

**APLIKASI PENDETEKSI JERAWAT PADA WAJAH DENGAN MENGGUNAKAN
TEKNIK PENGOLAHAN CITRA DIGITAL PADA FOTO**



DISUSUN OLEH :

Nama : Aslam Thariq Akbar Akrami

Nim : A11.2021.13224

Nama : Kang, Andini Wulandari

Nim : A11.2021.13273

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS DIAN NUSWANTORO

2022

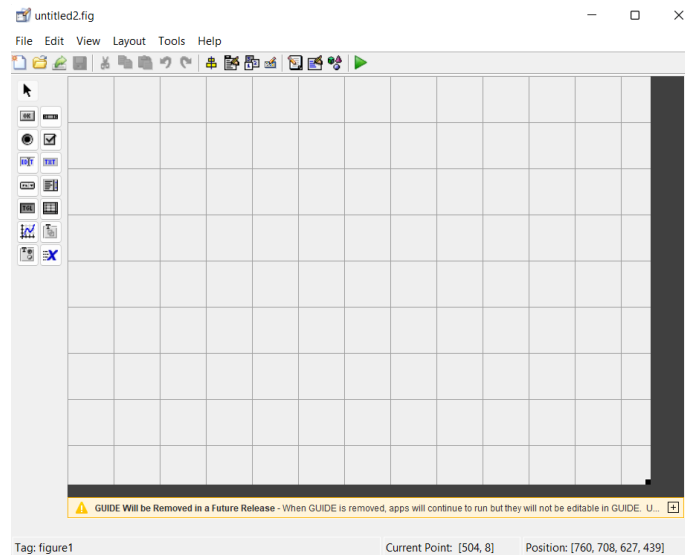
I. SOFTWARE YANG DIBUTUHKAN

Software yang dibutuhkan untuk membuat aplikasi pendeteksi jerawat di wajah dengan menggunakan teknik pengolahan citra digital pada foto yaitu Aplikasi MATLAB.

II. LANGKAH-LANGKAH PEMBUATAN GUI

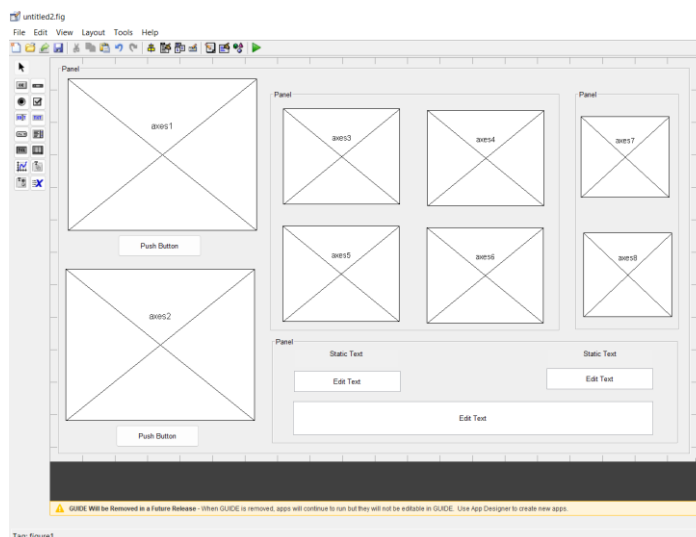
2.1 Buka Lembar Kerja GUI MATLAB

Buka Lembar kerja GUI MATLAB dengan tahapan ketikkan perintah “Guide” pada Command Windows. Kemudian akan muncul tampilan sebagai berikut :



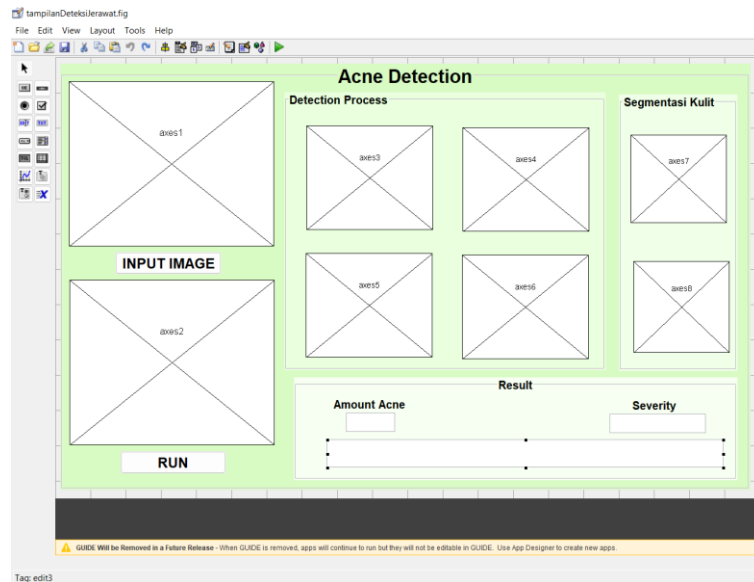
2.2 Merancang Tampilan

Setelah masuk ke lembar kerja GUI MATLAB, selanjutnya adalah merancang sebuah layout program pendeteksi jerawat. Layout tersebut kurang lebih seperti gambar di bawah.



2.3 Memperjelas Tampilan

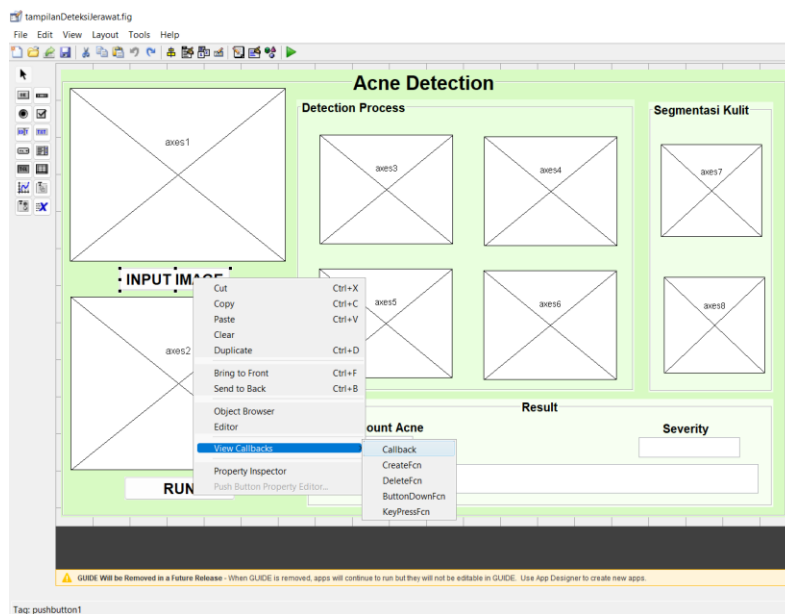
Setelah layout selesai dirancang, selanjutnya adalah memperjelas tampilan agar semua orang dapat menjalankan program tersebut. Seperti contoh, ubah nama Panel menjadi Acne Detection. Ubah nama Push Button menjadi Input Image dan lainnya.



