

**TUGAS 3**  
**OPERASI MORFOLOGI DAN SEGMENTASI CITRA**

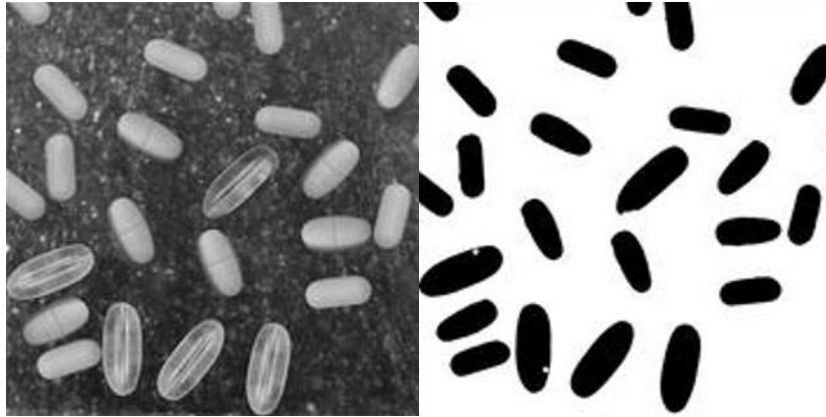


**Susaf Noor Azhar**  
**18/437646/PTK/12679**

**VISI KOMPUTER**  
**PROGRAM STUDI MAGISTER INFORMASI**  
**DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS GADJAH MADA**  
**FEBRUARI 2020**

## 1. Tujuan Eksperimen

Diberikan dua gambar pada Gambar 1 yang merupakan gambar tablet obat dan kapsul semi-transparan dengan latar belakang kasar. Adapun tujuan dari percobaan ini adalah mendapatkan segmentasi gambar untuk memisahkan objek dengan latar belakang (kanan).



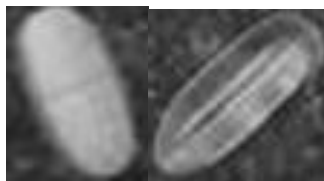
*Gambar 1. Gambar yang akan disegmentasi (kiri), hasil dari segmentasi yang diinginkan (kanan)*

## 2. Hipotesis

Dari gambar, bagian-bagian gambar dapat dikelompokkan menjadi:

1. Latar belakang kasar,
2. Objek berupa tablet,
3. Objek berupa kapsul,

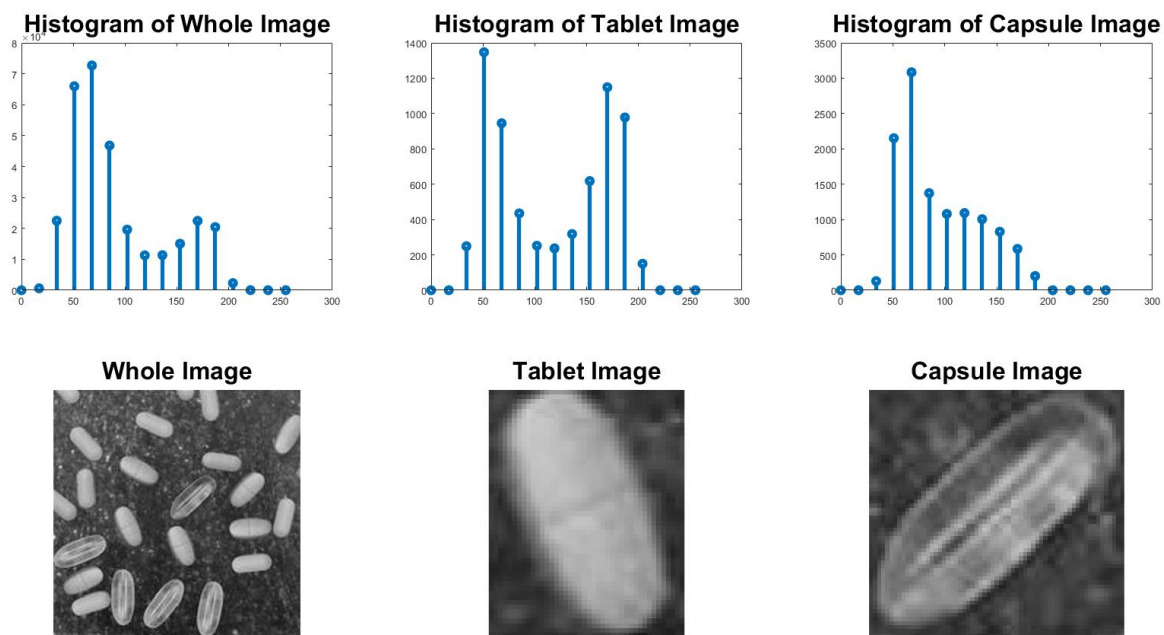
Walaupun tujuan utama dari eksperimen ini adalah hanya mendeteksi objek dengan latar belakang, namun penulis ingin mencoba lebih lanjut untuk memisahkan dua jenis objek yang berbeda berdasarkan komposisi warnanya.



*Gambar 2. Gambar tablet obat (kiri) dan kapsul semi-transparan (kanan)*

### 3. Eksperimen

Eksperimen ini dimulai dengan analisa histogram nilai piksel dari gambar, baik secara utuh, maupun berdasarkan masing-masing objek. Histogram dari gambar ditampilkan pada Gambar 3. Dari gambar dapat dilihat perbedaan histogram masing-masing. Dimana objek berupa tablet memiliki komposisi piksel terang yang lebih banyak (150-200) dibandingkan lainnya, sedangkan kapsul memiliki sebaran piksel dengan nilai diantara 50 dan 100. Angka-angka ini penting untuk menentukan treshold mask yang akan dibuat. ( $50/255 = 0,196$ ;  $100/255 = 0,392$ ;  $150/255 = 0,588$ )



Gambar 3. Histogram nilai piksel dari gambar (keseluruhan, tablet dan kapsul)

Untuk memisahkan ketiganya (latar belakang, tablet dan kapsul) proses dari eksperimen dapat dirangkum sebagai berikut.



Gambar 4. Diagram proses dari eksperimen

Keseluruhan eksperimen terbagi menjadi tiga proses untuk mendapatkan hasil yang diinginkan oleh penulis.

