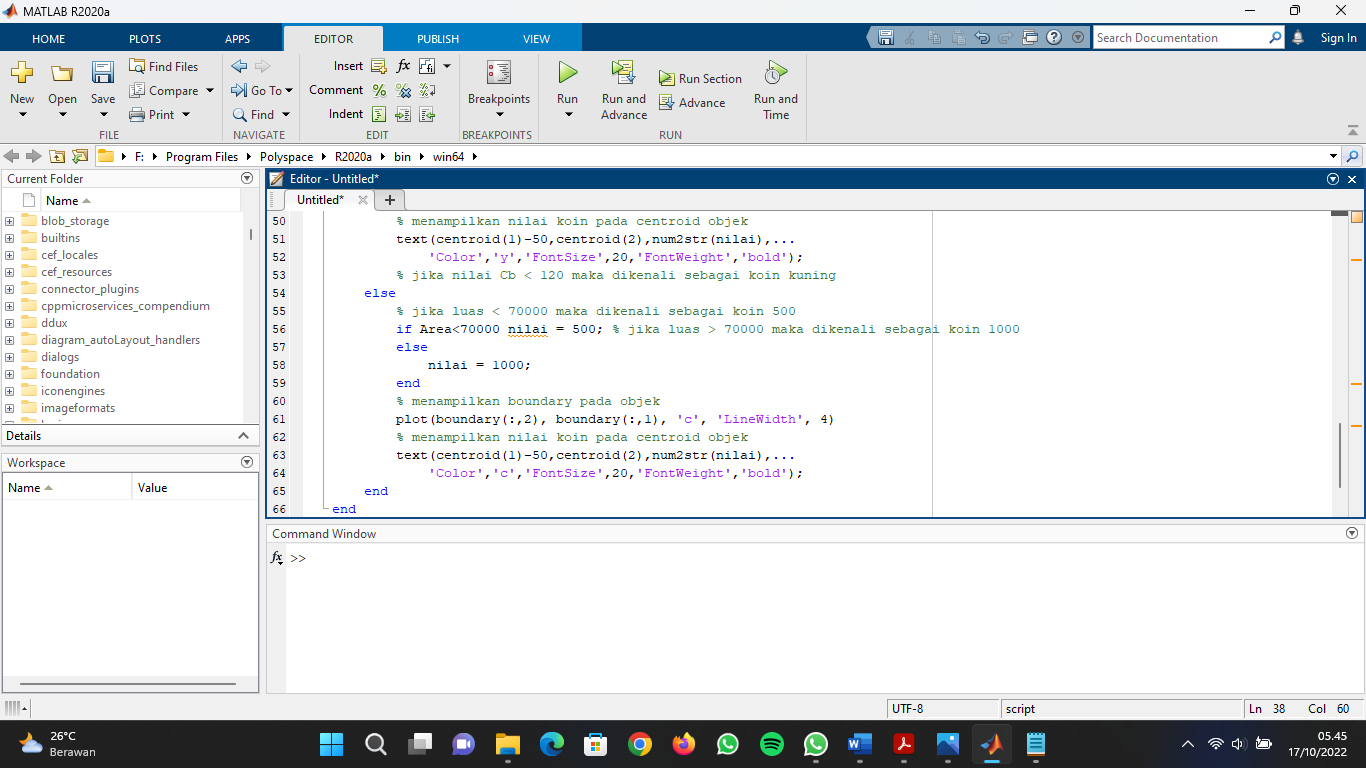
'Color','c','FontSize',20,'FontWeight','bold');