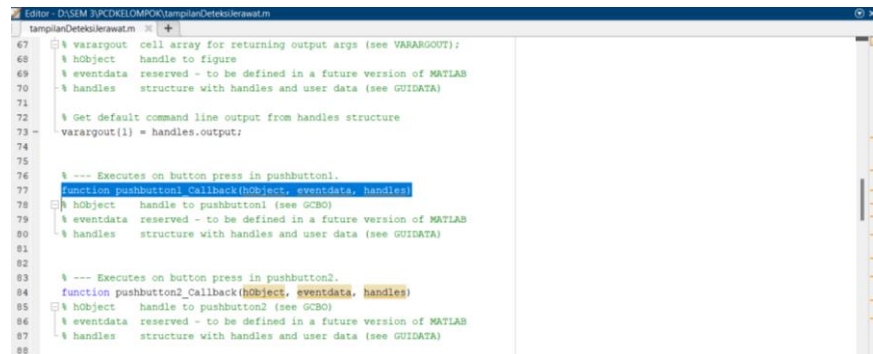
2.4 Memasukkan Fungsi

Setelah tampilan dirasa telah menarik dan mudah dipahami. Selanjutnya adalah memasukkan fungsi utama dengan tahapan sebagai berikut:

1. Klik kanan pada Push Button Input Image kemudian pilih View Callbacks dan klik Callback.



2. Secara Otomatis langsung diarahkan ke kode yang tertera pada M-file utama sebagai berikut.



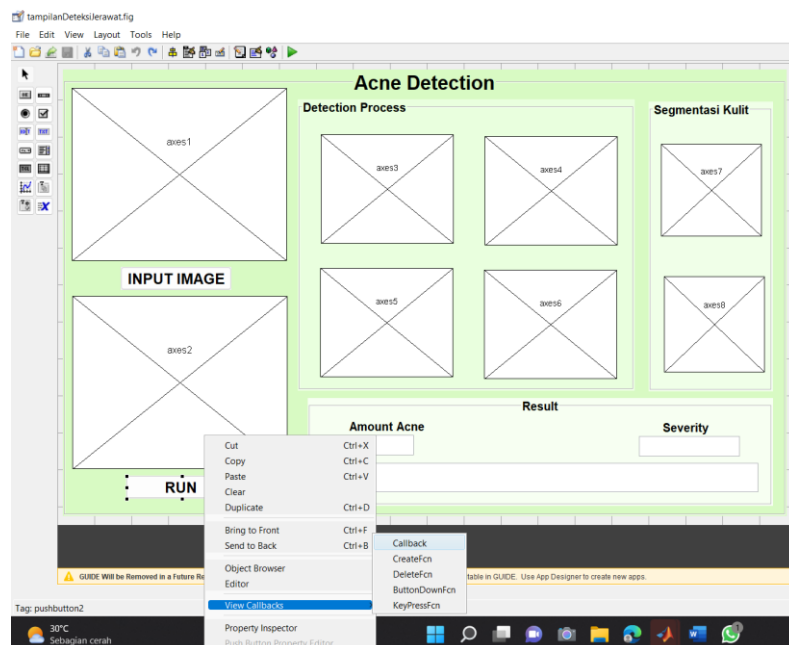
```
67 % varargin cell array for returning output args (see VARARGOUT);
68 % hObject handle to figure
69 % eventdata reserved - to be defined in a future version of MATLAB
70 % handles structure with handles and user data (see GUIDATA)
71
72 % Get default command line output from handles structure
73 varargin{1} = handles.output;
74
75
76 % --- Executes on button press in pushbutton1.
77 function pushbutton1_Callback(hObject, eventdata, handles)
78 % hObject handle to pushbutton1 (see GCBO)
79 % eventdata reserved - to be defined in a future version of MATLAB
80 % handles structure with handles and user data (see GUIDATA)
81
82
83 % --- Executes on button press in pushbutton2.
84 function pushbutton2_Callback(hObject, eventdata, handles)
85 % hObject handle to pushbutton2 (see GCBO)
86 % eventdata reserved - to be defined in a future version of MATLAB
87 % handles structure with handles and user data (see GUIDATA)
88
```

3. Masukkan kode berikut tepat dibawah kode di atas

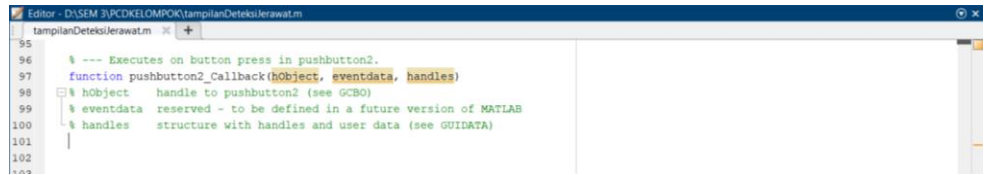
```
[filename,pathname] =
uigetfile('*.jpg;*.jpeg;*.png;*.tif');

try
    Img = imread(fullfile(pathname,filename));
    [~,~,m] = size(Img);
    if m == 3
        axes(handles.axes1)
        imshow(Img)
        handles.Img = Img;
        guidata(hObject, handles)
    end
catch
    msgbox('Please insert RGB Image')
end
```

4. Klik kanan pada Push Button Run kemudian pilih View Callbacks dan klik Callback.



5. Secara Otomatis langsung diarahkan ke kode yang tertera pada M-file utama sebagai berikut.



6. Masukkan kode berikut tepat dibawah kode di atas

```
%Resize gambar
gambar = handles.Img;
% Ukuran gambar Semula
[bar, kol, dlm] = size(gambar);

if (bar > kol)
    maxLength = bar;
    if (maxLength >= 480);
        gambar = imresize(gambar, [480 NaN]);
    end;
else
    maxLength = kol;
    if (maxLength >= 480);
        gambar = imresize(gambar, [NaN 480]);
    end;
end;

%Segmentasi Kulit

gambarG = rgb2gray(gambar);
level = graythresh(gambarG);

axes(handles.axes15);
imshow(gambarG);title('RGB2GRAY');

gambarB = im2bw(gambarG, level);
gambarB = imfill(gambarB, 'holes');
axes(handles.axes17);
imshow(gambarB);title('im2bw');

gambarCC = bwconncomp(gambarB);
gambarL = labelmatrix(gambarCC);

% Ukuran gambar Setelah Resize
[bar, kol, dlm] = size(gambar);
% Cari Connected Component Terbanyak
terluas = 0;
for i = 1 : length(gambarCC.PixelIdxList)
    if (length(gambarCC.PixelIdxList{i})) > terluas
        terluas = length(gambarCC.PixelIdxList{i});
        index = i;
    end;
end;
```

```

    % Ambil Label
    wajah = uint8(zeros(bar, kol, dlm));
    for i = 1 : bar
        for j = 1 : kol
            if gambarL(i, j) == index;
                wajah(i, j, :) = gambar(i, j, :);
            end;
        end;
    end;

    axes(handles.axes3);
    imshow(wajah);title('segmentasi gambar');

    %Perbaiki gambar

    % Sharpening
    wajahG = rgb2gray(wajah);
    h = fspecial('log', [9 9], 2);
    m = imfilter(wajahG, h, 'circular', 'same', 'conv');
    axes(handles.axes19);
    imshow(h);title('fspecial');
    axes(handles.axes20);
    imshow(m);title('imfilter');

    % Labeling
    candidate = logical(m);
    [labeledCandidate, numberOfCandidates] =
    bwlabel(candidate, 8);
    axes(handles.axes4);
    imshow(candidate);title('perbaiki gambar');

    %Ekstraksi Ciri Luas dan Bentuk
    % Eliminasi Berdasarkan Luas Dan Bentuk
    blobMeasurements = regionprops(labeledCandidate, 'Area',
    'Eccentricity');
    allArea = [blobMeasurements.Area];
    allEccentricity = [blobMeasurements.Eccentricity];
    meanArea = mean(allArea);
    stdArea = std(allArea);
    indexBlob = find(allArea >= 24 & allArea <= (meanArea +
    stdArea) & allEccentricity < 0.81);
    ambilBlob = ismember(labeledCandidate, indexBlob);
    blobBW = ambilBlob > 0;
    [labeledBlob, numberOfBlobs] = bwlabel(blobBW);
    axes(handles.axes5);
    imshow(blobBW);title('ekstraksi gambar luas dan bentuk');

    %Ekstraksi Ciri Warna
    % Eliminasi Berdasarkan Warna
    red = gambar(:, :, 1);
    green = gambar(:, :, 2);
    blue = gambar(:, :, 3);

    r = regionprops(labeledBlob, red, 'MeanIntensity');
    g = regionprops(labeledBlob, green, 'MeanIntensity');
    b = regionprops(labeledBlob, blue, 'MeanIntensity');

