

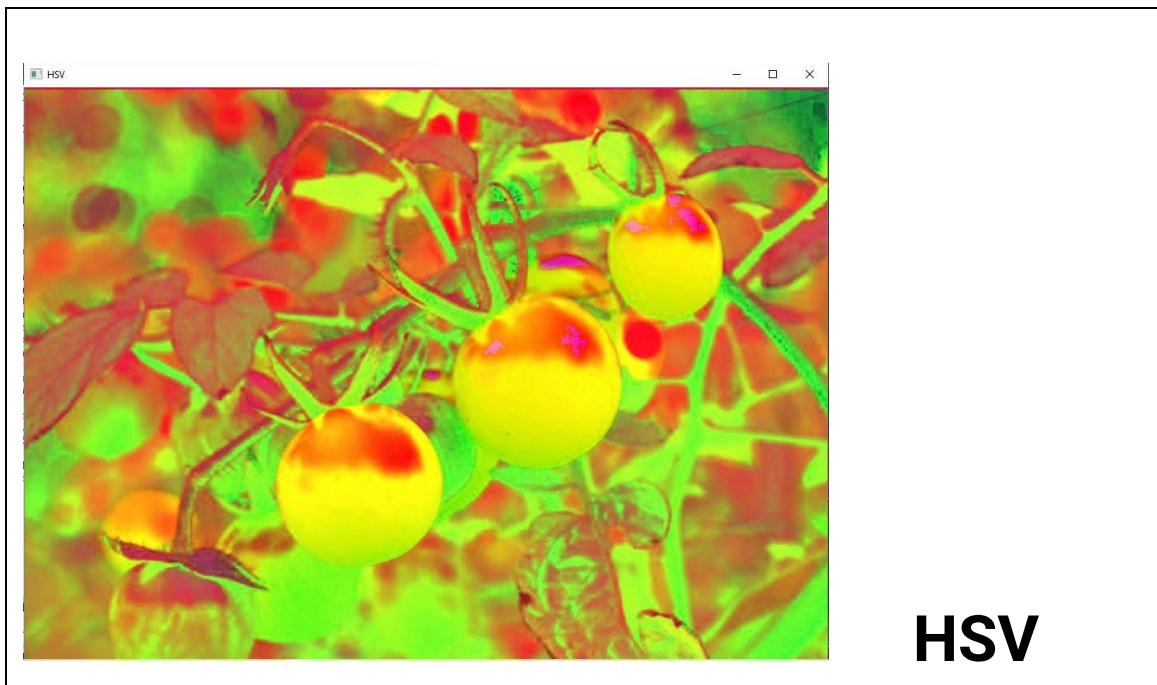
Nama : Fadia Ramadhana
NIM : G64170026

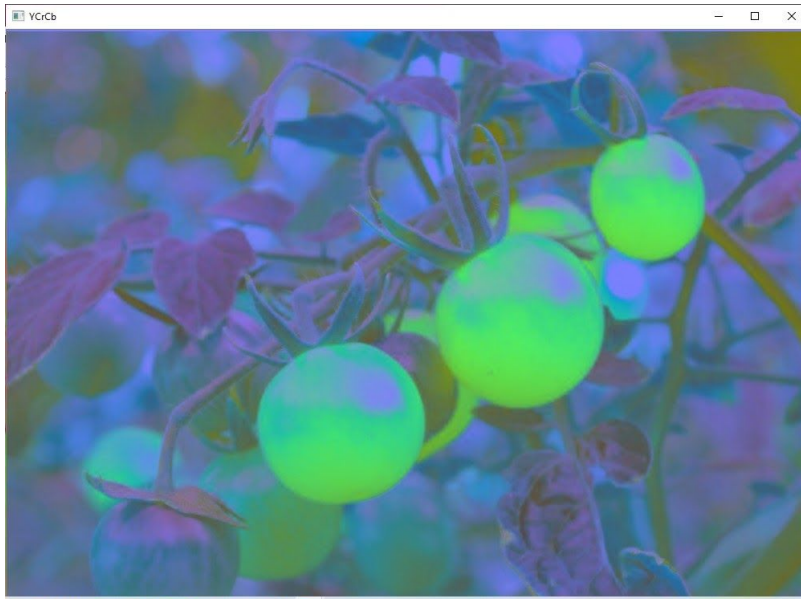
Problem #1: Color space dan Histogram

Gambar asli :

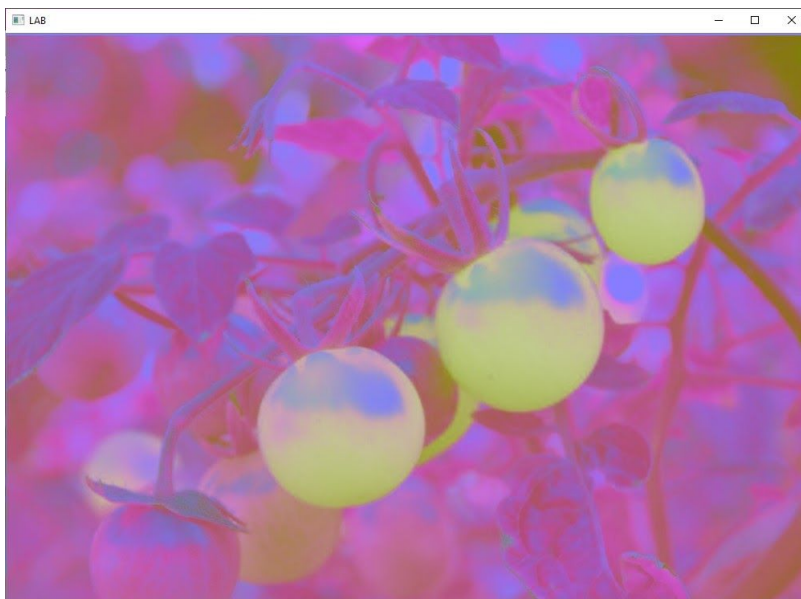


1. Ubah color space image tersebut menjadi berbagai jenis color space yang Anda ketahui, dan pilih color space dan color channel yang menurut Anda sesuai untuk dapat mendeteksi adanya buah tomat yang matang pada gambar tersebut.