end

end





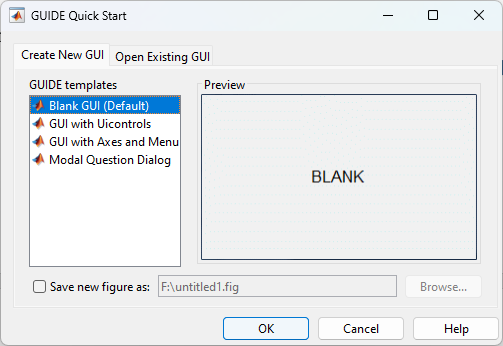


**Pembuatan GUI**

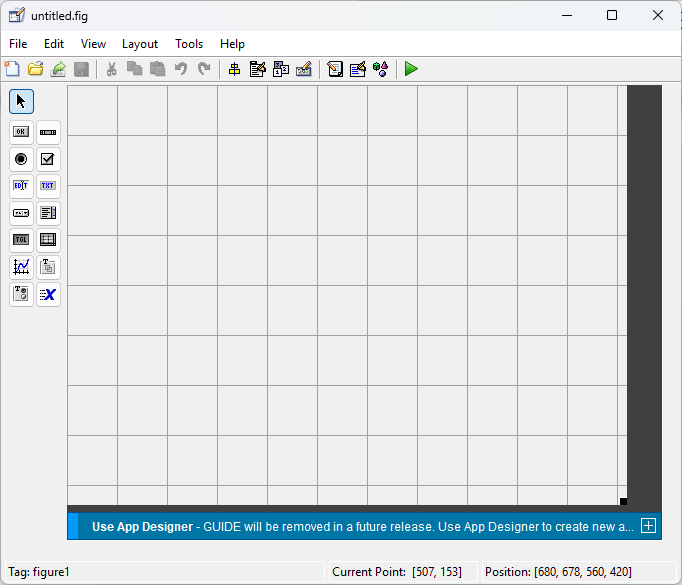
1. Ketik ‘guide’ pada bagian Command Window MATLAB



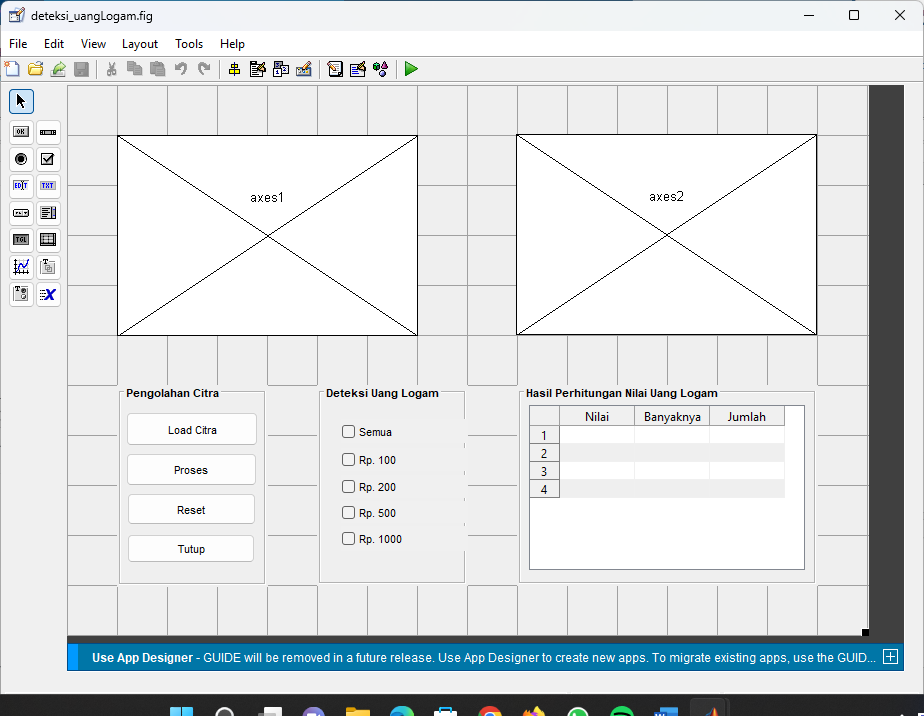
Kemudian pada jendela GUIDE Quick Start, pilih Blank GUI (Default) lalu klik OK



Setelah itu akan muncul jendela GUIDE baru untuk membuat GUI



1. Buat susunan bagian pada GUI dengan tampilan seperti dibawah ini



Dengan komposisi bagian terdiri dari:

* Axes1
* Axes2
* Uipanel1
* Pushbutton1
* Pushbutton2
* Pushbutton3
* Pushbutton4
* Uipanel2
* Checkbox1
* Checkbox2
* Checkbox3
* Checkbox4
* Checkbox5
* Uipanel3
* Uitable1

1. Untuk membuat bagian axes (axes1 dan axes2), klik tombol Axes. Axes digunakan untuk menampilkan citra gambar.



1. Untuk membuat bagian panel untuk tempat menampung tombol atau tabel, klik tombol Panel.



1. Untuk membuat bagian tombol di dalam panel, klik tombol Push Button. Push Button digunakan untuk menampung perintah yang akan dijalankan dalam program.