```

```

fiturR = [r.MeanIntensity]';
fiturG = [g.MeanIntensity]';
fiturB = [b.MeanIntensity]';
fitur = [fiturR fiturG fiturB];

meanR = mean(fiturR);
meanG = mean(fiturG);
meanB = mean(fiturB);
stdR = std(fiturR);
stdG = std(fiturG);
stdB = std(fiturB);

indexJerawat = [];
for i = 1 : numberOfBlobs
    if(fiturR(i) >= (meanR-stdR*1.75) && fiturR(i) <=
(meanR+stdR*1.75) && fiturG(i) >= (meanG-stdG*1.75) &&
fiturG(i) <= (meanG+stdG*1.75) && fiturB(i) >= (meanB-
stdB*1.75) && fiturB(i) <= (meanB+stdB*1.75))indexJerawat
= [indexJerawat i];
    end;
end;

jumlahJerawat = length(indexJerawat);
jerawatBW = ismember(labeledBlob, indexJerawat);

axes(handles.axes6);
imshow(jerawatBW); title('ekstraksi gambar warna');

%Marking

jerawatEdge = edge(jerawatBW, 'canny');
hasil = gambar;

for i = 1 : bar
    for j = 1 : kol
        if jerawatEdge(i, j) == 1;
            hasil(i, j, 1) = 0;
            hasil(i, j, 2) = 255;
            hasil(i, j, 3) = 0;
        end;
    end;
end;

hasil = uint8(hasil);

axes(handles.axes2);
imshow(hasil);
set(handles.edit1, 'String', jumlahJerawat);

print = strcat(severity(jumlahJerawat));
set(handles.edit2, 'String', print);

print = strcat(result(jumlahJerawat));
set(handles.edit3, 'String', print);

```

```

function [stringOut] = severity(jumlahJerawat)

Severity = {' Mild', ' Moderate', ' Severe', ' Very
Severe'};

if(jumlahJerawat<=5)
    i = 1;
else
    if(jumlahJerawat<=20)
        i=2;
    else
        if(jumlahJerawat<=50)
            i=3;
        else
            i=4;
        end
    end
end

stringOut = Severity{i};

function [stringOut] = result(jumlahJerawat)

Result = {'Your skin does not have any acne', 'You have
little acne on your skin, please wash your face
properly', 'You have a few acne on your skin, please
watch your diet', 'You have many acne on your skin,
please wash your face properly', 'You have so many acne
on your skin, please connect to doctor'};

if(jumlahJerawat<=0)
    c = 1;
else
    if(jumlahJerawat<=10)
        c = 2;
    else
        if(jumlahJerawat<=20)
            c = 3;
        else
            if (jumlahJerawat<=30)
                c = 4;
            else
                c = 5;
            end
        end
    end
end

stringOut = Result{c};