### 3.1. Proses 1

Proses pertama dapat dilihat pada Gambar 5. Dimana dimulai dengan pembuatan Mask1. Mask biner dibuat berdasarkan nilai threshold yang dihitung menggunakan metode Otsu [1]. Dimana penentuan nilai threshold dihitung berdasarkan algoritma berikut:

1. Menghitung histogram dan probabilitas setiap level intensitas,
2. Menetapkan nilai awal  $\omega_i(0)$  dan  $\mu_i(0)$ ,
3. Perhitungan berulang untuk semua kemungkinan threshold  $t = 1, \dots$  hingga intensitas maksimum:
  - a. Perbarui nilai  $\omega_i$  dan  $\mu_i$ ,
  - b. Hitung  $\sigma_b^2(t) = \omega_0(t)\omega_1(t)[\mu_0(t) - \mu_1(t)]^2$
4. Dipilih threshold yang memenuhi nilai  $\sigma_b^2(t)$  maksimum

Adapun dalam Matlab, didapat nilai Otsu Threshold dari gambar adalah sebesar 0.477.



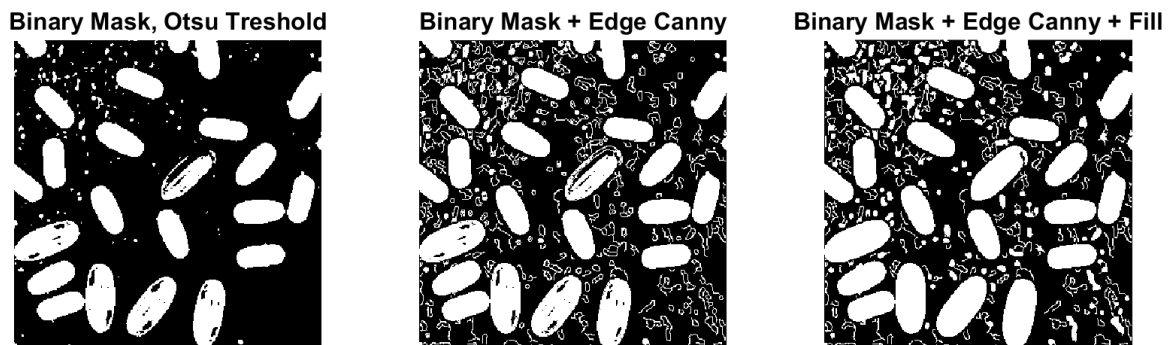
Gambar 5. Diagram alur pada Proses 1

Namun, pembuatan mask menggunakan metode Otsu saja dirasa tidak cukup. Terlihat pada Gambar 6, bahwa beberapa tepi dari objek dalam gambar tidak terdeteksi. Sehingga ditambahkan mask menggunakan edge detector berdasarkan metode Canny [2]. Metode ini dilakukan dalam lima langkah:

1. Menerapkan Gaussian Filter untuk memperhalus gambar dan menghilangkan noise,
2. Menghitung gradien intensity dari gambar,
3. Menerapkan non-maximum suppression untuk menghilangkan nilai yang tidak diinginkan,

4. Menerapkan dua threshold untuk menentukan edge potensial,
5. Edge diperoleh berdasarkan hysteresis, dan menghapus semua edge yang tidak penting.

Dengan menambahkan edge mask ini diperoleh mask yang mampu menggambarkan objek dengan lebih akurat. Untuk mempermudah proses selanjutnya, bagian-bagian yang kosong pada mask diisi menggunakan fungsi `imfill()` pada matlab. Hasil akhir dari pembuatan mask ini dapat dilihat pada gambar paling kanan pada Gambar 6.



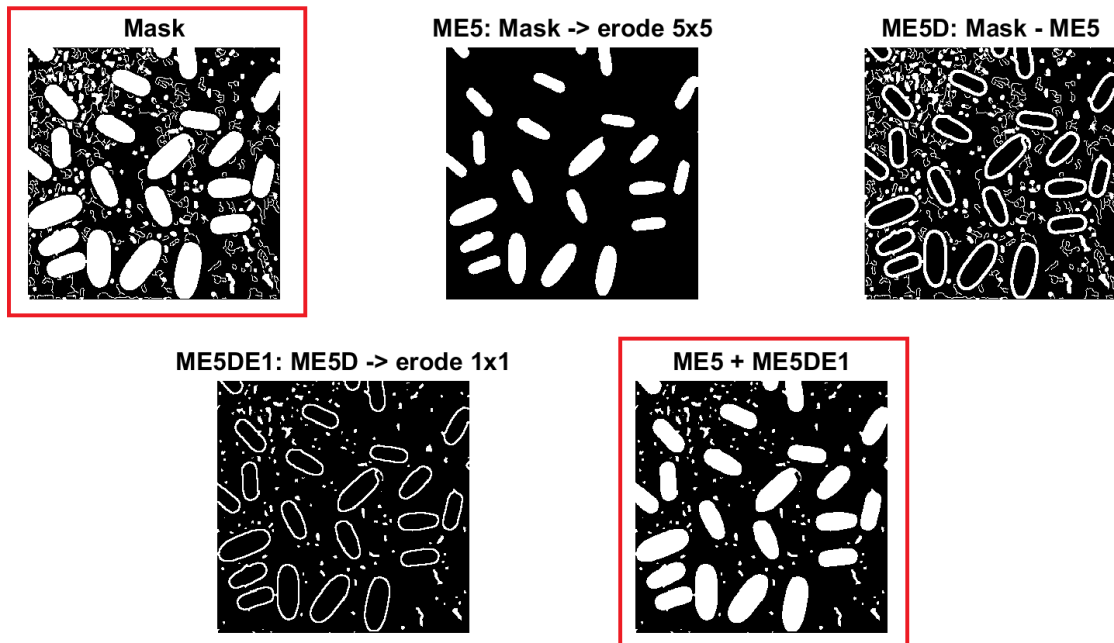
*Gambar 6. Pembuatan Mask1 berdasarkan metode Otsu, Otsu+Edge dan Otsu+Edge+Filling*

Setelah pembuatan mask, dilakukan operasi morfologi untuk menghilangkan artifak yang tidak diinginkan. Pada proses ini, penulis membuat metode Erode khusus yang merupakan gabungan dari dua proses erode.

