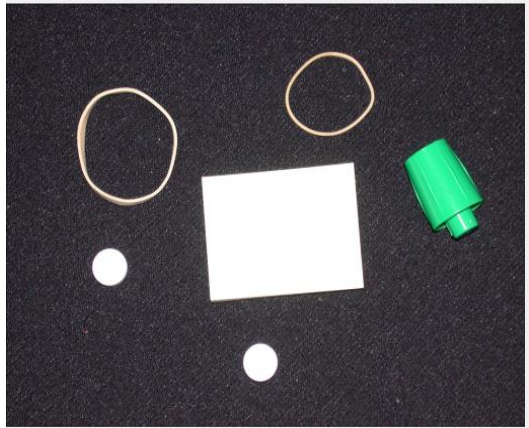
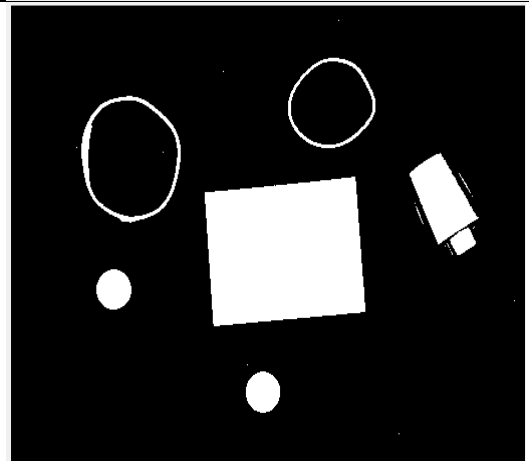
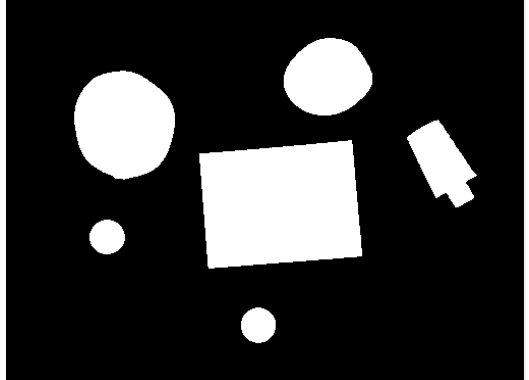
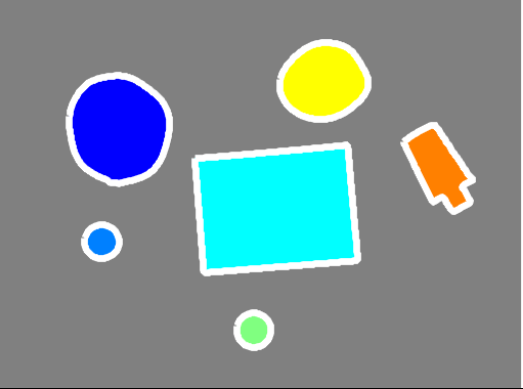
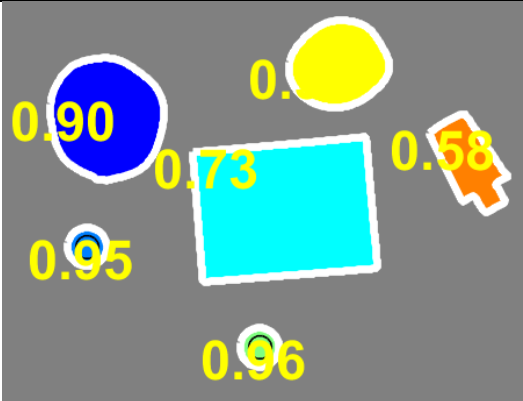


TUTORIAL 4.1: MATLAB IMAGE PROCESSING TOOLBOX—A GUIDED TOUR

- Identifying Round Objects Demo

Penjelasan blok	Gambar
Berfungsi untuk menampilkan gambar “pillsetc” dari folder C:\Program Files\MATLAB\R2015b\toolbox\images\imdata .	 A photograph of several objects on a black background: two yellow rubber bands, a white square, a green plastic cap, and three small white circular pills.
Mengubah warna gambar menjadi hitam dan putih	 A grayscale version of the same image, where the objects are white and the background is black.
Berfungsi menghapus objek berukuran lebih sedikit dari 30 pixel, mengisi <i>gap</i> pada botol agar menjadi satu kesatuan, dan mengisi ruang kosong pada gambar dengan warna putih.	 A binary (black and white) image where the objects are white and the background is black. The small white pills have been removed, and the gaps within the larger objects (like the bottle cap) have been filled in.

<p>Berfungsi untuk mendeteksi objek yang tidak ada lubangnya (sudah menyatu menjadi satu objek) kemudian mewarnai objek tersebut dan mengubah latar dari gambar menjadi warna abu-abu. Lalu memberikan batas berupa garis di tepi dari objek tersebut dengan warna putih.</p>	
<p>Berfungsi untuk menandai setiap gambar dengan tingkat kecocokan dengan lingkaran sempurna dan membuat judul pada gambar yang ditampilkan.</p>	

Pada demo ini dapat dilihat bahwa image processing toolbox ini memiliki kemampuan untuk mendeteksi dari suatu objek dan menghitung tingkat kesamaan dari objek tersebut.

- Deblurring Images Using a Wiener Filter

Penjelasan kode	Gambar
<p>Berfungsi untuk menampilkan gambar “cameraman.tif” dari folder C:\Program Files\MATLAB\R2015b\toolbox\images\imdata.</p>	<p>Original Image (courtesy of MIT)</p> 
<p>Berfungsi mengganti gambar dengan gambar yang seolah-olah pada saat pemotretan kamera/objek sedang bergerak yang menyebabkan gambar tidak fokus.</p>	<p>Blurred Image</p> 
<p>Berfungsi untuk mengembalikan keaslian gambar</p>	<p>Restored Image</p> 

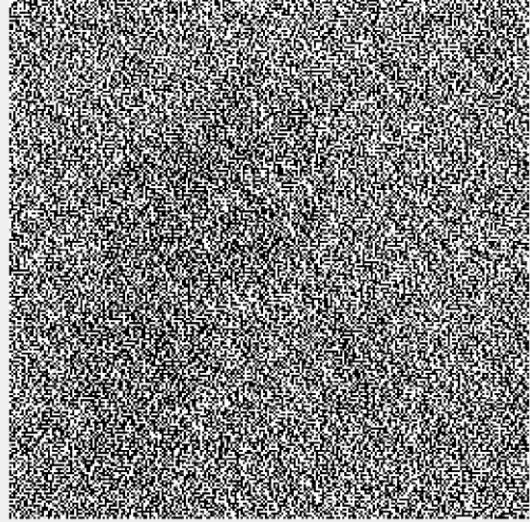
Berfungsi untuk menambahkan noise pada gambar dan memblurkan gambar.

Simulate Blur and Noise



Gambar sebelumnya berusaha dikembalikan dengan mengatakan noise tidak ada pada gambar.

Restoration of Blurred, Noisy Image Using $NSR = 0$



Pada blok ini gambar berusaha dikembalikan dengan menambahkan NSR yang sudah diperkirakan.