1. Untuk membuat bagian ceklis pilihan di dalam panel, klik tombol Check Box. Check Box digunakan untuk memilih pilihan hasil deteksi yang akan ditampilkan



1. Untuk membuat tabel hasil deteksi koin, klik tombol Table. Table digunakan untuk menampung hasil nominal deteksi koin.



1. Setelah membuat desain GUI, masukkan syntax pada setiap bagian satu per satu dengan cara klik kanan pada bagian yang ingin diisi, masuk ke bagian View Callbacks lalu klik Callback.
2. Bagian pushbutton1

function pushbutton1\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% menampilkan menu browse file

[filename, pathname] = uigetfile('\*.jpg');

% jika ada file yang dipilih maka akan menjalankan perintah di bawahnya

if ~isequal(filename,0)

% mereset button2

set(handles.pushbutton2,'Enable','on')

set(handles.uitable1,'Data',[])

axes(handles.axes1)

cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

axes(handles.axes2)

cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

% membaca citra rgb

Img = imread(fullfile(pathname,filename));

% menampilkan citra pada axes

axes(handles.axes1)

imshow(Img)

title('Citra Uang Logam','FontName','Eras Demi ITC')

% menyimpan variabel Img pada lokasi handles

% (lokasi penyimpanan variabel) agar dapat

% dipanggil oleh pushbutton yang lain

handles.Img = Img;

guidata(hObject, handles)

else

% jika tidak ada file yang dipilih maka akan kembali

return

end

1. Bagian pushbutton2

function pushbutton2\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% memanggil variabel Img yang ada di lokasi handles

Img = handles.Img;

% mengkonversi citra rgb menjadi grayscale

Img\_gray = rgb2gray(Img);

% mengkonversi citra grayscale menjadi biner

bw = im2bw(Img\_gray,graythresh(Img\_gray));

% melakukan komplemen citra agar objek bernilai satu

% dan background bernilai nol

bw = imcomplement(bw);

% operasi morfologi untuk menyempurnakan hasil segmentasi

% 1. area opening untuk menghilangkan noise

bw = bwareaopen(bw,100);

% 2. filling holes untuk mengisi objek yang berlubang

bw = imfill(bw,'holes');

% 3. closing untuk membuat bentuk objek lebih smooth

str = strel('disk',5);

bw = imclose(bw,str);

% 4. menghilangkan objek yang menempel pada border (tepian citra)

bw = imclearborder(bw);

% pelabelan terhadap masing2 objek

[B,L] = bwlabel(bw);

% menghitung luas dan centroid objek

stats = regionprops(B,'All');

% mengkonversi citra rgb menjadi YCbCr

YCbCr = rgb2ycbcr(Img);

% mengekstrak komponen Cb (Chrominance-blue)

Cb = YCbCr(:,:,2);

% menampilkan citra rgb hasil identifikasi pada axes

axes(handles.axes2)

imshow(Img);

title('Hasil Identifikasi','FontName','Eras Demi ITC')

hold on

% membaca value dari masing2 checkbox

val1 = get(handles.checkbox1,'Value');

val2 = get(handles.checkbox2,'Value');

val3 = get(handles.checkbox3,'Value');

val4 = get(handles.checkbox4,'Value');

val5 = get(handles.checkbox5,'Value');

% jika deteksi dilakukan pada semua uang logam

if val1==1

% menginisialisasi variabel data\_koin

data\_koin = zeros(L,1);

% membuat boundary pada koin yang terdeteksi

Boundaries = bwboundaries(bw,'noholes');