```


7. Hasil akhirnya yaitu sebagai berikut :

```
Editor - E:\1. KULIAH\SEMESTER 3\pemrograman citra digital\pertemuan 5\jerawat\tugasKelompok\deteksJerawat.m
pendeteksiJerawat.m deteksiJerawat.m tampilanDeteksiJerawat.m severity.m deteksiJerawat.m

76 % --- Executes on button press in pushbutton1.
77 function pushbutton1_Callback(hObject, eventdata, handles)
78 % hObject handle to pushbutton1 (see GCBO)
79 % eventdata reserved - to be defined in a future version of MATLAB
80 % handles structure with handles and user data (see GUIDATA)
81 [filename,pathname] = uigetfile('*.jpg;*.jpeg;*.png;*.tif');
82
83 try
84     Img = imread(fullfile(pathname,filename));
85     [~,~,m] = size(Img);
86     if m == 3
87         axes(handles.axes1)
88         imshow(Img)
89         handles.Img = Img;
90         guidata(hObject, handles)
91     end
92 catch
93     msgbox('Please insert RGB Image')
94 end
95
96 % --- Executes on button press in pushbutton2.
97 function pushbutton2_Callback(hObject, eventdata, handles)
98 % hObject handle to pushbutton2 (see GCBO)
99 % eventdata reserved - to be defined in a future version of MATLAB
100 % handles structure with handles and user data (see GUIDATA)
101
102 %Resize gambar
103 gambar = handles.Img;
104 % Ukuran gambar Semula
105 [bar, kol, dlm] = size(gambar);
106
107 if (bar > kol)
108     maxLength = bar;
109     if (maxLength >= 480);
110         gambar = imresize(gambar, [480 NaN]);
111     end;
112 else
113     maxLength = kol;
114     if (maxLength >= 480);
115         gambar = imresize(gambar, [NaN 480]);
116     end;
117 end;
118
119 %Segmentasi Kulit
120
121 gambarG = rgb2gray(gambar);
122 level = graythresh(gambarG);
123
124 axes(handles.axes15);
125 imshow(gambarG);title('RGB2GRAY');
126
127 gambarB = im2bw(gambarG, level);
128 gambarB = imfill(gambarB, 'holes');
129 axes(handles.axes17);
130 imshow(gambarB);title('im2bw');
131
132
133 gambarCC = bwconncomp(gambarB);
134 gambarL = labelmatrix(gambarCC);
135
136 % Ukuran gambar Setelah Resize
137 [bar, kol, dlm] = size(gambar);
138 % Cari Connected Component Terbanyak
139 terluas = 0;
140 for i = 1 : length(gambarCC.PixelIdxList)
141     if (length(gambarCC.PixelIdxList{i})) > terluas
142         terluas = length(gambarCC.PixelIdxList{i});
143         index = i;
144     end;
145 end;
146
147 % Ambil Label
148 wajah = uint8(zeros(bar, kol, dlm));
149 for i = 1 : bar
150     for j = 1 : kol
151         if gambarL(i, j) == index;
152             wajah(i, j, :) = gambar(i, j, :);
153         end;
154     end;
155 end;
156 axes(handles.axes3);
157 imshow(wajah);title('segmentasi gambar');
158
159 %Perbaiki gambar
160 % Sharpening
161 wajahG = rgb2gray(wajah);
162 h = fspecial('log', [9 9], 2);
163 m = imfilter(wajahG, h, 'circular', 'same', 'conv');
164
165 axes(handles.axes19);
166 imshow(h);title('fspecial');
167 axes(handles.axes20);
168 imshow(m);title('imfilter');
169
```

```

169
170
171 % Labeling
172 - candidate = logical(m);
173 - [labeledCandidate, numberOfCandidates] = bwlabel(candidate, 8);
174 - axes(handles.axes4);
175 - imshow(candidate); title('perbaikan gambar');
176
177 %Ekstraksi Ciri Luas dan Bentuk
178
179 % Eliminasi Berdasarkan Luas Dan Bentuk
180 - blobMeasurements = regionprops(labeledCandidate, 'Area', 'Eccentricity');
181
182 - allArea = [blobMeasurements.Area];
183 - allEccentricity = [blobMeasurements.Eccentricity];
184
185 - meanArea = mean(allArea);
186 - stdArea = std(allArea);
187
188 - indexBlob = find(allArea >= 24 & allArea <= (meanArea + stdArea) &
189 - allEccentricity < 0.81);
190
191 - ambilBlob = ismember(labeledCandidate, indexBlob);
192 - blobBW = ambilBlob > 0;
193 - [labeledBlob, numberOfBlobs] = bwlabel(blobBW);
194
195 - axes(handles.axes5);
196 - imshow(blobBW); title('ekstraksi gambar luas dan bentuk');
197
198 %Ekstraksi Ciri Warna
199 % Eliminasi Berdasarkan Warna
200 - red = gambar(:, :, 1);
201 - green = gambar(:, :, 2);
202 - blue = gambar(:, :, 3);
203
204 - r = regionprops(labeledBlob, red, 'MeanIntensity');
205 - g = regionprops(labeledBlob, green, 'MeanIntensity');
206 - b = regionprops(labeledBlob, blue, 'MeanIntensity');
207
208 - fiturR = [r.MeanIntensity]';
209 - fiturG = [g.MeanIntensity]';
210 - fiturB = [b.MeanIntensity]';
211 - fitur = [fiturR fiturG fiturB];
212
213 - meanR = mean(fiturR);
214 - meanG = mean(fiturG);
215 - meanB = mean(fiturB);
216 - stdR = std(fiturR);
217 - stdG = std(fiturG);
218 - stdB = std(fiturB);
219
220 - indexJerawat = [];
221 - for i = 1 : numberOfBlobs
222 -     if (fiturR(i) >= (meanR-stdR*1.75) && fiturR(i) <= (meanR+stdR*1.75) && fiturG(i) >= (meanG-stdG*1.75) && fiturG(i) <=
223 -         (meanG+stdG*1.75) && fiturB(i) >= (meanB-stdB*1.75) && fiturB(i) <= (meanB+stdB*1.75))
224 -         indexJerawat = [indexJerawat i];
225 -     end
226 - end
227
228 - jumlahJerawat = length(indexJerawat);
229 - jerawatBW = ismember(labeledBlob, indexJerawat);
230
231 - axes(handles.axes6);
232 - imshow(jerawatBW); title('ekstraksi gambar warna');
233
234 %Marking
235 jerawatEdge = edge(jerawatBW, 'canny');
236 hasil = gambar;
237
238 - for i = 1 : bar
239 -     for j = 1 : kol
240 -         if jerawatEdge(i, j) == 1
241 -             hasil(i, j, 1) = 0;
242 -             hasil(i, j, 2) = 255;
243 -             hasil(i, j, 3) = 0;
244 -         end
245 -     end
246 - end
247
248 - hasil = uint8(hasil);
249
250 - axes(handles.axes2);
251 - imshow(hasil);
252 - set(handles.edit1, 'String', jumlahJerawat);
253
254 - print = strcat(severity(jumlahJerawat));
255

```

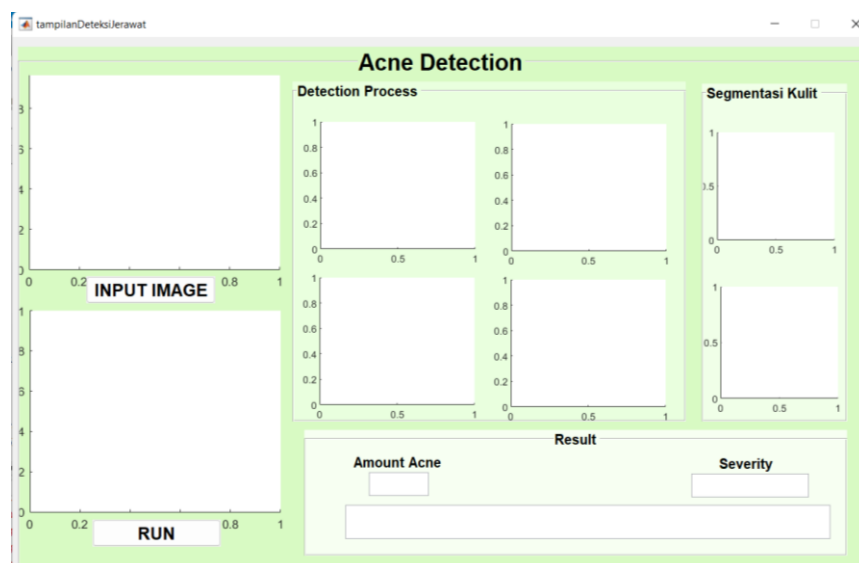
```

256 - set(handles.edit2,'String',print);
257 -
258 - print = strcat(result(jumlahJerawat));
259 - set(handles.edit3,'String',print);
260 -
261 -
262 -
263 - function [stringOut] = severity(jumlahJerawat)
264 -
265 - Severity = {'Mild', 'Moderate', 'Severe', 'Very Severe'};
266 -
267 - if(jumlahJerawat<=5)
268 -     i = 1;
269 - else
270 -     if(jumlahJerawat<=20)
271 -         i=2;
272 -     else
273 -         if(jumlahJerawat<=50)
274 -             i=3;
275 -         else
276 -             i=4;
277 -         end
278 -     end
279 - end
280 -
281 - stringOut = Severity{i};
282 -
283 - function [stringOut] = result(jumlahJerawat)
284 -
285 - Result = {'Your skin does not have any acne', 'You have little acne on your skin, please wash your face properly', 'You have a
286 -
287 - if(jumlahJerawat<=0)
288 -     c = 1;
289 - else
290 -     if(jumlahJerawat<=10)
291 -         c = 2;
292 -     else
293 -         if(jumlahJerawat<=20)
294 -             c = 3;
295 -         else
296 -             if(jumlahJerawat<=30)
297 -                 c = 4;
298 -             else
299 -                 c = 5;
300 -             end
301 -         end
302 -     end
303 - end
304 -
305 - stringOut = Result{c};