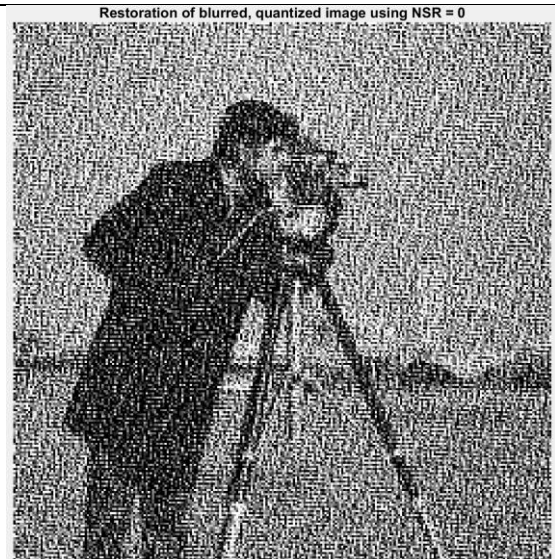
Restoration of Blurred, Noisy Image Using Estimated NSR



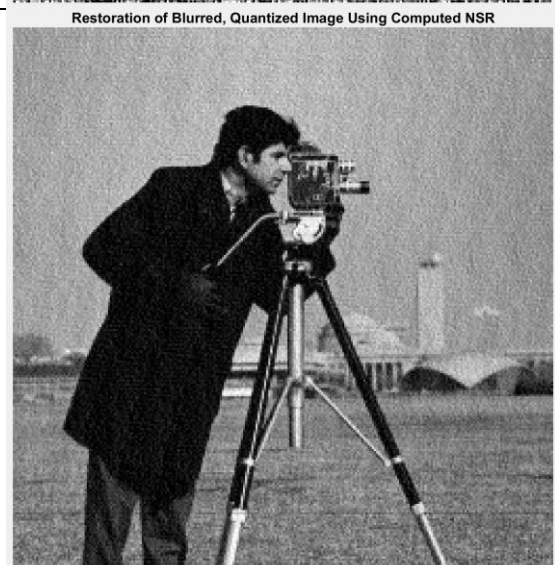
Pada blok ini, nilai noise dikuantisasi .

-

Kemudian lakukan proses pengembalian gambar dengan proses tanpa noise.

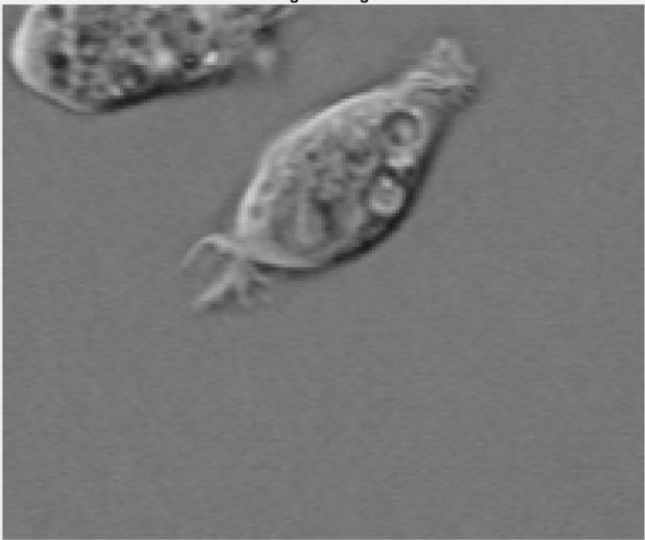
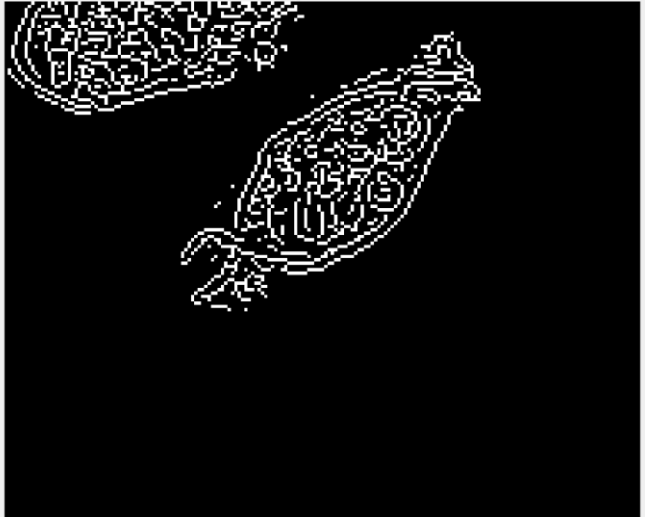


Lakukan pengembalian gambar dengan memasukkan NSR yang sudah diestimasi



Pada demo ini dapat dilihat bahwa image processing toolbox ini memiliki kemampuan untuk memblurkan dan memperjelas gambar suatu objek.

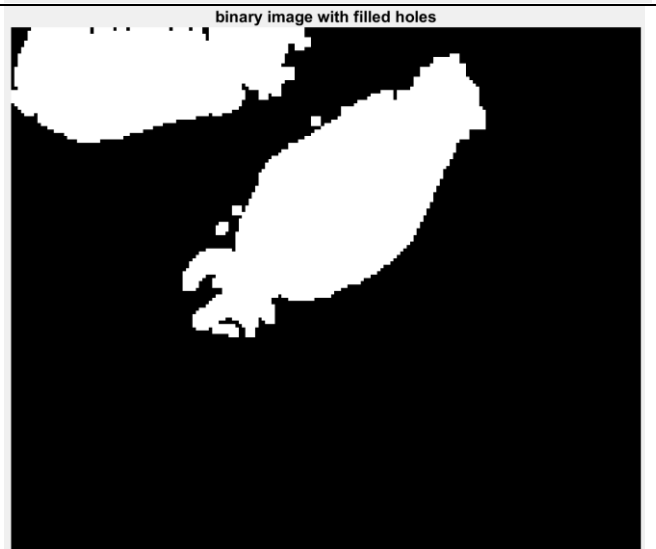
- Detecting a Cell Using Image Segmentation

Penjelasan kode	Hasil Gambar
<p>Berfungsi untuk membaca dan menampilkan gambar “cell.tif” dari folder C:\Program Files\MATLAB\R2015b\toolbox\images\imdat</p> <p>a. Blok ini juga berfungsi untuk menambahkan keterangan gambar.</p>	<p>original image</p>  <p>Image courtesy of Alan Partin Johns Hopkins University</p>
<p>Berfungsi mendeteksi perbedaan kontras antara objek dengan latar belakang gambar.</p>	<p>binary gradient mask</p> 

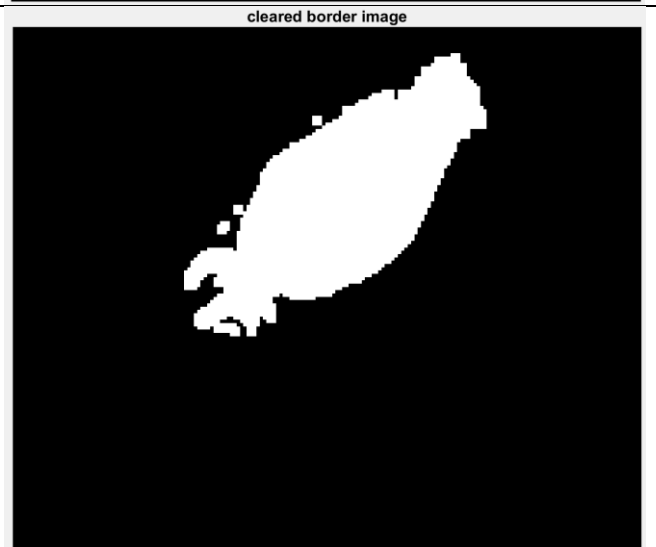
Berfungsi untuk menyatukan *gap* yang ada antara 2 garis pada objek.



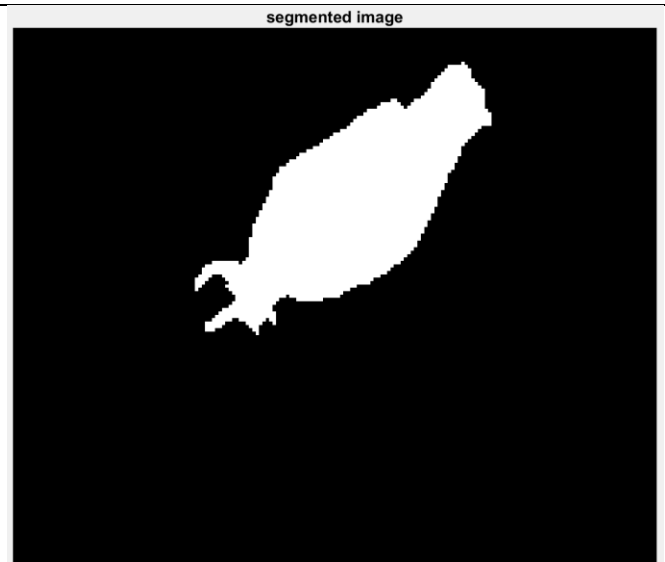
Karena masih ada *gap* pada objek, isi ruang kosong dengan warna putih agar objek menjadi satu kesatuan dan tidak ada ruang kosong pada objek.