% untuk n = 1 s.d n = jumlah objek

for n = 1:L

boundary = Boundaries{n};

% menghitung nilai Cb dari masing2 objek

bw\_label = (B==n);

Cb\_label = immultiply(Cb,bw\_label);

Cb\_label = (sum(sum(Cb\_label)))/(sum(sum(bw\_label)));

% menghitung luas dan centroid masing2 objek

Area = stats(n).Area;

centroid = stats(n).Centroid;

% jika nilai Cb > 120 maka dikenali sebagai koin silver

if Cb\_label>120

% jika luas < 70000 maka dikenali sebagai koin Rp. 100

if Area<70000

nilai = 100;

% jika luas < 80000 maka dikenali sebagai koin Rp. 200

elseif Area<80000 nilai = 200; % jika luas > 80000 maka dikenali sebagai koin Rp. 500

else

nilai = 500;

end

% menampilkan boundary pada objek

plot(boundary(:,2), boundary(:,1), 'y', 'LineWidth', 4)

% menampilkan nilai koin pada centroid objek

text(centroid(1)-100,centroid(2),num2str(nilai),...

'Color','y','FontSize',20,'FontWeight','bold');

% jika nilai Cb < 120 maka dikenali sebagai koin kuning

else

% jika luas < 70000 maka dikenali sebagai koin Rp. 500

if Area<70000 nilai = 500; % jika luas > 70000 maka dikenali sebagai koin Rp. 1000

else

nilai = 1000;

end

% menampilkan boundary pada objek

plot(boundary(:,2), boundary(:,1), 'c', 'LineWidth', 4)

% menampilkan nilai koin pada centroid objek

text(centroid(1)-100,centroid(2),num2str(nilai),...

'Color','c','FontSize',20,'FontWeight','bold');

end

% mengisi variabel data\_koin dengan nilai koin

data\_koin(n) = nilai;

end

% menghitung banyaknya masing2 koin berdasarkan nilainya

[~,n\_100] = find(data\_koin==100);

nilai\_100 = numel(n\_100);

[~,n\_200] = find(data\_koin==200);

nilai\_200 = numel(n\_200);

[~,n\_500] = find(data\_koin==500);

nilai\_500 = numel(n\_500);

[~,n\_1000] = find(data\_koin==1000);

nilai\_1000 = numel(n\_1000);

% menghitung nilai total koin

nilai\_total = nilai\_100+nilai\_200+nilai\_500+nilai\_1000;

% menghitung jumlah nilai koin

jumlah\_100 = nilai\_100\*100;

jumlah\_200 = nilai\_200\*200;

jumlah\_500 = nilai\_500\*500;

jumlah\_1000 = nilai\_1000\*1000;

jumlah\_total = jumlah\_100+jumlah\_200+jumlah\_500+jumlah\_1000;

% menginisialisasi variabel cell\_koin

cell\_koin = cell(5,3);

% mengisi variabel cell\_koin dengan data2 koin

cell\_koin{1,1} = 'Rp. 100';

cell\_koin{2,1} = 'Rp. 200';

cell\_koin{3,1} = 'Rp. 500';

cell\_koin{4,1} = 'Rp. 1000';

cell\_koin{5,1} = 'Total';

cell\_koin{1,2} = num2str(nilai\_100);

cell\_koin{2,2} = num2str(nilai\_200);

cell\_koin{3,2} = num2str(nilai\_500);

cell\_koin{4,2} = num2str(nilai\_1000);

cell\_koin{5,2} = num2str(nilai\_total);

cell\_koin{1,3} = ['Rp. ',num2str(jumlah\_100)];

cell\_koin{2,3} = ['Rp. ',num2str(jumlah\_200)];

cell\_koin{3,3} = ['Rp. ',num2str(jumlah\_500)];

cell\_koin{4,3} = ['Rp. ',num2str(jumlah\_1000)];

cell\_koin{5,3} = ['Rp. ',num2str(jumlah\_total)];

% menampilkan cell\_koin pada tabel

set(handles.uitable1,'Data',cell\_koin,...