```

2.5 Menjalankan GUI MATLAB

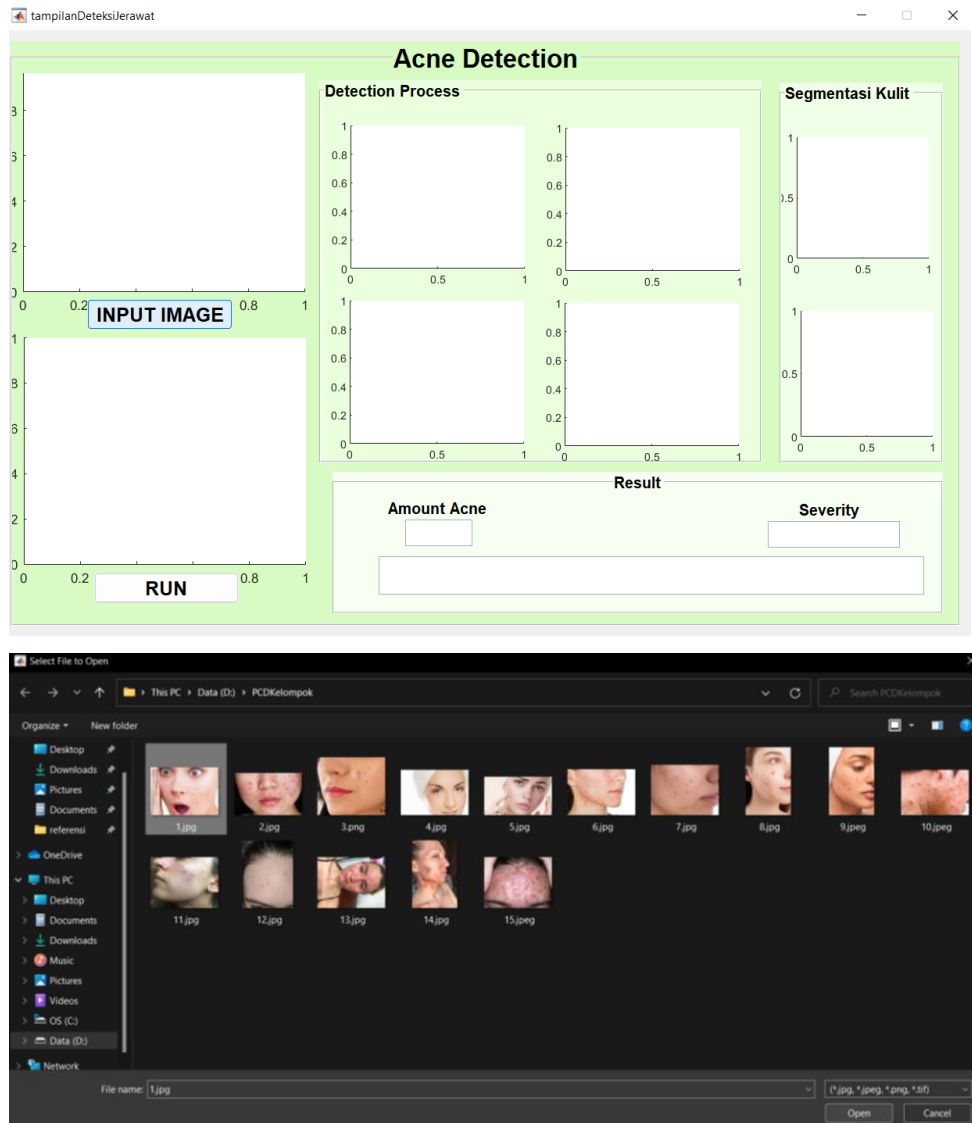
Setelah fungsi dimasukkan, tahap selanjutnya adalah menjalankan GUI MATAB. Caranya adalah dengan klik ikon segitiga warna Hijau pada lembar kerja GUI MATLAB. Jika berhasil maka program akan berjalan seperti berikut



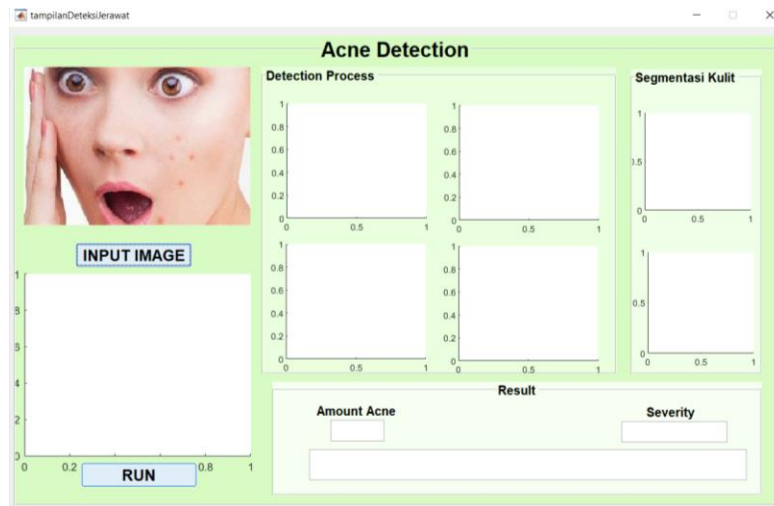
2.6 Menguji Fungsi GUI MATLAB

Tahap terakhir adalah menguji GUI MATLAB Program pendeteksi jerawat di wajah apakah dapat berjalan dengan baik atau tidak. Caranya adalah sebagai berikut

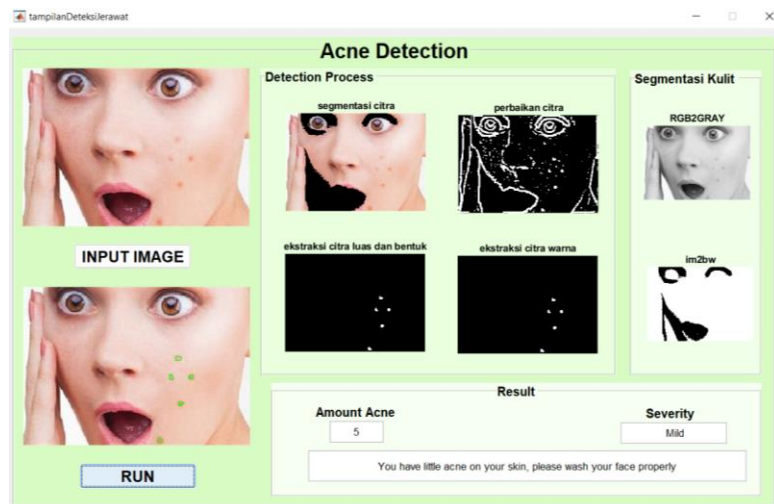
1. Klik Input Image dan masukkan gambar



2. Klik Run untuk menjalankan program



3. Maka akan muncul hasil sebagai berikut :



III. HASIL DAN PEMBAHASAN

1.1 Implementasi Perangkat Lunak

Semua metode dari perancangan perangkat lunak yang telah dirancang kemudian dilakukan implementasi dengan menggunakan MATLAB R2021a.