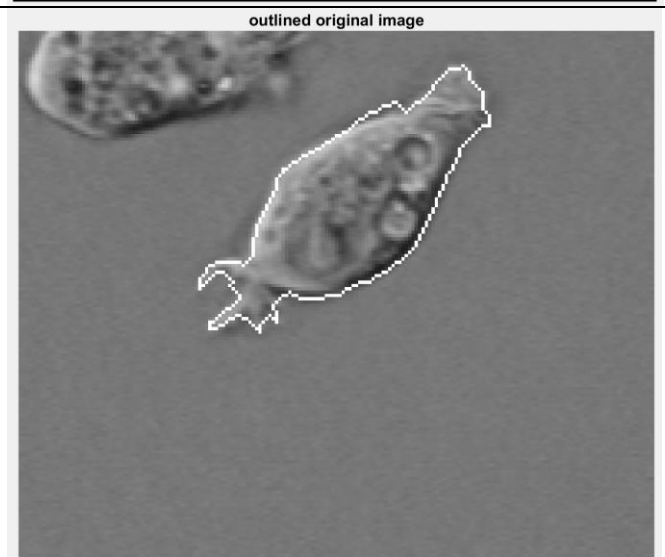
Hilangkan objek yang mengenai batas dari gambar.



Untuk memperhalus gambar, ubah elemen pembentuk objek seperti berlian. Lakukan proses ini 2 kali untuk membuat gambar tampak asli.



Mengambil garis tepi dari objek proses sebelumnya dan menimpa dengan gambar asli.



Pada demo ini dapat dilihat bahwa image processing toolbox ini memiliki kemampuan untuk mengsegmentasikan gambar untuk diambil garis tepi dari objek.

TUTORIAL 4.2: BASIC IMAGE MANIPULATION

- Question 1

Tipe dari gambar adalah PNG , yang menggunakan *color-based* untuk menginterpretasikan gambar.

- Question 2

Agar data/gambar yang akan diproses dapat masukkan ke variabel yang dibuat tanpa menampilkan data tersebut di command windows. Dengan menghilangkan tanda semicolon (;) , data/gambar akan disimpan dan ditampilkan ke command window.

The image shows two MATLAB variable viewers side-by-side. The left window, titled 'X', displays a 258x350 uint8 matrix. The right window, titled 'map', displays a 256x3 double matrix.

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	108	116	98	108	87	90	90	86	90	91	90	86	81
2	112	112	112	105	112	105	112	105	105	105	93	93	9
3	112	112	105	112	112	105	105	105	105	93	93	105	9
4	105	105	105	105	105	112	105	105	93	105	105	93	10
5	112	105	112	105	105	105	105	105	105	105	105	105	9
6	105	105	105	112	105	112	105	105	105	105	105	105	9
7	105	112	105	105	112	105	112	105	105	105	93	105	10
8	105	105	112	112	112	105	105	105	105	105	93	9	
9	105	105	112	114	117	112	112	105	105	105	105	9	
10	117	120	123	125	123	112	112	112	105	105	105	10	

	1	2	3	4	5
1	0	0	0	0	
2	0.0627	0.0627	0.0314		
3	0.2902	0.0314	0		
4	0	0	1		
5	0.2902	0.0627	0.0627		
6	0.3882	0.0314	0.0941		
7	0.4510	0.0627	0		
8	0.2588	0.1608	0.0627		
9	0.0941	0.2588	0.0314		
10	0.2235	0.1922	0.2235		

Variabel X

Variabel map

The image shows a MATLAB command window with the following text:

```
X_rgb =
258x350x3 double

val(:,:,1) =

Columns 1 through 5

    0.6118    0.8392    0.5804    0.6118    0.6471
    0.6471    0.6471    0.6471    0.5490    0.6471
    0.6471    0.6471    0.5490    0.6471    0.6471
    0.5490    0.5490    0.5490    0.5490    0.5490
    0.6471    0.5490    0.6471    0.5490    0.5490
    0.5490    0.5490    0.5490    0.6471    0.5490
    0.5490    0.6471    0.5490    0.5490    0.6471
    0.5490    0.5490    0.6471    0.6471    0.6471
    0.5490    0.5490    0.6471    0.7098    0.7412
    0.7412    0.7412    0.8392    0.9059    0.8392
    0.9059    1.0000    1.0000    1.0000    1.0000
    1.0000    1.0000    1.0000    1.0000    1.0000
    1.0000    1.0000    1.0000    1.0000    1.0000
    1.0000    1.0000    1.0000    1.0000    1.0000
    1.0000    1.0000    1.0000    1.0000    1.0000
```

Variabel X_rgb

- Question 3

Variabel X_rgb mempunyai 3 dimensi dengan tipe data yaitu double. Sehingga ukuran memori untuk tipe data double yaitu sebesar 8 byte (64 bit).

X_gray													
258x350 uint8													
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	184	210	168	184	153	156	156	150	156	158	156	150	150
2	198	198	198	179	198	179	198	179	179	179	159	159	159
3	198	198	179	198	198	179	179	179	179	159	159	179	159
4	179	179	179	179	179	198	179	179	159	179	179	159	179
5	198	179	198	179	179	179	179	179	179	179	179	179	159
6	179	179	179	198	179	198	179	179	179	179	179	179	159
7	179	198	179	179	198	179	198	179	179	179	159	179	179
8	179	179	198	198	198	179	179	179	179	179	179	159	159
9	179	179	198	205	213	198	198	179	179	179	179	179	159
10	213	221	233	243	233	198	198	198	179	179	179	179	179
11	243	255	255	255	255	221	205	198	198	179	179	179	179
12	255	255	255	255	255	230	221	205	198	198	198	179	179
13	255	255	255	255	255	243	233	230	233	233	205	198	179
14	255	255	255	255	255	255	255	255	255	243	243	198	179
15	255	255	255	255	255	255	255	255	255	255	243	221	179
16	255	255	255	255	255	255	255	255	255	255	243	213	179
17	255	255	255	255	255	255	255	255	255	255	243	198	179
18	255	255	255	255	255	255	255	255	255	255	230	213	179

Variabel X_gray

- Question 4

Variabel X_gray memiliki tipe data uint8 (unsigned integer 8 bit).

Mencari nilai max dan min dari variabel X_gray

```
>> Baris_5
```

```
ans =
```

```
255
```

```
ans =
```

```
0
```

- Question 5

Operator colon (:) digunakan agar operasi min dan max dilakukan pada setiap cell pada matriks. Jika operator ini dihilangkan maka operasi min dan max akan dioperasikan pada setiap kolom pada matriks.