'RowName',1:4)

% jika deteksi dilakukan hanya pada uang logam Rp. 100

elseif val2==1

% menginisialisasi variabel data\_koin

data\_koin = 0;

% membuat boundary pada koin yang terdeteksi

Boundaries = bwboundaries(bw,'noholes');

% untuk n = 1 s.d n = jumlah objek

for n = 1:L

% menghitung nilai Cb dari masing2 objek

bw\_label = (B==n);

Cb\_label = immultiply(Cb,bw\_label);

Cb\_label = (sum(sum(Cb\_label)))/(sum(sum(bw\_label)));

% menghitung luas dan centroid masing2 objek

Area = stats(n).Area;

centroid = stats(n).Centroid;

% jika nilai Cb > 120 maka dikenali sebagai koin silver

if Cb\_label>120

% jika luas < 70000 maka dikenali sebagai koin Rp. 100

if Area<70000 % menampilkan boundary pada objek boundary = Boundaries{n}; plot(boundary(:,2), boundary(:,1), 'y', 'LineWidth', 4) nilai = 100; % menampilkan nilai koin pada centroid objek text(centroid(1)-100,centroid(2),num2str(nilai),... 'Color','y','FontSize',20,'FontWeight','bold'); % mengisi variabel data\_koin dengan nilai koin data\_koin = [data\_koin;nilai]; end end end % menghitung banyaknya koin Rp. 100 [~,n\_100] = find(data\_koin==100); nilai\_100 = numel(n\_100); % menghitung jumlah nilai koin Rp. 100 jumlah\_100 = nilai\_100\*100; % menginisialisasi variabel cell\_koin cell\_koin = cell(1,3); % mengisi variabel cell\_koin dengan data2 koin cell\_koin{1,1} = 'Rp. 100'; cell\_koin{1,2} = num2str(nilai\_100); cell\_koin{1,3} = ['Rp. ',num2str(jumlah\_100)]; % menampilkan cell\_koin pada tabel set(handles.uitable1,'Data',cell\_koin,... 'RowName',1) % jika deteksi dilakukan hanya pada uang logam Rp. 200 elseif val3==1 % menginisialisasi variabel data\_koin data\_koin = 0; % membuat boundary pada koin yang terdeteksi Boundaries = bwboundaries(bw,'noholes'); % untuk n = 1 s.d n = jumlah objek for n = 1:L boundary = Boundaries{n}; % menghitung nilai Cb dari masing2 objek bw\_label = (B==n); Cb\_label = immultiply(Cb,bw\_label); Cb\_label = (sum(sum(Cb\_label)))/(sum(sum(bw\_label))); % menghitung luas dan centroid masing2 objek Area = stats(n).Area; centroid = stats(n).Centroid; % jika nilai Cb > 120 maka dikenali sebagai koin silver

if Cb\_label>120

% jika luas > 70000 & luas < 80000 maka dikenali sebagai koin Rp. 200 if Area>70000 && Area <80000 nilai = 200; % menampilkan boundary pada objek plot(boundary(:,2), boundary(:,1), 'y', 'LineWidth', 4) % menampilkan nilai koin pada centroid objek text(centroid(1)-100,centroid(2),num2str(nilai),... 'Color','y','FontSize',20,'FontWeight','bold'); % mengisi variabel data\_koin dengan nilai koin data\_koin = [data\_koin;nilai]; end end end % menghitung banyaknya koin Rp. 200 [~,n\_200] = find(data\_koin==200); nilai\_200 = numel(n\_200); % menghitung jumlah nilai koin jumlah\_200 = nilai\_200\*200; % menginisialisasi variabel cell\_koin cell\_koin = cell(1,3); % mengisi variabel cell\_koin dengan data2 koin cell\_koin{1,1} = 'Rp. 200'; cell\_koin{1,2} = num2str(nilai\_200); cell\_koin{1,3} = ['Rp. ',num2str(jumlah\_200)]; % menampilkan cell\_koin pada tabel set(handles.uitable1,'Data',cell\_koin,... 'RowName',1) % jika deteksi dilakukan hanya pada uang logam Rp. 500 elseif val4==1 % menginisialisasi variabel data\_koin data\_koin = 0; % membuat boundary pada koin yang terdeteksi Boundaries = bwboundaries(bw,'noholes'); % untuk n = 1 s.d n = jumlah objek for n = 1:L boundary = Boundaries{n}; % menghitung nilai Cb dari masing2 objek bw\_label = (B==n); Cb\_label = immultiply(Cb,bw\_label); Cb\_label = (sum(sum(Cb\_label)))/(sum(sum(bw\_label))); % menghitung luas dan centroid masing2 objek Area = stats(n).Area; centroid = stats(n).Centroid; % jika nilai Cb > 120 maka dikenali sebagai koin silver