1.1.1. Resize Gambar

Resize citra merupakan proses untuk mengubah ukuran citra menjadi lebih kecil jika ukuran dimensi citra terlalu besar.

```
%Resize gambar
gambar = handles.Img;
% Ukuran gambar Semula
[bar, kol, dlm] = size(gambar);

if (bar > kol)
    maxLength = bar;
    if (maxLength >= 480);
        gambar = imresize(gambar, [480 NaN]);
    end;
else
    maxLength = kol;
    if (maxLength >= 480);
        gambar = imresize(gambar, [NaN 480]);
    end;
end;
```

Kode Program Resize Gambar

Citra input akan di-resize jika memiliki ukuran baris atau kolom lebih dari 480 piksel. Setelah memenuhi kondisi tersebut, ukuran baris dan kolom akan dibandingkan. Jika baris lebih besar dari kolom, maka ukuran baris akan diubah menjadi 480 piksel dan ukuran kolom akan disesuaikan dengan perbandingan ukuran baris agar bentuk citra tidak berubah. Sedangkan jika kolom lebih besar dari baris, maka ukuran kolom akan diubah menjadi 480 piksel dan ukuran baris akan disesuaikan dengan perbandingan ukuran kolom agar bentuk citra tidak berubah.

1.1.2. Segmentasi Kulit

Segmentasi kulit merupakan proses untuk memisahkan objek kulit dengan background dan objek selain kulit. Proses pemisahan kulit dalam penelitian ini memanfaatkan fungsi connected-component yang sudah disediakan MATLAB..

```

%Segmentasi Kulit

gambarG = rgb2gray(gambar);
level = graythresh(gambarG);

axes(handles.axes15);
imshow(gambarG);title('RGB2GRAY');

gambarB = im2bw(gambarG, level);
gambarB = imfill(gambarB, 'holes');
axes(handles.axes17);
imshow(gambarB);title('im2bw');

gambarCC = bwconncomp(gambarB);
gambarL = labelmatrix(gambarCC);

    % Ukuran gambar Setelah Resize
[bar, kol, dlm] = size(gambar);
    % Cari Connected Component Terbanyak
terluas = 0;
for i = 1 : length(gambarCC.PixelIdxList)
    if (length(gambarCC.PixelIdxList{i})) > terluas
        terluas = length(gambarCC.PixelIdxList{i});
        index = i;
    end;
end;

    % Ambil Label
wajah = uint8(zeros(bar, kol, dlm));
for i = 1 : bar
    for j = 1 : kol
        if gambarL(i, j) == index;
            wajah(i, j, :) = gambar(i, j, :);
        end;
    end;
end;
axes(handles.axes3);
imshow(wajah);title('segmentasi gambar');

```

Kode Program Segmentasi Kulit

Citra yang sudah melalui proses resize citra dikonversi menjadi citra biner dengan metode Otsu. Objek pada citra biner dikelompokkan menggunakan connected-component dan diberi label untuk tiap objek yang terpisah. Setiap objek akan dihitung luasnya dan dicari objek dengan luas terbesar untuk menentukan objek kulit. Objek tersebut diambil nilai indeksnya dan dikembalikan nilai warnanya menjadi RGB.

1.1.3. Perbaikan Citra

Proses perbaikan citra merupakan proses untuk mencari kandidat jerawat pada citra. Proses perbaikan citra menggunakan metode sharpening Laplacian of Gaussian dengan memanfaatkan fungsi `fspecial` dan `imfilter` yang disediakan MATLAB.

```
%Perbaikan gambar
% Sharpening
wajahG = rgb2gray(wajah);
h = fspecial('log', [9 9], 2);
m = imfilter(wajahG, h, 'circular', 'same', 'conv');

axes(handles.axes19);
imshow(h);title('fspecial');
axes(handles.axes20);
imshow(m);title('imfilter');

% Labeling
candidate = logical(m);
[labeledCandidate, numberOfCandidates] = bwlabeled(candidate, 8);
axes(handles.axes4);
imshow(candidate);title('perbaikan gambar');
```

Kode Program Perbaikan Citra

Citra yang telah disegmentasi dikonversi menjadi citra grayscale. Kemudian dilakukan sharpening pada citra tersebut dengan menggunakan metode Laplacian of Gaussian, sehingga menghasilkan sebuah keluaran citra LoG dalam bentuk citra biner. Citra yang dihasilkan dari sharpening menggunakan filtering Laplacian of Gaussian ini sudah terdapat jerawat di 38 dalamnya. Sehingga citra Laplacian of Gaussian ini disebut sebagai kandidat jerawat.

1.1.4. Ekstraksi Ciri

Proses ekstraksi ciri bertujuan untuk mengenali objek jerawat dan memisahkannya berdasarkan kondisi yang mematuhi. Adapun dalam penelitian ini menggunakan metode ekstraksi ciri berdasarkan luas, bentuk dan warna.

Luas dan Bentuk

```
% Eliminasi Berdasarkan Luas Dan Bentuk
blobMeasurements = regionprops(labeledCandidate, 'Area',
'Eccentricity');
allArea = [blobMeasurements.Area];
allEccentricity = [blobMeasurements.Eccentricity];
meanArea = mean(allArea);
stdArea = std(allArea);
axes(handles.axes5);
imshow(blobBW);title('ekstraksi citra luas dan bentuk');
```



```

indexBlob = find(allArea >= 24 & allArea <= (meanArea +
stdArea) & allEccentricity < 0.81);

ambilBlob = ismember(labeledCandidate, indexBlob);
blobBW = ambilBlob > 0;
[labeledBlob, numberOfBlobs] = bwlablel(blobBW);

axes(handles.axes5);
imshow(blobBW);title('ekstraksi gambar luas dan bentuk');

```

Kode Program Ekstraksi Luas dan Bentuk

Pada Gambar diatas, input proses yaitu citra LoG yang kemudian diberi label setiap objeknya. Setiap objek yang telah diberi label dihitung nilai eccentricity dan luasnya. Kemudian objek-objek tersebut diseleksi dengan kondisi luas area lebih dari sama dengan 24 piksel dan eccentricity kurang dari sama dengan 0,81. Selanjutnya objek yang tidak memenuhi kondisi akan dieliminasi dari citra tersebut, sehingga hanya tersisa objek yang memenuhi kondisi tersebut.

Warna

```

% Eliminasi Berdasarkan Warna
red = gambar(:, :, 1);
green = gambar(:, :, 2);
blue = gambar(:, :, 3);

r = regionprops(labeledBlob, red, 'MeanIntensity');
g = regionprops(labeledBlob, green, 'MeanIntensity');
b = regionprops(labeledBlob, blue, 'MeanIntensity');

fiturR = [r.MeanIntensity]';
fiturG = [g.MeanIntensity]';
fiturB = [b.MeanIntensity]';
fitur = [fiturR fiturG fiturB];

meanR = mean(fiturR);
meanG = mean(fiturG);
meanB = mean(fiturB);
stdR = std(fiturR);
stdG = std(fiturG);
stdB = std(fiturB);

indexJerawat = [];
for i = 1 : numberOfBlobs
    if(fiturR(i) >= (meanR-stdR*1.75) && fiturR(i) <= (meanR+stdR*1.75)
    && fiturG(i) >= (meanG-stdG*1.75) && fiturG(i) <= (meanG+stdG*1.75) &&
    fiturB(i) >= (meanB-stdB*1.75) && fiturB(i) <=
    (meanB+stdB*1.75))indexJerawat = [indexJerawat i];
    end;
end;

```

```

jumlahJerawat = length(indexJerawat);
jerawatBW = ismember(labeledBlob, indexJerawat);

axes(handles.axes6);
imshow(jerawatBW); title('ekstraksi gambar warna');

```

Kode Program Ekstraksi Warna

Pada Gambar diatas, input berupa citra eliminasi yang telah disegmentasi berdasarkan luas dan bentuk. Setiap objek pada citra tersebut diberi label dan dihitung nilai mean intensity dari komponen warna RGB per objek. Kemudian objek tersebut diseleksi nilai mean intensity komponen warna RGB tiap objek dengan rentang warna berupa komputasi nilai mean dan standar deviasi dari komponen warna RGB objek dari citra tersebut, sehingga tersisa objek-objek yang dikenali sebagai jerawat. Keluaran dari proses ini adalah citra jerawat berupa citra biner.