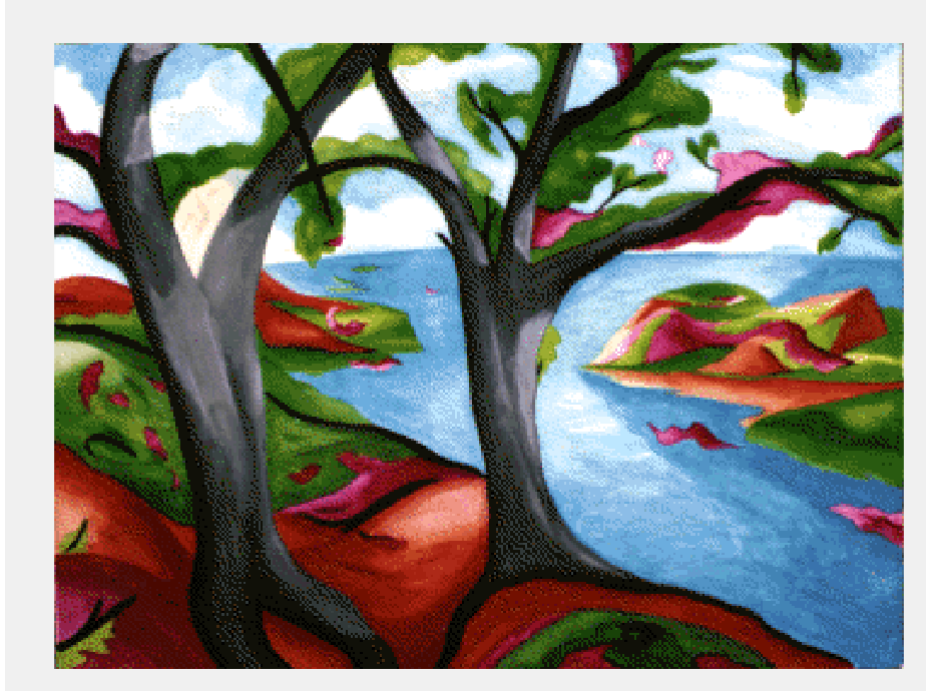
X_gray_dbl																		
258x350 double																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0.7...	0.8...	0.6...	0.7...	0.6...	0.6...	0.6...	0.5...	0.6...	0.6...	0.6...	0.5...	0.5...	0.5...	0.6...	0.6...	0.7...	0.7...
2	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.6...	0.6...	0.6...	0.6...	0.6...	0.7...	0.6...	0.6...
3	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.6...	0.6...	0.7...	0.6...	0.6...	0.6...	0.6...	0.6...	0.6...
4	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.6...	0.7...	0.7...	0.6...	0.7...	0.7...	0.7...	0.6...	0.6...	0.6...
5	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.6...	0.7...	0.6...	0.7...	0.7...	0.6...
6	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.6...	0.7...	0.6...	0.7...	0.6...	0.7...
7	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.6...	0.7...	0.7...	0.7...	0.7...	0.6...	0.7...	0.7...
8	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.6...	0.6...	0.6...	0.7...	0.7...	0.7...	0.7...	0.7...
9	0.7...	0.7...	0.7...	0.8...	0.8...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.6...	0.7...	0.7...	0.7...	0.7...	0.6...
10	0.8...	0.8...	0.9...	0.9...	0.9...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.6...
11	0.9...	1	1	1	1	0.8...	0.8...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...
12	1	1	1	1	1	0.9...	0.8...	0.8...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...	0.7...
13	1	1	1	1	1	0.9...	0.9...	0.9...	0.9...	0.9...	0.8...	0.7...	0.7...	0.6...	0.7...	0.6...	0.7...	0.7...
14	1	1	1	1	1	1	1	1	1	0.9...	0.9...	0.7...	0.7...	0.6...	0.7...	0.7...	0.6...	0.7...
15	1	1	1	1	1	1	1	1	1	1	0.9...	0.8...	0.7...	0.6...	0.7...	0.6...	0.7...	0.6...
16	1	1	1	1	1	1	1	1	1	1	0.9...	0.8...	0.7...	0.7...	0.7...	0.7...	0.7...	0.6...
17	1	1	1	1	1	1	1	1	1	1	0.9...	0.7...	0.7...	0.6...	0.7...	0.6...	0.6...	0.7...
18	1	1	1	1	1	1	1	1	1	1	0.9...	0.8...	0.7...	0.7...	0.6...	0.6...	0.7...	0.6...
19	1	1	1	1	1	1	1	1	1	1	1	0.8...	0.8...	0.7...	0.7...	0.6...	0.7...	0.6...
20	1	1	1	1	1	1	1	1	1	1	1	1	0.9...	0.8...	0.7...	0.7...	0.6...	0.7...
21	1	1	1	1	1	1	1	1	1	1	1	1	1	0.9...	0.9...	0.8...	0.7...	0.7...
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.8...	0.7...	0.7...
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.9...	0.8...	0.7...
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.9...	0.7...
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.8...

- Question 6
Range dari variabel X_gray_dbl antara 0 sampai 1.

Dengan perintah “imshow(I), imixelinfo” , dapat mengeluarkan tampilan berikut:



Dengan perintah “`imshow(X,map)`, `impixelinfo`” , dapat mengeluarkan tampilan berikut:



- Question 7

Dengan membagikan setiap sel dengan perbandingan antara nilai 255 dan 99 yaitu 2,5757 .

Dengan menggunakan perintah dibawah ini :

$$A = X_gray/2,5757;$$

```
Z = X_gray./2,57575757;
```

```
A = X_gray/3;
```

```
max(A(:))
```

```
min(A(:))
```

```
Command Window
```

```
ans =
```

```
85
```

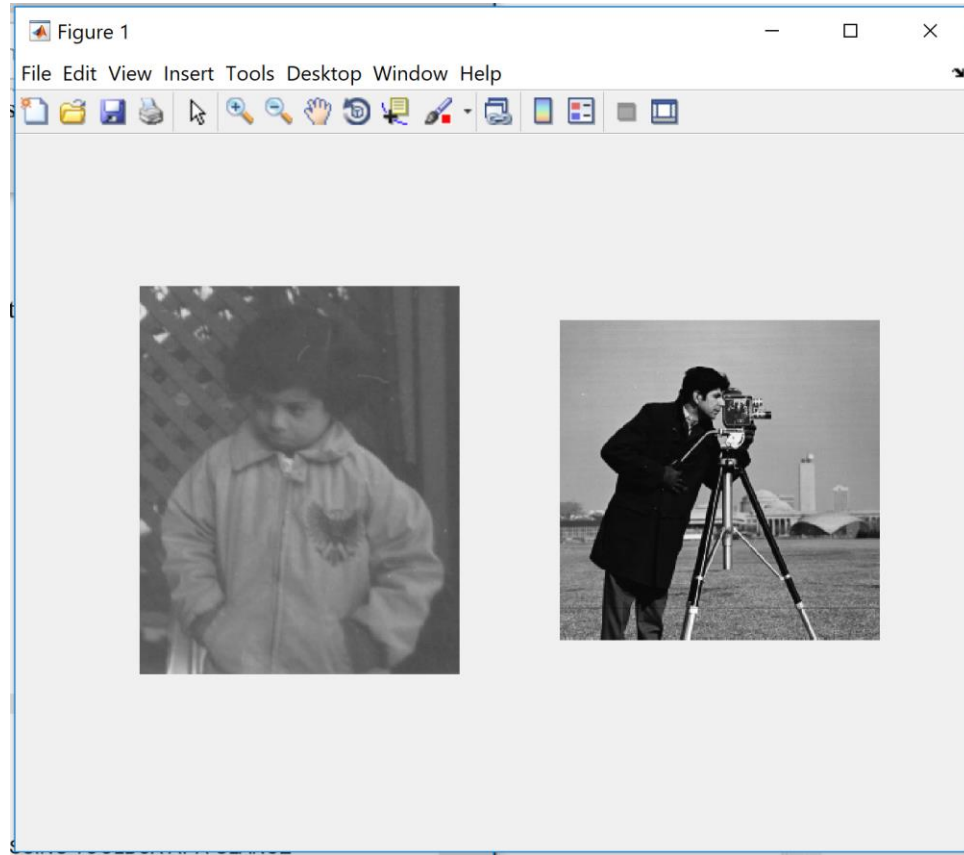
```
ans =
```

```
0
```

Rentang nilai yang didapat dari 0 – 85 dikarenakan nilai pembagi dibulatkan menjadi 3.

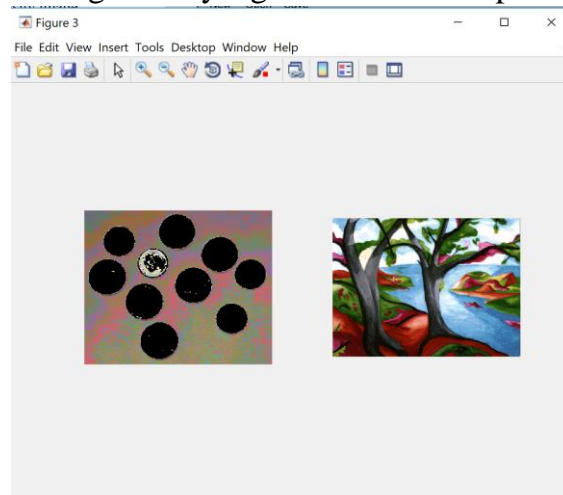
- Question 8
Variabel c, r, p merepresentasikan nilai *column*, *row*, dan *pixel* untuk pixel yang dipilih dari sebuah gambar.

Menampilkan 2 gambar pada 1 frame .