if Cb\_label>120

% jika luas > 80000 maka dikenali sebagai koin Rp. 500

if Area>80000

nilai = 500;

% menampilkan boundary pada objek

plot(boundary(:,2), boundary(:,1), 'y', 'LineWidth', 4)

% menampilkan nilai koin pada centroid objek

text(centroid(1)-100,centroid(2),num2str(nilai),...

'Color','y','FontSize',20,'FontWeight','bold');

% mengisi variabel data\_koin dengan nilai koin

data\_koin = [data\_koin;nilai];

end

% jika nilai Cb < 120 maka dikenali sebagai koin kuning

else

% jika luas < 70000 maka dikenali sebagai koin Rp. 500

if Area<70000

nilai = 500;

% menampilkan boundary pada objek

plot(boundary(:,2), boundary(:,1), 'c', 'LineWidth', 4)

% menampilkan nilai koin pada centroid objek

text(centroid(1)-100,centroid(2),num2str(nilai),...

'Color','c','FontSize',20,'FontWeight','bold');

% mengisi variabel data\_koin dengan nilai koin

data\_koin = [data\_koin;nilai];

end

end

end

end

end

end

% menghitung banyaknya koin Rp. 500

[~,n\_500] = find(data\_koin==500);

nilai\_500 = numel(n\_500);

% menghitung jumlah nilai koin

jumlah\_500 = nilai\_500\*500;

% menginisialisasi variabel cell\_koin

cell\_koin = cell(1,3);

% mengisi variabel cell\_koin dengan data2 koin

cell\_koin{1,1} = 'Rp. 500';

cell\_koin{1,2} = num2str(nilai\_500);

cell\_koin{1,3} = ['Rp. ',num2str(jumlah\_500)];

% menampilkan cell\_koin pada tabel

set(handles.uitable1,'Data',cell\_koin,...

'RowName',1)

% jika deteksi dilakukan hanya pada uang logam Rp. 1000

elseif val5==1

% menginisialisasi variabel data\_koin

data\_koin = 0;

% membuat boundary pada koin yang terdeteksi

Boundaries = bwboundaries(bw,'noholes');

% untuk n = 1 s.d n = jumlah objek

for n = 1:L

boundary = Boundaries{n};

% menghitung nilai Cb dari masing2 objek

bw\_label = (B==n);

Cb\_label = immultiply(Cb,bw\_label);

Cb\_label = (sum(sum(Cb\_label)))/(sum(sum(bw\_label)));

% menghitung luas dan centroid masing2 objek

Area = stats(n).Area;

centroid = stats(n).Centroid;

% jika nilai Cb < 120 maka dikenali sebagai koin kuning

if Cb\_label<120 % jika luas > 70000 maka dikenali sebagai koin Rp. 1000

if Area>70000

nilai = 1000;

% membuat boundary pada objek

plot(boundary(:,2), boundary(:,1), 'c', 'LineWidth', 4)

% menampilkan nilai koin pada centroid objek

text(centroid(1)-100,centroid(2),num2str(nilai),...