Algoritma Erode-S dapat dilihat sebagai berikut:

1. Menghitung ME, yaitu hasil erode pada gambar mask dengan ukuran filter yang diinginkan,
2. Menghitung MED, yaitu perbedaan nilai ME dengan gambar mask,
3. Melakukan erode dengan filter 1x1 pada MED, sehingga didapat MEDE,
4. Melakukan dilate pada ME dengan filter 1x1,
5. Menggabungkan nilai ME dengan MEDE.

Hasil dari proses ini tampak pada Gambar 7, terlihat bahwa dengan menggunakan metode ini, bagian-bagian penting dari objek tidak banyak hilang.

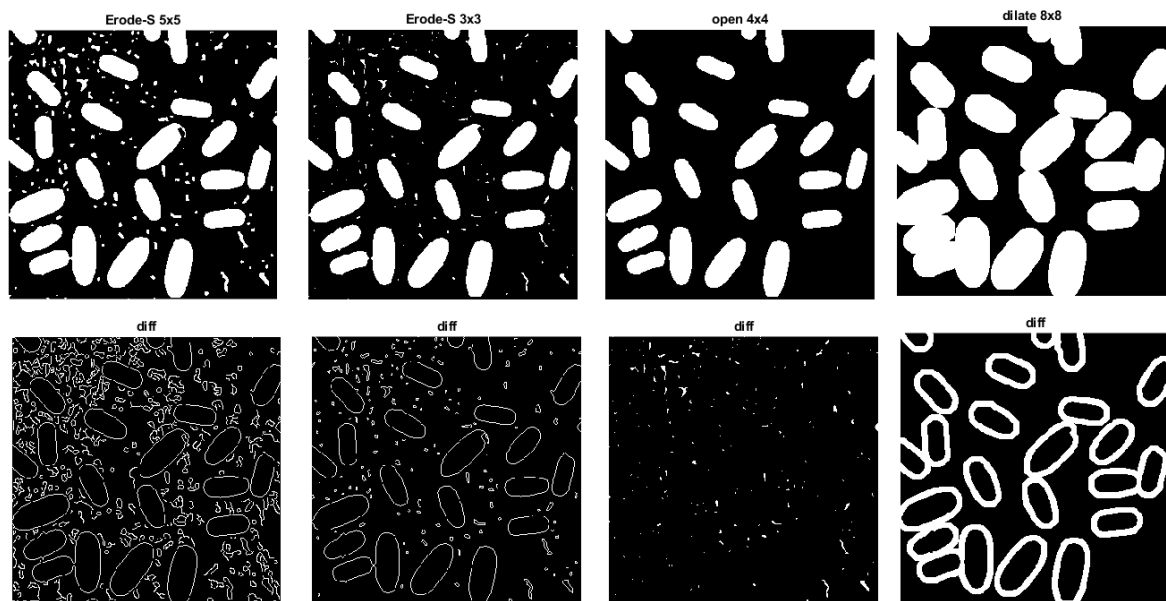


Gambar 7. Metode Erode S, untuk menghilangkan artifak halus pada objek

Dalam Proses1, operasi morfologi yang digunakan secara berurutan adalah:

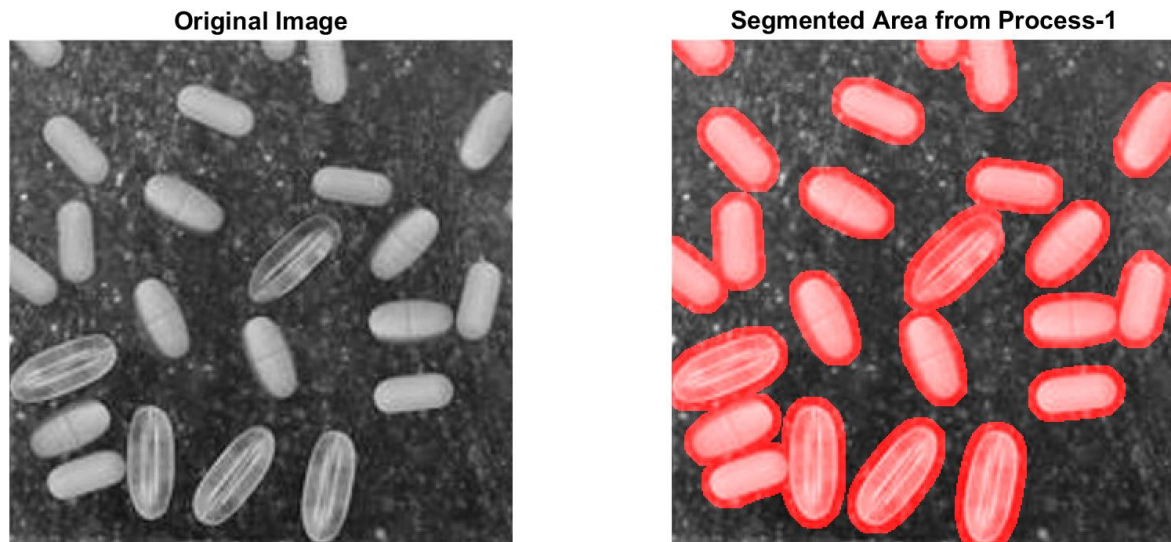
1. Erode-S 5x5
2. Erode-S 3x3
3. Open 4x4
4. Dilate 8x8

Dengan hasil yang dapat dilihat pada Gambar 8, dimana semua artifak dari latar belakang berhasil dihilangkan, dan bentuk kasar dari objek berhasil didapatkan.



Gambar 8. Operasi morfologi pada Mask1(atas = mask yang dihasilkan dari setiap proses, bawah = perbedaan dengan mask sebelumnya)

Mask1 yang didapatkan pada Proses1 apabila ditumpukkan pada gambar, didapat hasil sebagai berikut:



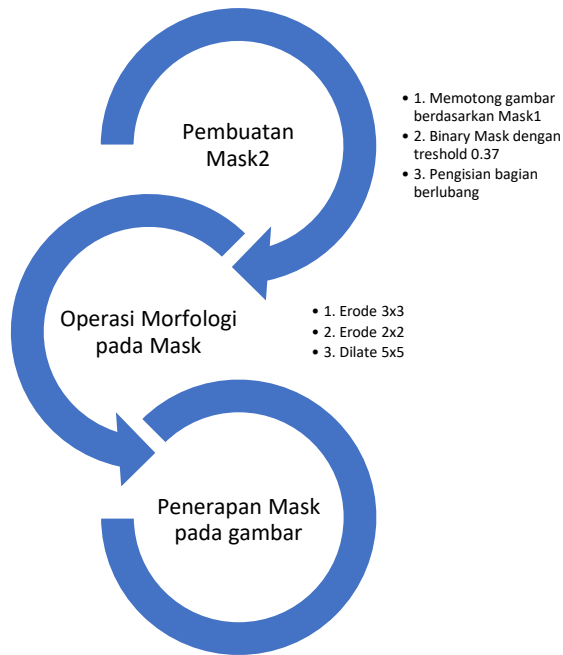
*Gambar 9. Hasil segmentasi pada Proses1*

Pada proses ini, penulis sengaja membuat mask segmentasi yang lebih besar dari objek. Adapun alasannya karena dalam proses ini objek kapsul pada tengah gambar belum dapat dipisahkan secara akurat. Hal ini diakibatkan oleh sifat objek yang semi-transparan pada gambar.