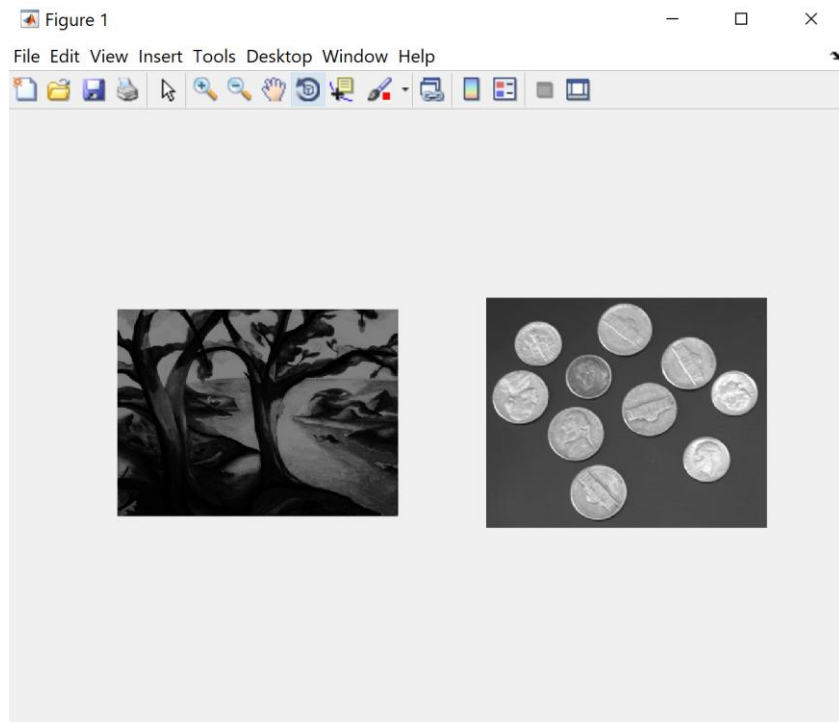


- Question 9
Karena Gambar A dan B bertipe data unsigned integer 8 bit , makarentang nilainya 0 sampai dengan 255.

Menampilkan 2 gambar yang berbeda format pada satu frame.



```
figure
subplot(1,2,1), imshow(I)
subplot(1,2,2), imshow(X,map)
```



```
figure
subplot(1,2,1), imshow(X,map)
subplot(1,2,2), imshow(I)
```

- Question 10

Dari perbandingan 2 perintah dengan menukar 2 urutan dari subplot pada gambar. Dapat disimpulkan gambar pada subplot pertama mengikuti warna dari gambar subplot kedua.

- Question 11

I_ind berisikan indeks warna dari gambar
I_map berisikan color map dari gambar.

TUTORIAL 6.1: ARITHMETIC OPERATIONS

- Question 1

Gambar original		Gambar modifikasi	
min	Max	min	Max
0	255	75	255

Dengan meningkatkan batas bawah dari tingkat kecerahan akan membuat nilai dibawah dari batas yang dibuat ditambah dengan 75, yang mengakibatkan kecerahan dari pixel tersebut akan meningkat.

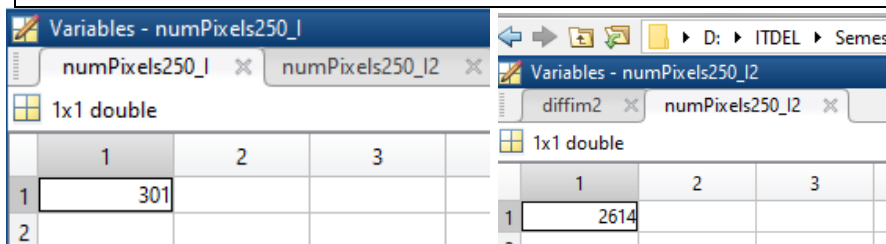
- Question 2

Untuk mengetahui berapa pixel yang bernilai 255, digunakan fungsi sum pada Matlab.

```
I = imread('tire.tif');
I2 = imadd(I,75);
figure
subplot(1,2,1), imshow(I), title('Original Image');
subplot(1,2,2), imshow(I2), title('Brighter Image');

numPixels250_I = sum(sum(I==255));
numPixels250_I2 = sum(sum(I2==255));
```

Hasil:



	1	2	3
1	301		
2			

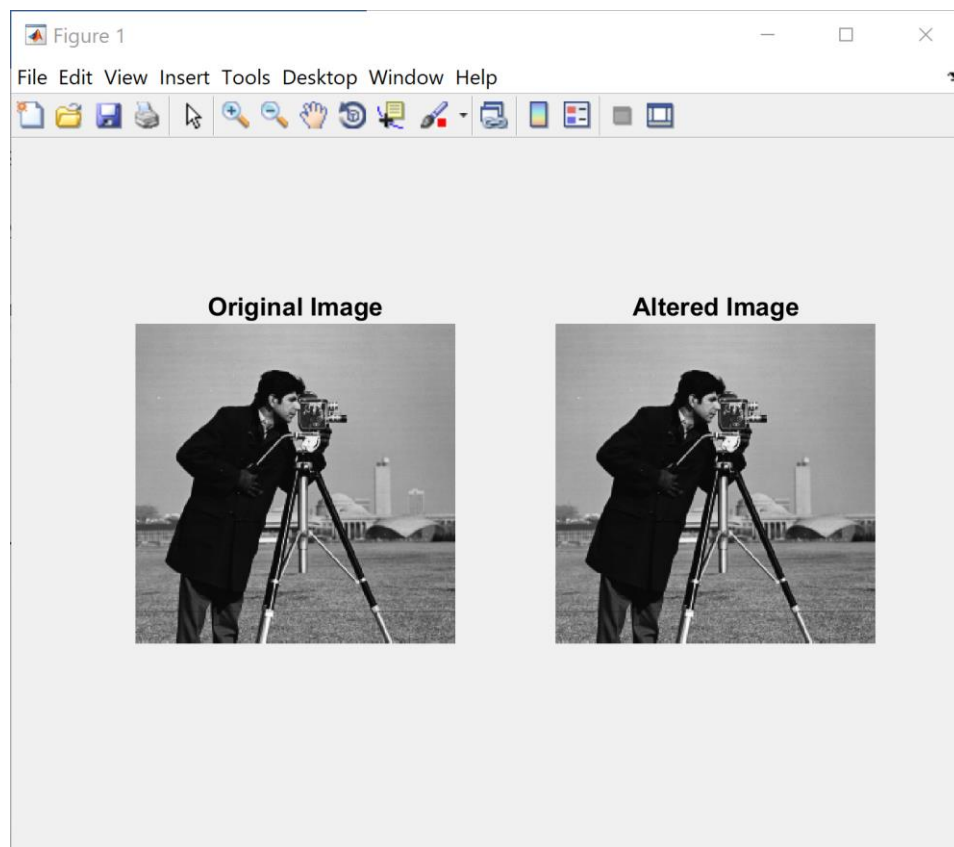
	1	2	3
1	2614		
2			

Diperoleh banyak pixel untuk gambar awal yang bernilai 255 yaitu sebanyak 301 pixel sedangkan untuk gambar yang kedua sebanyak 2614 pixel.