1.1.5. Marking

Proses marking merupakan proses untuk menandai setiap objek jerawat yang telah dikenali dengan memberi garis tepi pada jerawat yang telah dikenali.

```

%Marking
jerawatEdge = edge(jerawatBW, 'canny');
hasil = gambar;

for i = 1 : bar
    for j = 1 : kol
        if jerawatEdge(i, j) == 1;
            hasil(i, j, 1) = 0;
            hasil(i, j, 2) = 255;
            hasil(i, j, 3) = 0;
        end;
    end;
end;
hasil = uint8(hasil);

axes(handles.axes2);
imshow(hasil);

```

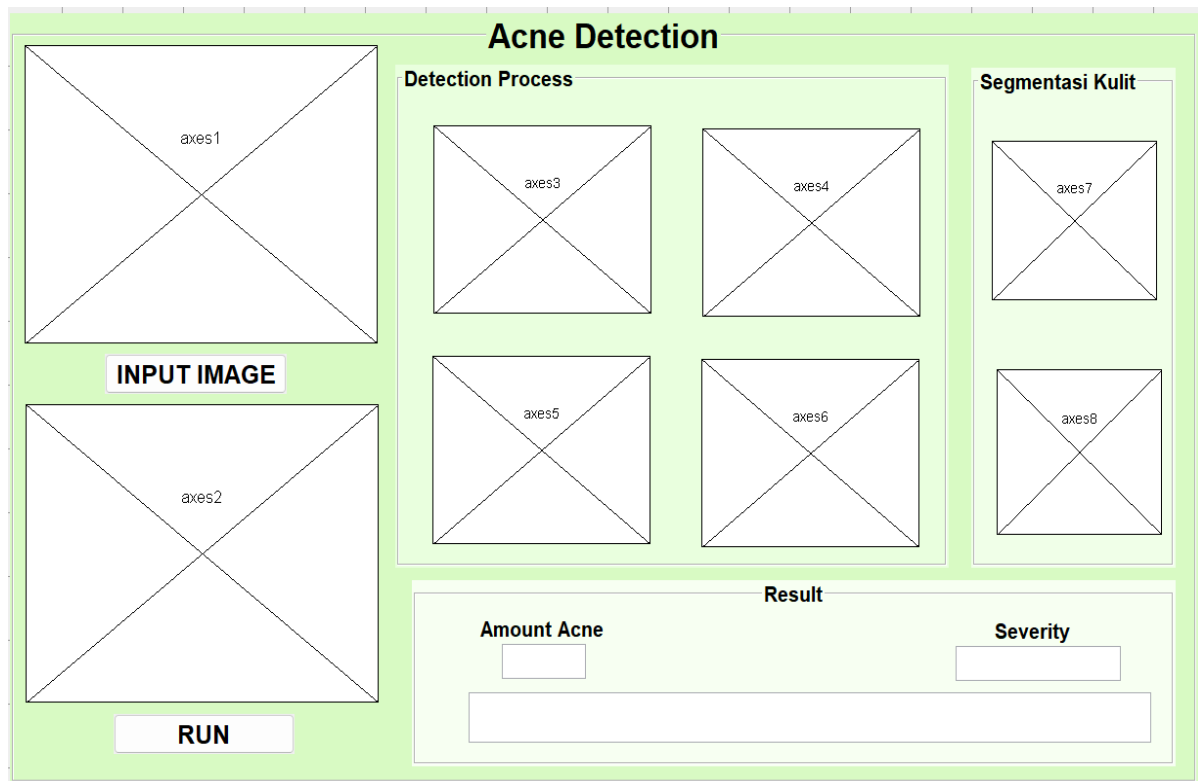
Kode Program Marking

Input dari proses marking adalah citra jerawat biner yang sudah disegmentasi. Citra tersebut akan dideteksi tepi dengan metode Canny, sehingga citra hanya terdapat garis tepi dari objek jerawat saja. Koordinat garis tepi dari objek jerawat tersebut akan dijadikan acuan untuk

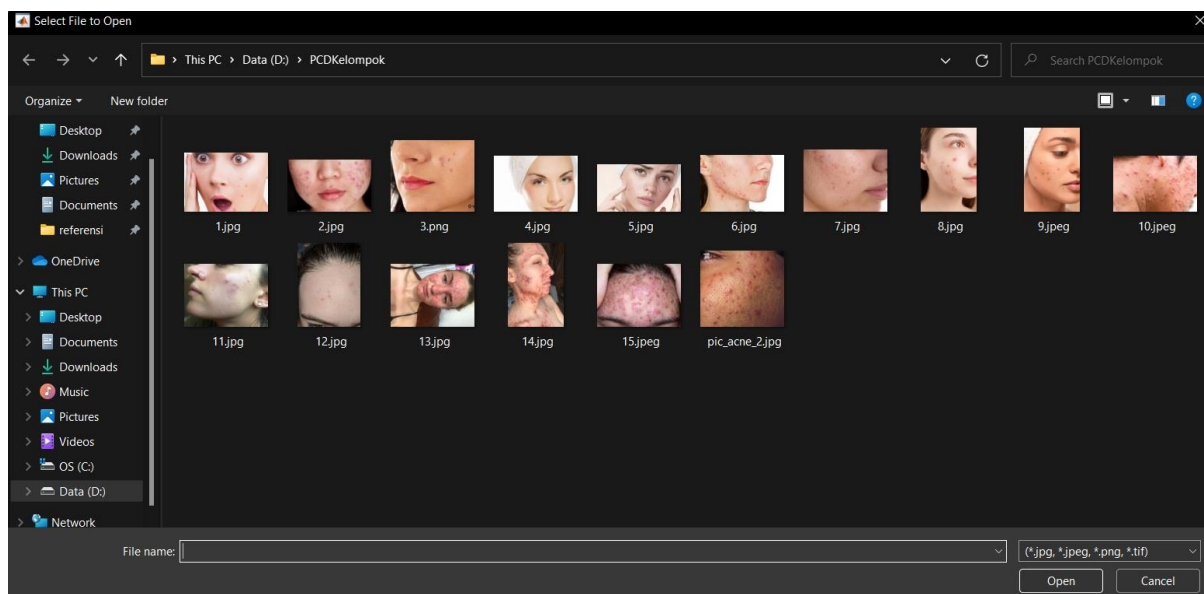
garis tanda objek jerawat pada citra awal. Garis tepi objek jerawat pada citra awal akan diberi warna hijau untuk menandakan bahwa objek yang diberi garis tersebut adalah jerawat.