'Color','c','FontSize',20,'FontWeight','bold');

% mengisi variabel data\_koin dengan nilai koin

data\_koin = [data\_koin;nilai];

end

end

end

% menghitung banyaknya koin Rp. 1000

[~,n\_1000] = find(data\_koin==1000);

nilai\_1000 = numel(n\_1000);

% menghitung jumlah nilai koin

jumlah\_1000 = nilai\_1000\*1000;

% menginisialisasi variabel cell\_koin

cell\_koin = cell(1,3);

% mengisi variabel cell\_koin dengan data2 koin

cell\_koin{1,1} = 'Rp. 1000';

cell\_koin{1,2} = num2str(nilai\_1000);

cell\_koin{1,3} = ['Rp. ',num2str(jumlah\_1000)];

% menampilkan cell\_koin pada tabel

set(handles.uitable1,'Data',cell\_koin,...

'RowName',1)

end

1. Bagian pushbutton3

function pushbutton3\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% mereset button2

set(handles.pushbutton2,'Enable','off')

set(handles.uitable1,'Data',[])

set(handles.checkbox1,'Value',1)

set(handles.checkbox2,'Value',0)

set(handles.checkbox3,'Value',0)

set(handles.checkbox4,'Value',0)

set(handles.checkbox5,'Value',0)

axes(handles.axes1)

cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

axes(handles.axes2)

cla reset

set(gca,'XTick',[])

set(gca,'YTick',[])

1. Bagian pushbutton4

function pushbutton4\_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% menutup halaman deteksi\_uangLogam

close all;

1. Bagian checkbox1

function checkbox1\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox1

set(handles.checkbox1,'Value',1)

set(handles.checkbox2,'Value',0)

set(handles.checkbox3,'Value',0)

set(handles.checkbox4,'Value',0)

set(handles.checkbox5,'Value',0)

1. Bagian checkbox2

function checkbox2\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox2

set(handles.checkbox1,'Value',0)

set(handles.checkbox2,'Value',1)

set(handles.checkbox3,'Value',0)

set(handles.checkbox4,'Value',0)

set(handles.checkbox5,'Value',0)

1. Bagian checkbox3

function checkbox3\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox3 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox3

set(handles.checkbox1,'Value',0)

set(handles.checkbox2,'Value',0)

set(handles.checkbox3,'Value',1)

set(handles.checkbox4,'Value',0)

set(handles.checkbox5,'Value',0)

1. Bagian checkbox4

function checkbox4\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox4

set(handles.checkbox1,'Value',0)

set(handles.checkbox2,'Value',0)

set(handles.checkbox3,'Value',0)

set(handles.checkbox4,'Value',1)

set(handles.checkbox5,'Value',0)

1. Bagian checkbox5

function checkbox5\_Callback(hObject, eventdata, handles)

% hObject handle to checkbox5 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of checkbox5

set(handles.checkbox1,'Value',0)

set(handles.checkbox2,'Value',0)

set(handles.checkbox3,'Value',0)

set(handles.checkbox4,'Value',0)

set(handles.checkbox5,'Value',1)

Jika ingin menambahkan background pada GUI, bisa menggunakan syntax berikut ini (cari terlebih dahulu pada bagian

→ function deteksi\_uangLogam\_OpeningFcn(hObject, eventdata, handles, varargin) ) :

% Choose default command line output for deteksi\_uangLogam

handles.output = hObject;

hback = axes('unit','normalized','position',[0 0 1 1]);

uistack(hback,'bottom');

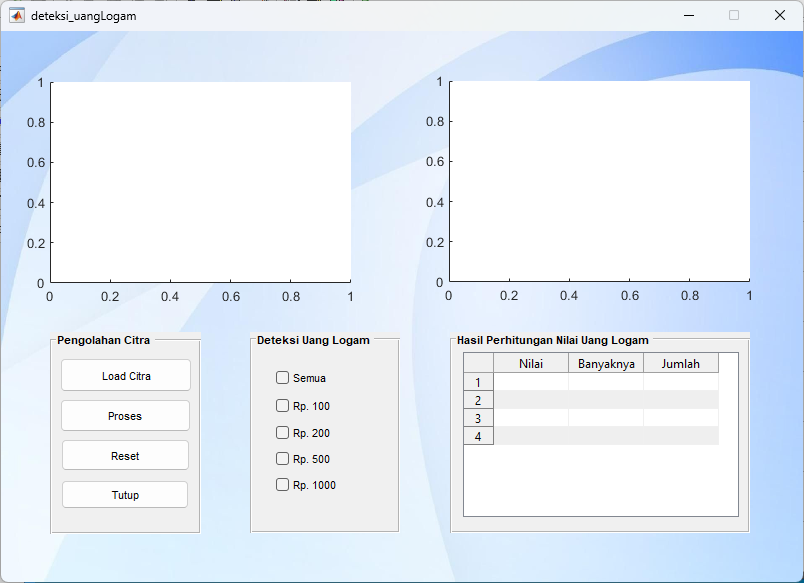
[back map]=imread('light-color-12.jpg');