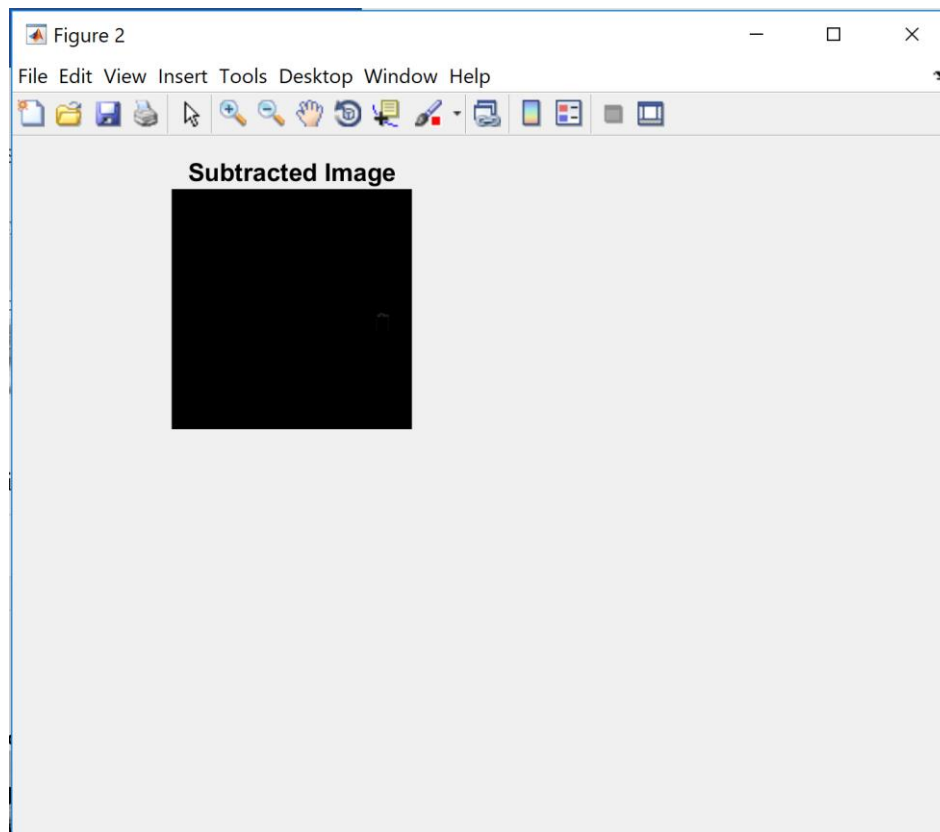
Pengabungan 2 gambar menjadi 1 frame



Menampilkan 2 gambar untuk dibandingkan



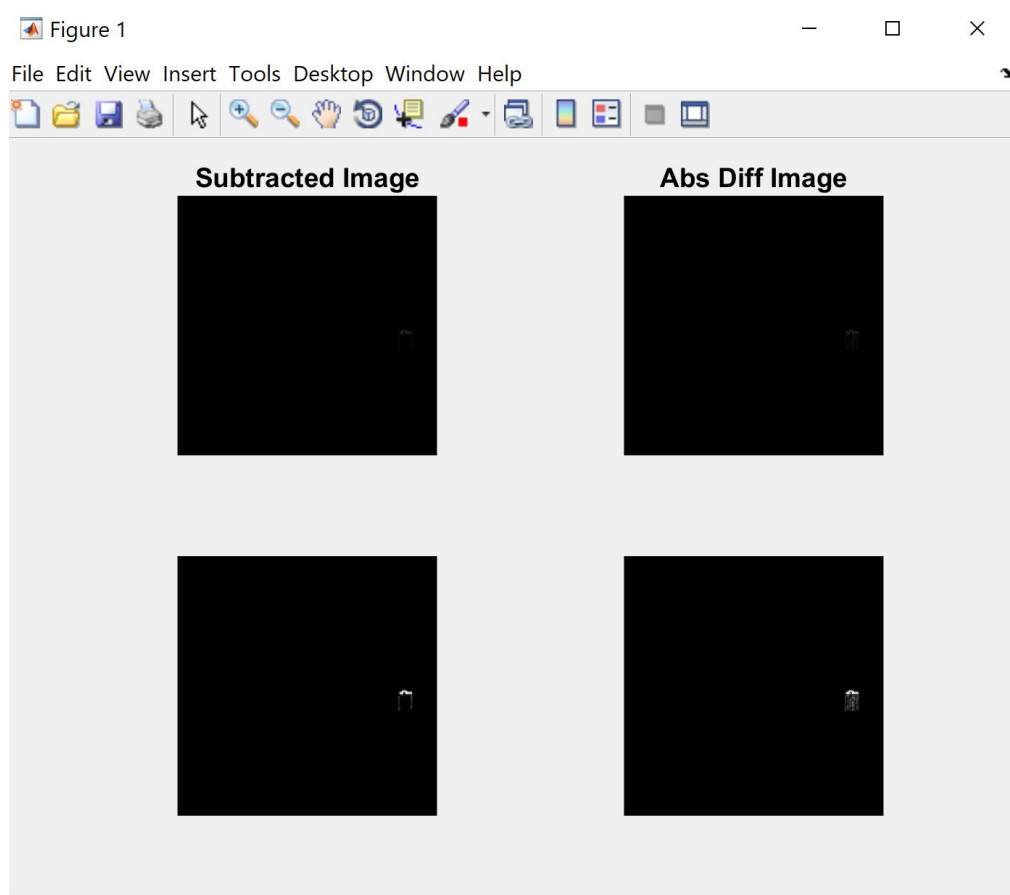
Dengan mengurangi kedua gambar tersebut dengan perintah `imsubtract` , maka akan didapat gambar berikut :




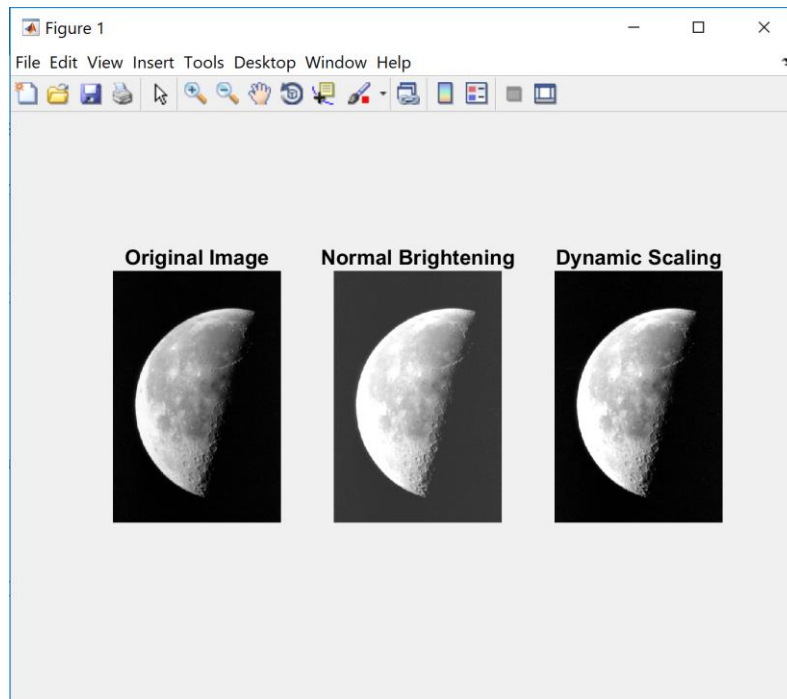
Gambar berikut apabila dilihat maka dapat disimpulkan bahwa gambar memiliki warna hitam semua atau nilainya 0 tetapi kalau ditinjau pada variabel diffim maka dapat dilihat bahwa ada pixel yang tidak bernilai nol.

[illegible]

Membandingkan 4 gambar yang sudah didapat sebelumnya, dapat dilihat bahwa ada perbedaan di beberapa segmen pada gambar.



- Question 3
Dengan menggunakan fungsi zoom pada figure window untuk dapat membesarkan atau mengecilkan gambar yang ada di figure windows. Fungsi tersebut bergambar seperti berikut :

- Question 4
Dapat dilihat lebih jelas bahwa ada perbedaan warna / gambar berbeda antara subtract images dengan abs differ images.
- Question 5
Karena pada gambar terakhir sudah dilakukan proses abstrak diferensiasi gambar. Sehingga gambar semakin jelas.

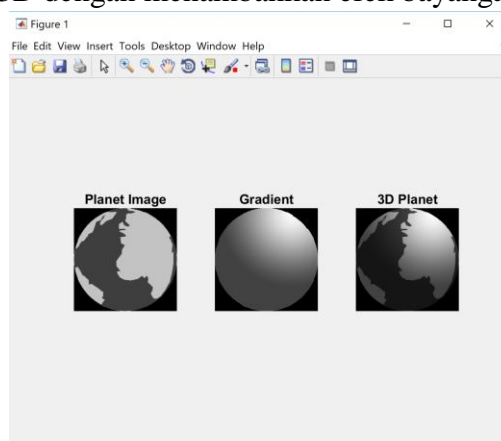


- **Question 6**
 Karena pada gambar normal brightening, nilai setiap sel akan ditambahkan 50 tidak terkecuali yang pixel yang menginisiasi gelap. Sedangkan pada gambar dynamic scaling, nilai setiap sel akan dikalikan dengan 1,2.