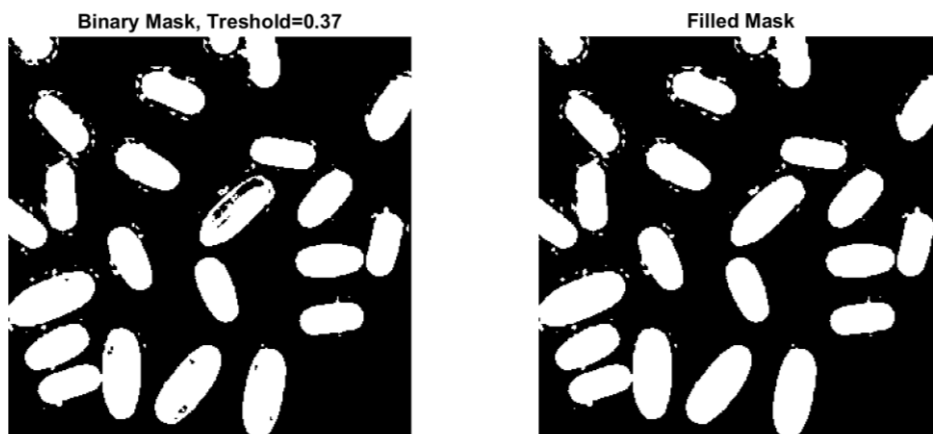
### **3.2. Proses 2**

Pada proses kedua, dibuat mask yang lebih detail untuk mendeteksi bentuk kapsul secara lebih akurat. Proses ini dimulai dengan menerapkan Mask1 pada gambar asli. Gambar yang didapat kemudian diubah menjadi mask biner menggunakan treshold 0,37. Nilai ini didapatkan dari nilai histogram dari kapsul, dimana nilai sebarannya adalah pada nilai 50-100. Adapun  $100/255 = 0,392$  dengan sedikit kalibrasi dan eksperimen, didapatkan nilai 0,37 memberikan hasil terbaik untuk pembuatan mask biner pada tahap ini.

Pada Gambar 11, diuperlihatkan mask yang dihasilkan menggunakan nilai treshold ini. Terlihat bahwa bentuk kapsul dapat digambarkan lebih jelas, selain itu mask ini memiliki artifak latar belakang yang jauh lebih sedikit dibandingkan pada Proses1, karena bagian tersebut sudah dipotong pada Proses1. Kemudian, mask yang didapatkan disempurnakan dengan mengisi bagian yang kosong.



Gambar 10. Diagram Alur pada Proses2



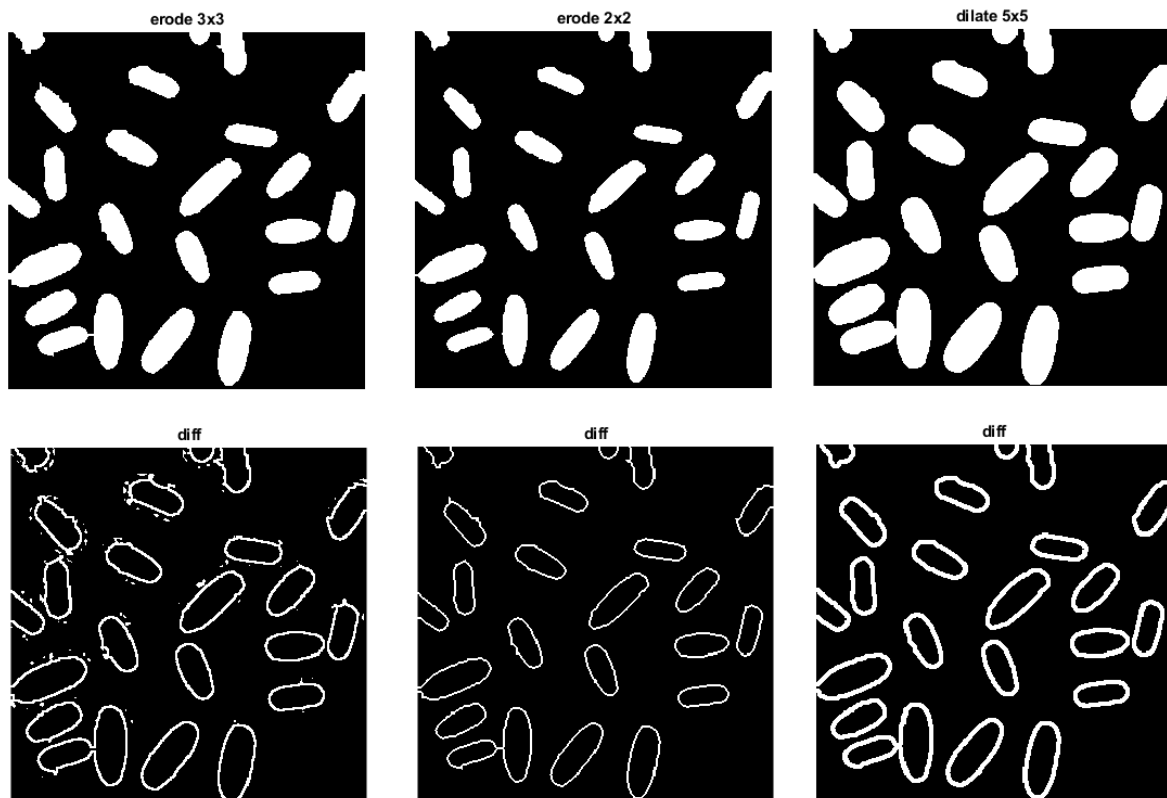
Gambar 11. Pembuatan Mask2, menggunakan threshold 0,37 dan filling.