1.2 Implementasi Antarmuka

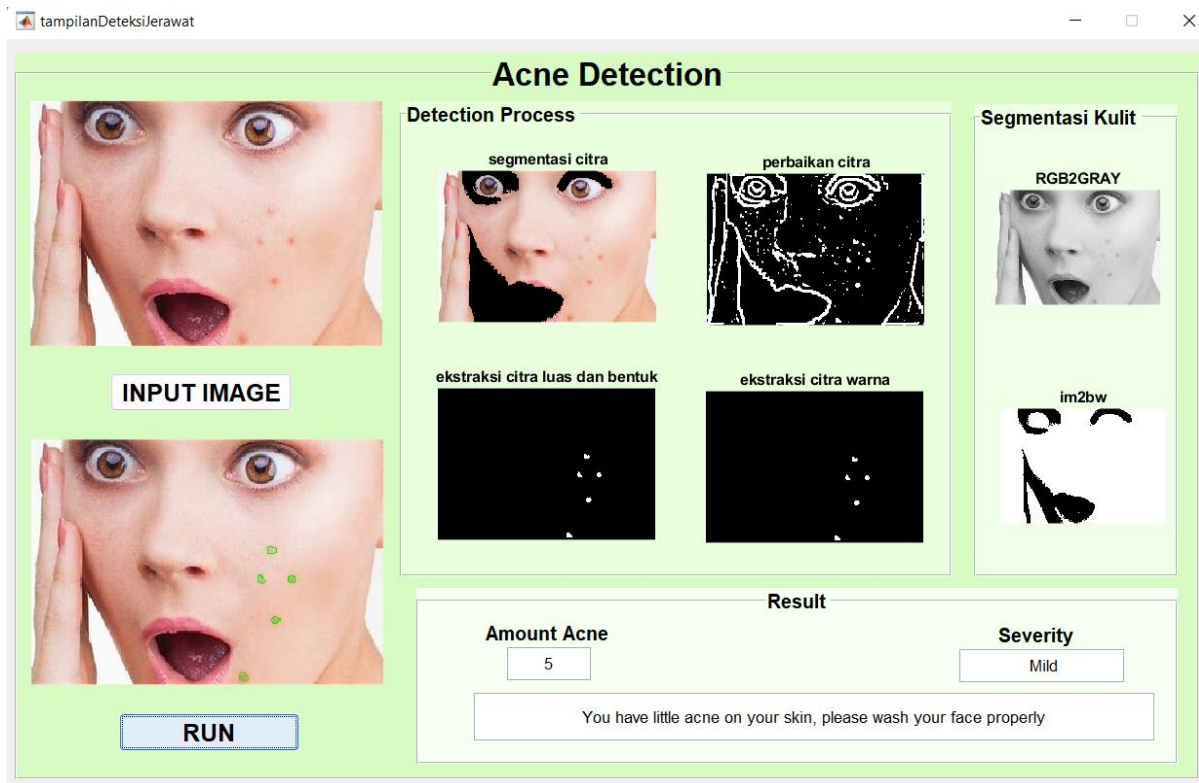
Rancangan dari antarmuka diimplementasikan dengan tools GUIDE pada MATLAB.



Saat menekan tombol INPUT IMAGE, maka sistem akan mengeluarkan pop-up window file input untuk memilih file foto seperti terlihat pada Gambar berikut.


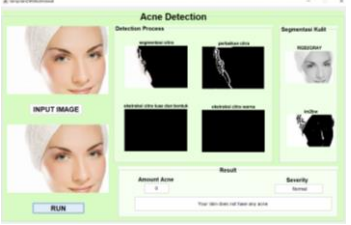

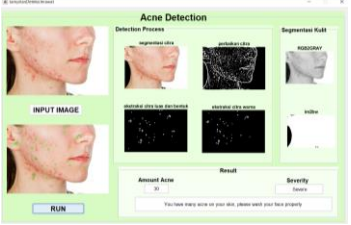


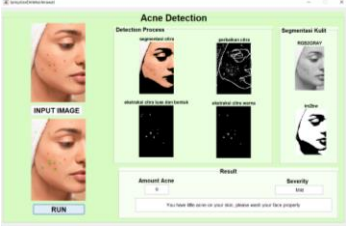


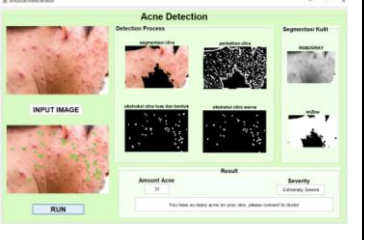

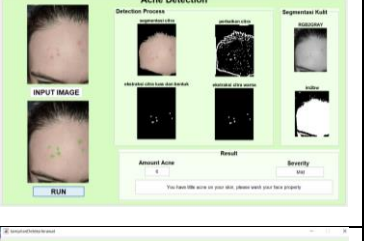
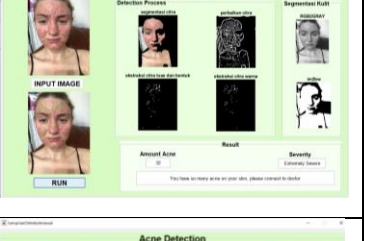
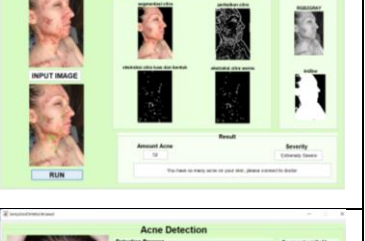
Setelah file berhasil dimasukkan maka foto akan ter-load dan dengan menekan tombol RUN, maka sistem akan menampilkan hasil dari citra dengan jerawat yang telah ditandai seperti pada Gambar.



1.3 Pengujian Sistem

No	Nama File	Jumlah Jerawat	Severity	Hasil	Tampilan GUI
1	1.jpg	5	Mild	You have little acne on your skin, please wash your face properly	
2	2.jpg	31	Extremely Severe	You have so many acne on your skin, please connect to doctor	

3	3.jpg	7	Mild	You have little acne on your skin, please wash your face properly	
4	4.jpg	0	Normal	Your skin does not have any acne	
5	5.jpg	4	Mild	You have little acne on your skin, please wash your face properly	
6	6.jpg	30	Severe	You have many acne on your skin, please wash your face properly	
7	7.jpg	21	Severe	You have many acne on your skin, please wash your face properly	
8	8.jpg	13	Moderate	You have a few acne on your skin, please watch your diet	
9	9.jpg	9	Mild	You have little acne on your skin, please wash your face properly	

10	10.jpg	37	Extremely Severe	You have so many acne on your skin, please connect to doctor	
11	11.jpg	10	Mild	You have little acne on your skin, please wash your face properly	
12	12.jpg	6	Mild	You have little acne on your skin, please wash your face properly	
13	13.jpg	32	Extremely Severe	You have so many acne on your skin, please connect to doctor	
14	14.jpg	52	Extremely Severe	You have so many acne on your skin, please connect to doctor	
15	15.jpg	22	Severe	You have many acne on your skin, please wash your face properly	