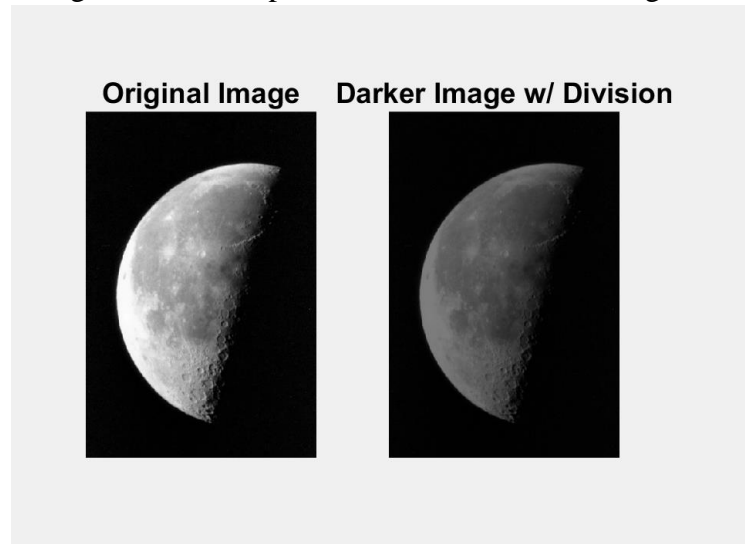
	Original image	Normal brightening	Dynamic scaling
Nilai pixel	3	53	3,6
	10	60	12
	16	66	19,2

Dari tabel diatas dapat , dilihat bahwa perbedaan yang terjadi antara normal dan dynamic scaling cukup besar. Oleh karena itu gambar hitam yang disekitar bulan tidak berubah secara signifikan karena nilainya yang kecil. Karena semakin kecil nilai apabila dikalikan dengan suatu konstanta , maka akan semakin kecil juga perubahannya. Berlaku juga sebaliknya.

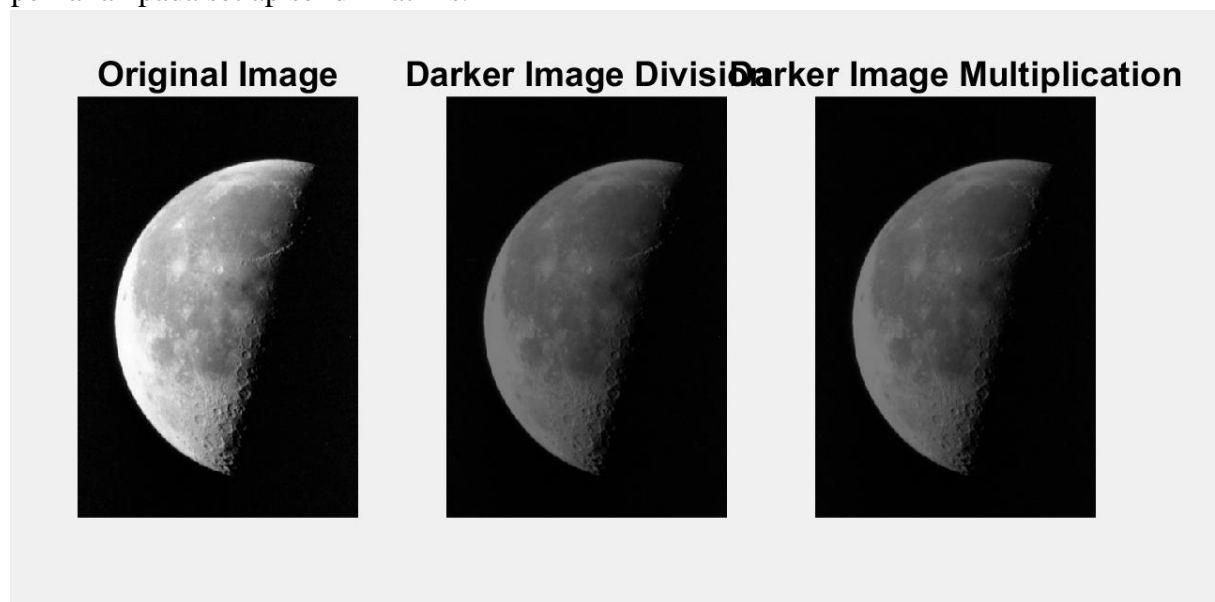
Membuat gambar bumi 3D dengan menambahkan efek bayangannya



Dengan membagikan nilai cell pada matriks, akan membuat gambar lebih gelap.



Membuat gambar yang sama gelapnya dengan menggunakan operasi pembagian dan perkalian pada setiap sel di matriks.

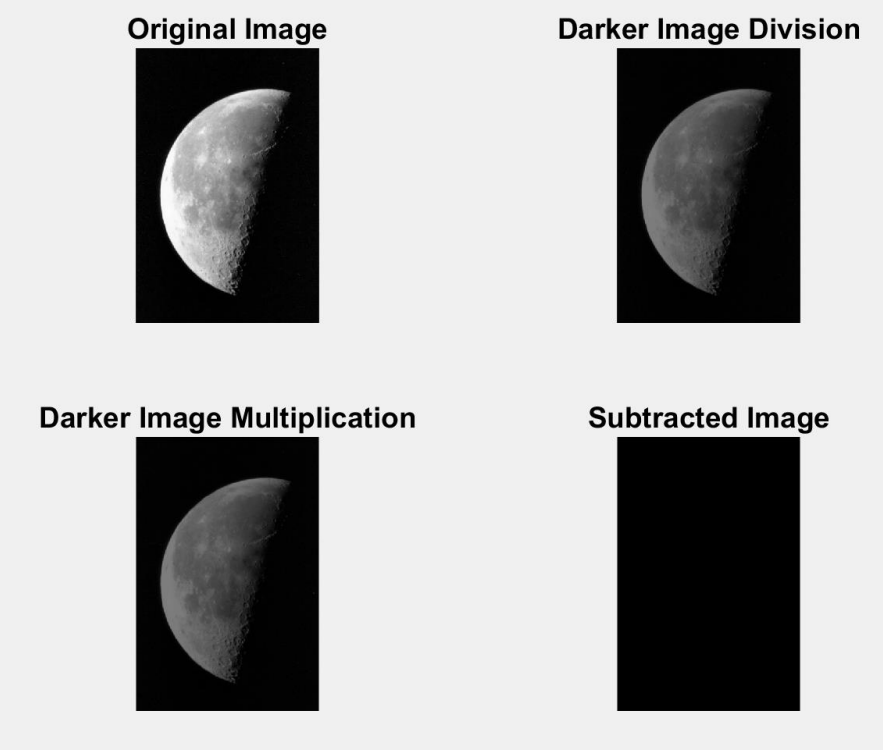


- Question 7
Karena pada program sel dikalikan dengan 0,5 dan dibagi dengan 2 . Dapat dilihat bahwa operasi kedua *statement* tersebut sama (equivalent) satu sama lainnya. Yang mengakibatkan hasil dari gambar akan sama.
- Question 8
Dengan mengurangi variabel I2 dan I3, apabila gambar sama/equivalent maka nilai dari setiap sel pada matriks akan bernilai nol.