image(back)

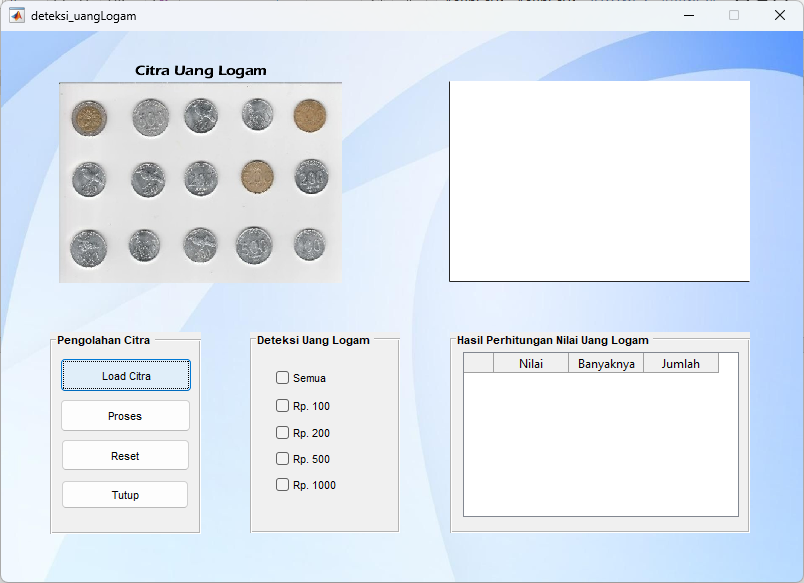
colormap(map)

set(hback,'handlevisibility','off','visible','off')

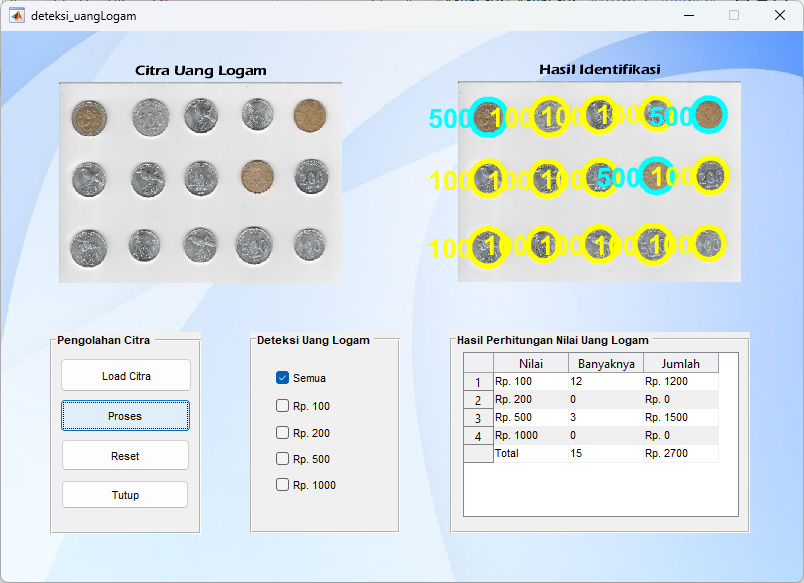
1. Setelah mengisikan semua bagian dengan syntax, bisa melakukan uji coba run program dengan menekan tombol Run Figure pada jendela GUIDE. Jika sudah dijalankan maka program tampak seperti ini:



Saat citra di load



Hasil proses citra



Saat program di reset