Proses selanjutnya adalah operasi morfologi pada mask yang diperoleh. Adapun urutan operasi yang dilakukan adalah

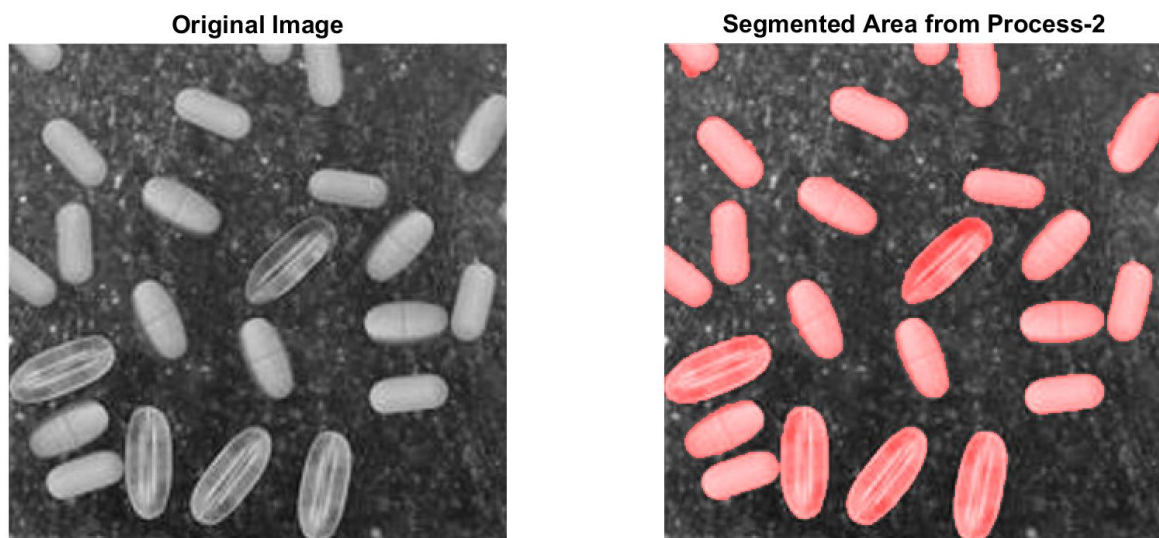
1. Erode 3x3,
2. Erode 2x2,
3. Dilate 5x5,

Hasil dari operasi ini dapat dilihat pada Gambar 12, dimana bentuk semua objek dapat dikenali lebih akurat dibandingkan pada Proses1. Adapun hasil penerapan segmentasi ini pada gambar ditunjukkan pada Gambar 13.





*Gambar 12. Operasi morfologi pada Mask2(atas = mask yang dihasilkan dari setiap proses, bawah = perbedaan dengan mask sebelumnya)*

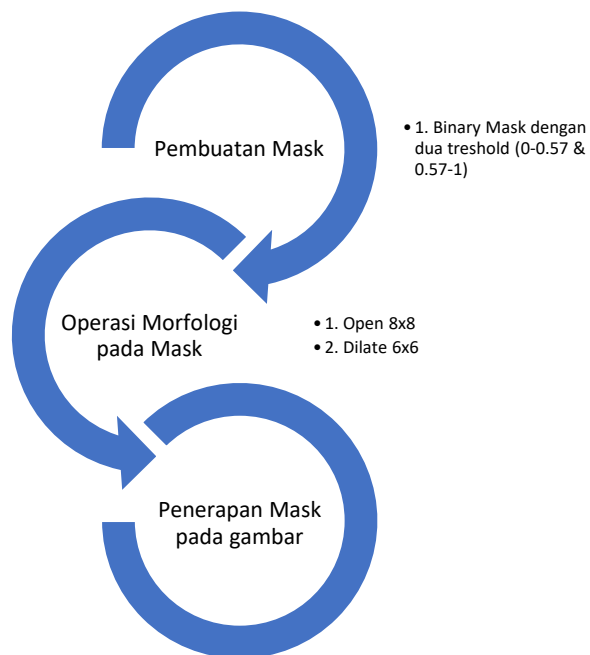


*Gambar 13. Hasil segmentasi pada Proses2*

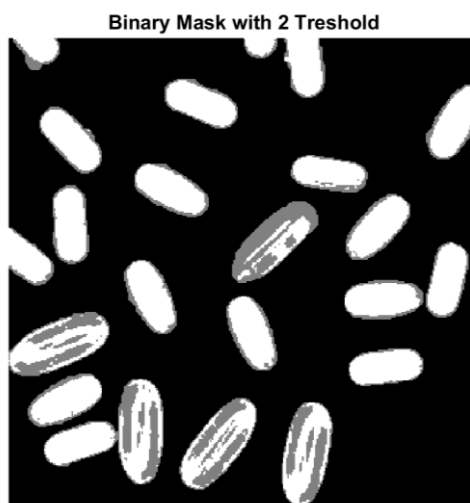
### 3.3. Proses 3

Proses ketiga, merupakan proses tambahan untuk memisahkan antara kapsul dengan tablet. Berdasarkan histogram pada Gambar 3, nilai atas dari tablet adalah 150. Dimana  $150/255$  adalah 0,588. Dengan eksperimen dan kalibrasi, didapat nilai 0,57 memberikan hasil yang terbaik.

Dengan treshhold tersebut dibuat binary mask dengan nilai 0-0.57 dan 0.57-1. Mask yang didapat ditunjukkan pada Gambar 15.