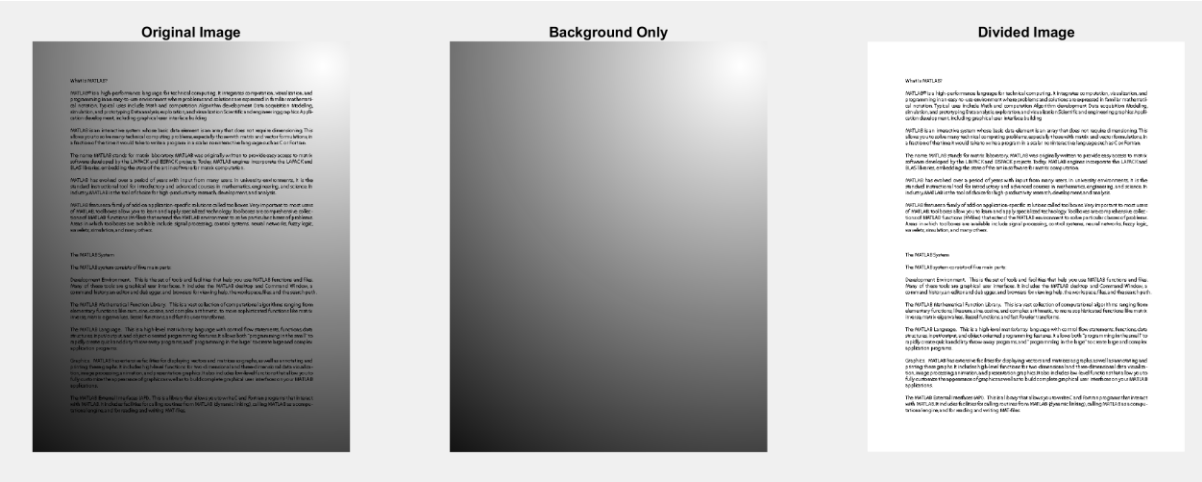
```
I = imread('moon.tif');  
I2 = imdivide(I,2);  
I3 = immultiply(I,0.5);
```

```
diffim = imsubtract(I2,I3);
figure
subplot(2,2,1), imshow(I), title('Original Image');
subplot(2,2,2), imshow(I2), title('Darker Image Division');
subplot(2,2,3), imshow(I3), title('Darker Image Multiplication');
subplot(2,2,4), imshow(diffim), title('Subtracted Image');
```

Dengan mensimulasikan program diatas, maka akan didapat gambar subtracted image yang hitam yang menandakan gambar pada variabel I2 dan I3 adalah sama.



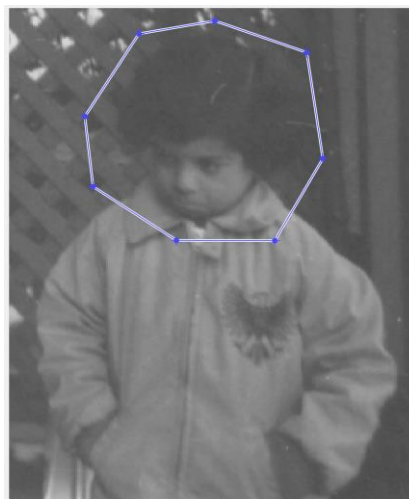
Dengan membagikan original image dengan background only, maka didapat gambar divided image.



- Question 9
Teknik ini tidak akan berhasil jika tidak mendapatkan gambar backgroundnya terlebih dahulu karena apabila gambar originalnya dibagikan dengan gambar lain, besar kemungkinan gambar originalnya akan berubah dan hasilnya bukan seperti divided images lagi.

TUTORIAL 6.2: LOGIC OPERATIONS AND REGION OF INTEREST PROCESSING

- Question 1
Untuk menambah nilai pada polygon, klik kiri mouse pada gambar sehingga membentuk sebuah polygon.
- Question 2
Untuk menghapus nilai dari polygon tekan Backspace, Escape atau Delete, atau klik kanan di dalam wilayah dan pilih Cancel dari menu konteks.
- Question 3
Setelah selesai membentuk sebuah polygon pada gambar, untuk mengakhirinya *double click* pada *mouse*, kemudian tekan *close* pada lembar kerja gambar.
- Question 4
Variabel bw bertipe data logical.
- Question 5
Variabel bw mempresentasikan daerah yang sudah ditandai(dimask) yang dapat dilihat pada gambar berikut:



Daerah yang didalam segi-sembilan merupakan daerah yang dimask.

- Question 6

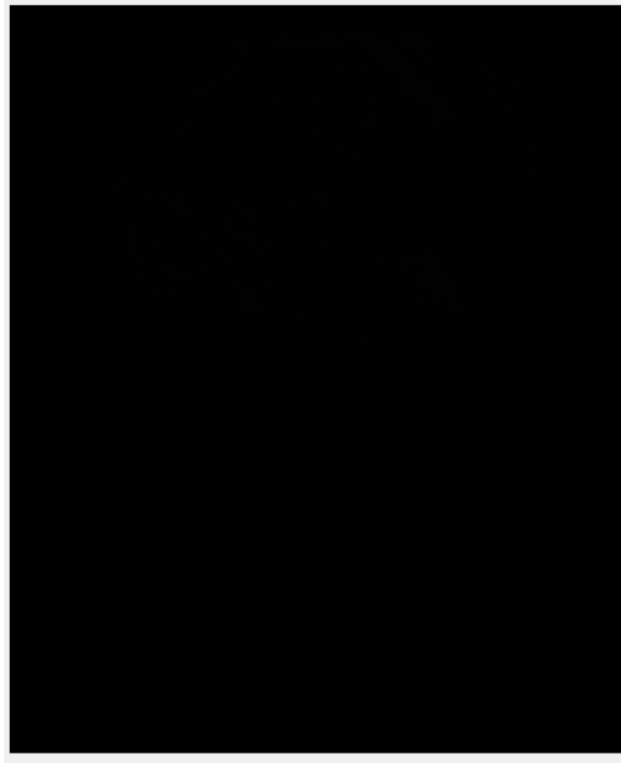
Apabila kita mengubah variabel logical ke uint8 maka variabel bw2 akan berubah menjadi tipe data uint8 tetapi nilai pada variabel tersebut tidak akan berubah. Nilai variabel maksimum dari matriks adalah 1 dan nilai variabel minimum dari matriks adalah 0.

```
max(bw2(:))  
min(bw2(:))
```

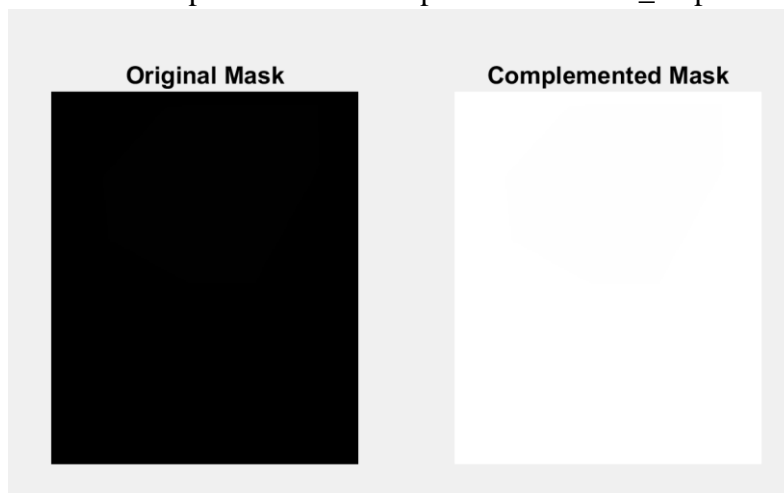
Command Window

```
ans =  
1  
  
ans =  
0
```

Gambar akan seperti yang dibawah jika gambar I dan variabel bw2 dioperasikan menggunakan operasi bitand.



- Question 7
Gambar akan menjadi gelap karena variabel I2 hanya memiliki rentang antara 0 – 1.
- Question 8
Gambar akan putih karena nilai pada variabel bw_cmp semuanya bernilai 255.



[illegible]

- Question 9

Jika kita menggunakan bw2 untuk mengoperasikan bitor maka keluaran yng dihasilkan sama dengan gambar original sedangkan jika menggunakan bw_cmp untuk mengoperasikan bitor maka keluaran yang dihasilkan akan berrwarna putih.

- Question 10

Dengan mengecek menggunakan fungsi XOR, apabila gambar XOR hitam secara keseluruhan maka kedua gambar sama.

```
I_xor = bitxor(bw_cmp2,bw_cmp);
figure
subplot(1,3,1), imshow(bw_cmp), title('Image 1');
subplot(1,3,2), imshow(bw_cmp2), title('Image 2');
subplot(1,3,3), imshow(I_xor,[]), title('XOR Image');
```



Gambar dengan beberapa bagian digelapkan dengan fungsi roipoly untuk menandai bagian mana yang akan digelapkan.



- Question 11

Dengan memodifikasi kode menjadi seperti dibawah ini:

```
bw_cmp = bitcmp(bw); %mask complement
roi = bitor(I_adj,bw); %roi image
not_roi = bitor(I,bw_cmp); %non_roi image
new_img = bitand(roi,not_roi); %generate new image
imshow(new_img) %display new image
```

