Import Modules

In [1]:
```
import cv2

import numpy as np
import matplotlib.pyplot as plt

from skimage import exposure
```

Initialize and load original image

In [2]:
```
# raw_image = cv2.imread('../images/lenna.png')
# raw_image = cv2.imread('../images/cameraman.png')
# raw_image = cv2.imread('../images/edin_castle.png')
# raw_image = cv2.imread('../images/bowl_fruit.png')
# raw_image = cv2.imread('../images/peppers.png')
# raw_image = cv2.imread('../images/sap_of_spain.png')
raw_image = cv2.imread('../images/lung.png')

img_rgb = cv2.cvtColor(raw_image, cv2.COLOR_BGR2RGB)
```

Display original image and description

In [3]:
```
plt.imshow(img_rgb)
plt.title('Original Image')
plt.show()

height, width, color_channel = img_rgb.shape

print(f'Height        : {height} pixels')
print(f'Width         : {width} pixels')
print(f'Color channel : {color_channel} (RGB)')
```


```
Height        : 793 pixels
Width         : 722 pixels
Color channel : 3 (RGB)
```

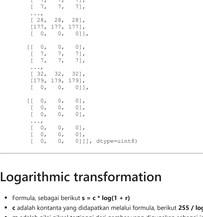
Display original image histogram

In [4]:
```
color = ('r', 'g', 'b')

for i, col in enumerate(color):
    histr = cv2.calcHist([img_rgb], [i], None, [256], [0, 256])

    plt.plot(histr, color = col)
    plt.xlim([0, 256])

plt.title('Original Image Histogram')
plt.xlabel('Intensity Value')
plt.ylabel('Number of Intensity Value')
plt.show()
```



Show list value each pixel

In [5]:
```
img_rgb
```

Out[5]:
```
array([[[  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0],
        ...,
        [  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [  7,   7,   7],
        [  5,   7,   7],
        ...,
        [ 32,  32,  32],
        [175, 179, 179],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [  7,   7,   7],
        [  7,   7,   7],
        ...,
        [ 28,  28,  28],
        [177, 177, 177],
        [  0,   0,   0]],

       ...,

       [[  0,   0,   0],
        [  7,   7,   7],
        [  7,   7,   7],
        ...,
        [ 28,  28,  28],
        [177, 177, 177],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [  7,   7,   7],
        [  7,   7,   7],
        ...,
        [ 32,  32,  32],
        [179, 179, 179],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0],
        ...,
        [  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0]]], dtype=uint8)
```

Logarithmic transformation

- Formula, sebagai berikut **s = c * log(1 + r)**
- **c** adalah konstanta yang didapatkan melalui formula, berikut **255 / log(1 + m)**
- **m** adalah nilai piksel tertinggi dari gambar yang digunakan sebagai input.
- Nilai piksel yang berada pada rentang terang yang jumlah kecil akan ditingkatkan nilainya, sehingga menjadi lebih terlihat.

- Calculate constant

In [6]:
```
c = 255 / (np.log(1 + np.max(img_rgb)))

print(f'Constant value           : {c}')
print(f'Max value pixel from image : {np.max(img_rgb)}')
```

```
Constant value           : 47.358201038974431
Max value pixel from image : 217
```

- Calculate logarithmic transformation

In [7]:
```
log_transformed = c * np.log(1 + img_rgb)
```

- Specify the data type

In [8]:
```
log_transformed = np.array(log_transformed, dtype=np.uint8)
```

- Show value of log_transformed variable

In [9]:
```
log_transformed
```

Out[9]:
```
array([[[  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0],
        ...,
        [  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [ 98,  98,  98],
        [ 98,  98,  98],
        ...,
        [165, 165, 165],
        [245, 245, 245],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [ 98,  98,  98],
        [ 98,  98,  98],
        ...,
        [159, 159, 159],
        [245, 245, 245],
        [  0,   0,   0]],

       ...,

       [[  0,   0,   0],
        [ 98,  98,  98],
        [ 98,  98,  98],
        ...,
        [159, 159, 159],
        [245, 245, 245],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [ 98,  98,  98],
        [ 98,  98,  98],
        ...,
        [165, 165, 165],
        [245, 245, 245],
        [  0,   0,   0]],

       [[  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0],
        ...,
        [  0,   0,   0],
        [  0,   0,   0],
        [  0,   0,   0]]], dtype=uint8)
```

- Display image with logarithmic transformation

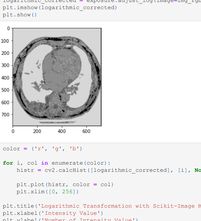
In [10]:
```
plt.imshow(log_transformed)
plt.title('Logarithmic Transformation from Scratch')
plt.show()
```

In [11]:
```
color = ('r', 'g', 'b')

for i, col in enumerate(color):
    histr = cv2.calcHist([log_transformed], [i], None, [256], [0, 256])

    plt.plot(histr, color = col)
    plt.xlim([0, 256])

plt.title('Logarithmic Transformation from Scratch Histogram')
plt.xlabel('Intensity Value')
plt.ylabel('Number of Intensity Value')
plt.show()
```

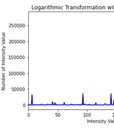


With image processing module

- Logarithmic transformation image with scikit-image module

Menggunakan nilai konstanta yang sama dengan pendekatan from scratch. Terjadi perbedaan hasil.

In [12]:
```
logarithmic_corrected = exposure.adjust_log(image=img_rgb, gain=c)
plt.imshow(logarithmic_corrected)
plt.show()
```



In [13]:
```
color = ('r', 'g', 'b')

for i, col in enumerate(color):
    histr = cv2.calcHist([logarithmic_corrected], [i], None, [256], [0, 256])

    plt.plot(histr, color = col)
    plt.xlim([0, 256])

plt.title('Logarithmic Transformation with Scikit-Image Module')
plt.xlabel('Intensity Value')
plt.ylabel('Number of Intensity Value')
plt.show()
```