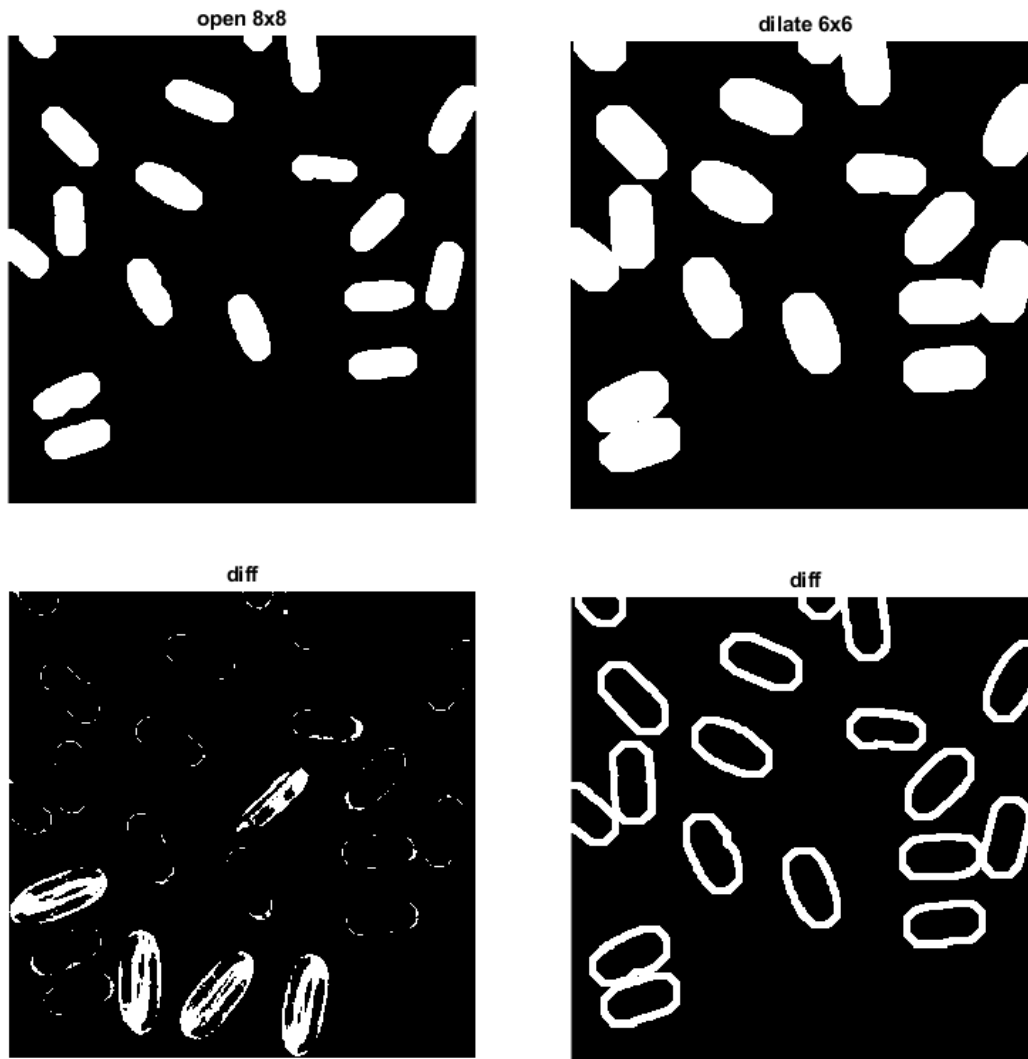
*Gambar 14. Diagram Alur Proses3*



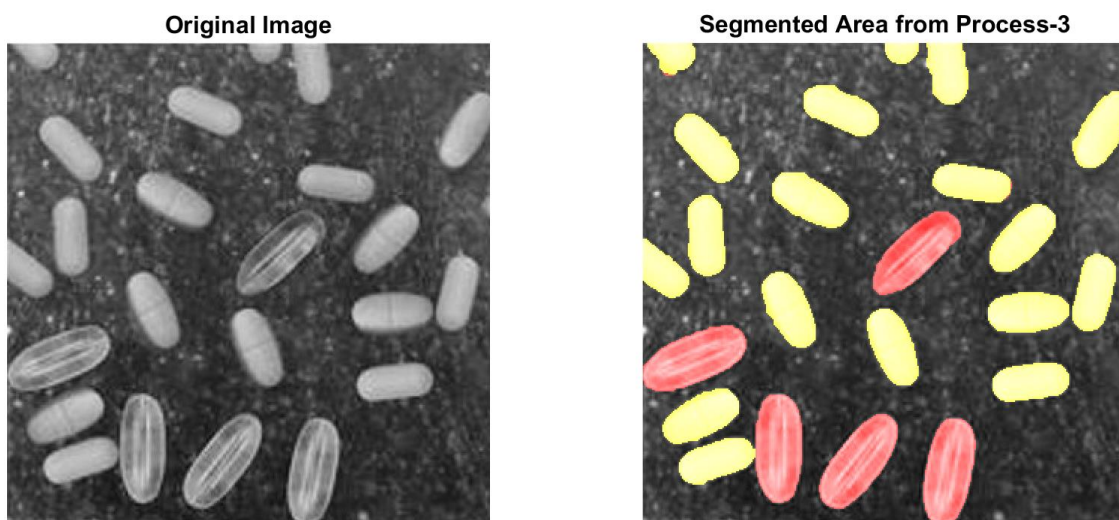
*Gambar 15. Binary Mask3*

Terlihat bahwa pada bagian tablet hampir seluruh bagiannya terisi warna putih, sedangkan pada kapsul hanya sebagian saja. Untuk menghapus bagian dari kapsul dilakukan proses morfologi sebagai berikut:

1. Open 8x8
2. Dilate 6x6

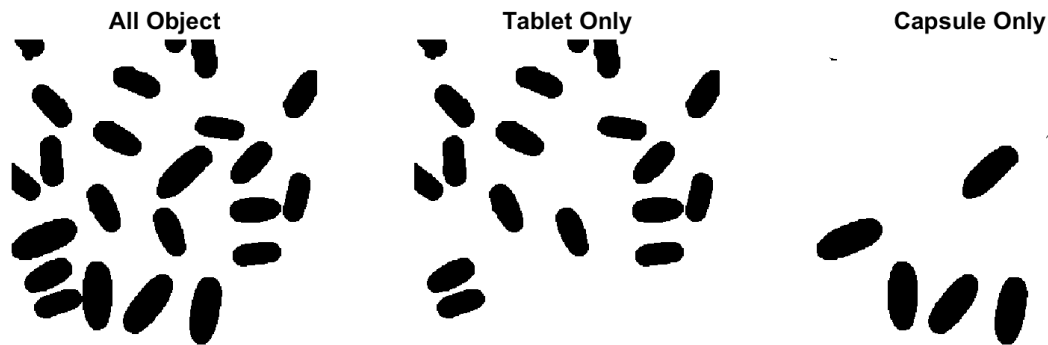


Gambar 16. Operasi morfologi pada Mask2(atas = mask yang dihasilkan dari setiap proses, bawah = perbedaan dengan mask sebelumnya)



Gambar 17. Hasil segmentasi pada Proses3

Menggunakan semua mask yang didapatkan, diperoleh hasil akhir sebagai berikut:



*Gambar 18. Hasil akhir dari segmentasi gambar*

#### **4. Kesimpulan**

Proses segmentasi berhasil dilakukan untuk memisahkan tablet dan kapsul dengan latar belakang. Adapun proses segmentasi dilakukan menggunakan dengan membuat Mask biner berdasarkan treshold yang berbeda-beda, yaitu 0,477 (otsu); 0,37 dan 0,57. Beberapa proses lain juga ditambahkan untuk mendapatkan Mask yang baik, yaitu edge detection menggunakan metode Canny dan holes filling.

Untuk menghilangkan artifak pada Mask, dilakukan operasi morfologi. Penulis juga menerapkan operasi morfologi khusus “Erode-S” untuk menghilangkan artifak halus. Keseluruhan proses eksperimen dilakukan dalam tiga tahap, yaitu pembuatan Mask1 yang memisahkan objek dengan latar belakang secara kasar. Diikuti tahap kedua yang membuat mask lebih detail dan terakhir pembuatan mask untuk memisahkan dua tipe objek yang berbeda.

#### **5. Daftar Pustaka**

- [1] N. Otsu, “THRESHOLD SELECTION METHOD FROM GRAY-LEVEL HISTOGRAMS.,” *IEEE Trans Syst Man Cybern*, 1979.
- [2] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, 1986.

## 6. Lampiran

```
% Image Segmentation
% By: Susaf N.A
% Computer Vision Class
% March 2020 MTI UGM

%% Inizialization and data acquisition
clc; %clearing command window
clear all; close all; %clear all variable and close all window
fontSize = 24; %label font size

% load image name with the extention type
filename = 'Picture1.jpg';

% Read image
imgOri = imread(filename);
imgCrop1 = imread('crop1.png');
imgCrop2 = imread('crop2.png');
I = imgOri(:,:,1);

% Convert to double
dRGB_Ori = im2double(imgOri);
dRC_Ori = dRGB_Ori(:,:,1); %double Red Channel - Original value
dGC_Ori = dRGB_Ori(:,:,2); %double Green Channel - Original value
dBC_Ori = dRGB_Ori(:,:,3); %double Blue Channel - Original value

%% Histogram Analisis

[counts1,x1] = imhist(imgOri,16);
[counts2,x2] = imhist(imgCrop1,16);
[counts3,x3] = imhist(imgCrop2,16);

subplot(2,3,1);stem(x1,counts1,'LineWidth',4);title('Histogram of Whole
Image', 'FontSize', fontSize);
subplot(2,3,2);stem(x2,counts2,'LineWidth',4);title('Histogram of Tablet
Image', 'FontSize', fontSize);
subplot(2,3,3);stem(x3,counts3,'LineWidth',4);title('Histogram of Capsule
Image', 'FontSize', fontSize);
subplot(2,3,4);imshow(imgOri);title('Whole Image', 'FontSize', fontSize);
subplot(2,3,5);imshow(imgCrop1);title('Tablet Image', 'FontSize',
fontSize);
subplot(2,3,6);imshow(imgCrop2);title('Capsule Image', 'FontSize',
fontSize);

%% PROCESS 1
% Binary Mask
Tlvl_1 = graythresh(imgOri); %otsu threshold = 0.477
BW = im2bw(imgOri, Tlvl_1);
BW_mask = BW(:,:,1)==1;

% Edge Detection
Edge_Canny = edge(I,'Canny'); %using Canny

% Merging mask
img_BE = BW_mask | Edge_Canny; %image with binary color and edge
detection
% Filling holes
img_BEf = imfill(img_BE,'holes'); %fill the empty area
```

```

figure;
subplot(1,3,1);imshow(BW_mask);title('Binary Mask, Otsu Treshold',
'FontSize', fontSize);
subplot(1,3,2);imshow(img_BE);title('Binary Mask + Edge Canny',
'FontSize', fontSize);
subplot(1,3,3);imshow(img_BEf);title('Binary Mask + Edge Canny + Fill',
'FontSize', fontSize);

% Testing of "Erode-S" Method
%{
sel = strel('disk', 5);
P1 = imerode(img_BEf,sel);
P1_diff = img_BEf~= P1;
sel = strel('disk', 1);
P2 = imerode(P1_diff,sel);
P3 = imdilate(P1,sel);
img_merge = P2 | P3;

figure;
subplot(2,3,1);imshow(img_BEf);title('Mask', 'FontSize', fontSize);
subplot(2,3,2);imshow(P1);title('ME5: Mask -> erode 5x5', 'FontSize',
fontSize);
subplot(2,3,3);imshow(P1_diff);title('ME5D: Mask - ME5', 'FontSize',
fontSize);
subplot(2,3,4);imshow(P2);title('ME5DE1: ME5D -> erode 1x1', 'FontSize',
fontSize);
subplot(2,3,5);imshow(img_merge);title('ME5 + ME5DE1', 'FontSize',
fontSize);
%}

% morphological process
%1 = erode
%2 = dilate
%3 = open (ed)
%4 = close (de)
%5 = special
morph_sequence = [5 5 3 2];
morp_val = [5 3 4 8];

img_sgmnt = img_morph(img_BEf, morph_sequence, morp_val); %morphological
function
inv_sgmnt = ~img_sgmnt;

% Apply mask to original image (dummy)
imgProc1 = imgOri;
A = (dRC_Ori .* img_sgmnt)>0;
An = A * 255;
B = (dRC_Ori .* inv_sgmnt) * 255;
imgProc1(:,:,1) = An + B;

figure;
subplot(1,2,1);imshow(imgOri);title('Original Image','FontSize',
fontSize);
subplot(1,2,2);imshow(imgProc1);title('Segmented Area from Process-
1','FontSize', fontSize);
%% PROCESS 2
% Apply previous mask to original image
imgRC_masked = dRC_Ori .* img_sgmnt;
imgGC_masked = dGC_Ori .* img_sgmnt;

```

```

imgBC_masked = dBC_Ori .* img_sgmnt;
imgMasked = cat(3, imgRC_masked, imgGC_masked, imgBC_masked);

Tlvl_2 = 0.37;%graythresh(imgOri);
BW_mask = im2bw(imgMasked, Tlvl_2);
img_OF = imfill(BW_mask, 'holes');

figure;
subplot(1,2,1);imshow(BW_mask);title(['Binary Mask,
Treshold=', num2str(Tlvl_2)], 'FontSize', fontSize);
subplot(1,2,2);imshow(img_OF);title('Filled Mask', 'FontSize', fontSize);

morph_sequence = [1 1 2];
morp_val = [3 2 5];
img_sgmnt = img_morph(img_OF, morph_sequence, morp_val);
inv_sgmnt = ~img_sgmnt;

imgProc2 = imgOri; %overlay to original image
A = (dRC_Ori .* img_sgmnt)>0;
An = A * 255;
B = (dRC_Ori .* inv_sgmnt) * 255;
imgProc2(:,:,1) = An + B;

figure;
subplot(1,2,1);imshow(imgOri);title('Original Image', 'FontSize',
fontSize);
subplot(1,2,2);imshow(imgProc2);title('Segmented Area from Process-
2', 'FontSize', fontSize);

close all;
%% PROCESS 3
% Apply previous mask to original image
imgRC_masked = dRC_Ori .* img_sgmnt;
imgGC_masked = dGC_Ori .* img_sgmnt;
imgBC_masked = dBC_Ori .* img_sgmnt;
imgMasked = cat(3, imgRC_masked, imgGC_masked, imgBC_masked);

% Double Treshold
Tlvl_3 = 0.57;
% low value
imgRes1(:,:,1)=(imgMasked(:,:,1) > 0 & imgMasked(:,:,1) < Tlvl_3)*0.5;
imgRes1(:,:,2)=(imgMasked(:,:,2) > 0 & imgMasked(:,:,2) < Tlvl_3)*0.5;
imgRes1(:,:,3)=(imgMasked(:,:,3) > 0 & imgMasked(:,:,3) < Tlvl_3)*0.5;
% high value
imgRes2(:,:,1)=imgMasked(:,:,1) > Tlvl_3;
imgRes2(:,:,2)=imgMasked(:,:,2) > Tlvl_3;
imgRes2(:,:,3)=imgMasked(:,:,3) > Tlvl_3;

imgNew = im2double(imgRes1) + im2double(imgRes2);

figure;
imshow(imgNew);title('Binary Mask with 2 Treshold', 'FontSize', fontSize);

% only take the high value
img_OF = imgRes2(:,:,1);

% morphological process
%1 = erode
%2 = dilate

```

```

%3 = open (ed)
%4 = close (de)
%5 = special
morph_sequence = [3 2];
morp_val = [8 6];
img_sgmnt2 = img_morph(img_OF, morph_sequence, morp_val);
img_sgmnt3 = img_sgmnt2 & img_sgmnt;
inv_sgmnt3 = ~img_sgmnt3;

% Apply mask to original image (dummy)
imgProc3 = imgProc2; %overlay to previous result
A = (dBC_Ori .* img_sgmnt3)>0;
An = A * 255;
B = (dBC_Ori .* inv_sgmnt3) * 255;
imgProc3(:,:,2) = An + B;

figure;
subplot(1,2,1);imshow(imgOri);title('Original Image','FontSize',
fontSize);
subplot(1,2,2);imshow(imgProc3);title('Segmented Area from Process-
3','FontSize', fontSize);

close all;
%% Final Result

img_all = inv_sgmnt;
img_tablet = inv_sgmnt3;
img_capsule = ~(img_sgmnt - img_sgmnt3);

figure;
subplot(1,2,1);imshow(imgOri);title('Original Image','FontSize',
fontSize);
subplot(1,2,2);imshow(imgProc3);title('Segmented Image','FontSize',
fontSize);
figure;
subplot(1,3,1);imshow(img_all);title('All Object','FontSize', fontSize);
subplot(1,3,2);imshow(img_tablet);title('Tablet Only','FontSize',
fontSize);
subplot(1,3,3);imshow(img_capsule);title('Capsule Only','FontSize',
fontSize);

% Image Morpholgycal Operation
% By: Susaf N.A
% Computer Vision Class
% March 2020 MTI UGM

function [out_img] = img_morph(imgBW, morph_sequence, morp_val)
%imageRestoration function: generating restored image using median filter
% -Usage-
% [out_img]: imageMorphology(imgBW, morph_sequence,morp_valn)
%
% -Inputs-
% imgBW: original binary image
% morph_sequence: morphology methods sequence (1:erode, 2:dilate,
3:Open, 4:Close,
% 5: Erode-S)
% morp_val: filter size
%
% -Outputs-
% out_img: processed image
%Author: Susaf Noor Azhar, University of Gadjah Mada

```



```

[C, R] = size(morph_sequence); %calculate the task size

spC = 2; %subplot column
spR = R; %subplot row

imgProc = imgBW;
figure();

for i=1:R

    ms = morph_sequence(i); %assign the task
    mv = morph_val(i); %assign the value
    sel = strel('disk', mv); %create the filter
    imgprev = imgProc; %previous image

    if ms == 1 %erode
        imgProc = imerode(imgProc,sel);
        t = ['erode ', num2str(mv), 'x', num2str(mv)];
    elseif ms == 2 %dilate
        imgProc = imdilate(imgProc,sel);
        t = ['dilate ', num2str(mv), 'x', num2str(mv)];
    elseif ms == 3 %open
        imgProc = imopen(imgProc,sel);
        t = ['open ', num2str(mv), 'x', num2str(mv)];
    elseif ms == 4 %close
        imgProc = imclose(imgProc,sel);
        t = ['close ', num2str(mv), 'x', num2str(mv)];
    elseif ms == 5 %erode Special
        % erode by val
        sel = strel('disk', mv);
        P1 = imerode(imgProc,sel);
        P1_diff = imgProc ~= P1;

        % erode the difference by 1x1
        sel = strel('disk', 1);
        P2 = imerode(P1_diff,sel);
        % dilate the P1 by 1x1
        P3 = imdilate(P1,sel);
        imgProc = P2 | P3;

        t = ['Erode-S ', num2str(mv), 'x', num2str(mv)];
    end

    imgDiff = imgprev ~= imgProc; %difference with previous image

    subplot(spC,spR,i);imshow(imgProc);title(t);
    subplot(spC,spR,i+R);imshow(imgDiff);title('diff');
end
out_img = imgProc;
end

```