YCrCb



L*a*b*



HLS

Untuk mengubah color space image tersebut menjadi beberapa jenis color space dapat dilakukan dengan menggunakan fungsi `cv2.cvtColor()` yang nantinya akan mengkonversi nilai-nilai pada channel R, G, dan B ke channel lainnya pada masing-masing color space yang dikehendaki.

Setelah mengubah color space, saya memutuskan untuk memilih *color space* **L*a*b*** dengan *color channel* "a*" (green to red). Melalui fungsi `cv2.cvtColor()` selanjutnya akan mengkonversi nilai R, G, dan B untuk membentuk channel L*, a*, dan b* dengan ketentuan seperti berikut :

RGB \leftrightarrow CIE L*a*b*

In case of 8-bit and 16-bit images, R, G, and B are converted to the floating-point format and scaled to fit the 0 to 1 range.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$X \leftarrow X/X_n, \text{ where } X_n = 0.950456$$

$$Z \leftarrow Z/Z_n, \text{ where } Z_n = 1.088754$$

$$L \leftarrow \begin{cases} 116 * Y^{1/3} - 16 & \text{for } Y > 0.008856 \\ 903.3 * Y & \text{for } Y \leq 0.008856 \end{cases}$$

$$a \leftarrow 500(f(X) - f(Y)) + \text{delta}$$

$$b \leftarrow 200(f(Y) - f(Z)) + \text{delta}$$

where

$$f(t) = \begin{cases} t^{1/3} & \text{for } t > 0.008856 \\ 7.787t + 16/116 & \text{for } t \leq 0.008856 \end{cases}$$

and

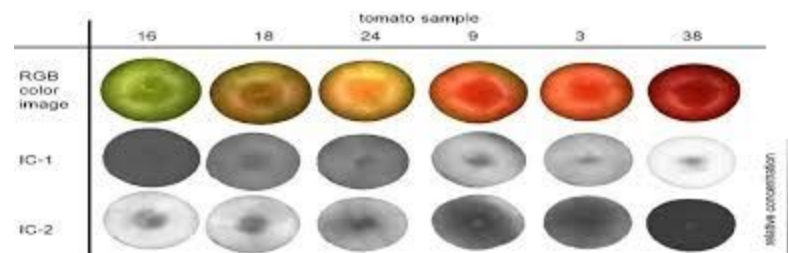
$$\text{delta} = \begin{cases} 128 & \text{for 8-bit images} \\ 0 & \text{for floating-point images} \end{cases}$$

This outputs $0 \leq L \leq 100$, $-127 \leq a \leq 127$, $-127 \leq b \leq 127$. The values are then converted to the destination data type:

- 8-bit images: $L \leftarrow L * 255/100$, $a \leftarrow a + 128$, $b \leftarrow b + 128$
- 16-bit images: (currently not supported)
- 32-bit images: L, a, and b are left as is

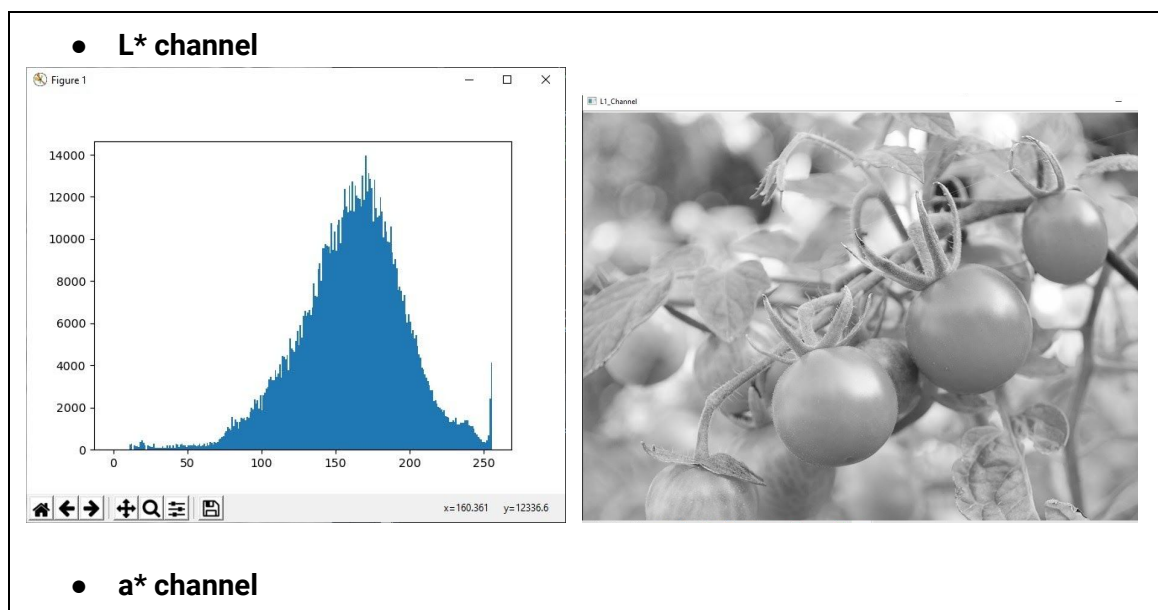
2. Berikan penjelasan mengenai pemilihan color space tersebut. Tuliskan dampak terhadap perubahan color space yang dilakukan.

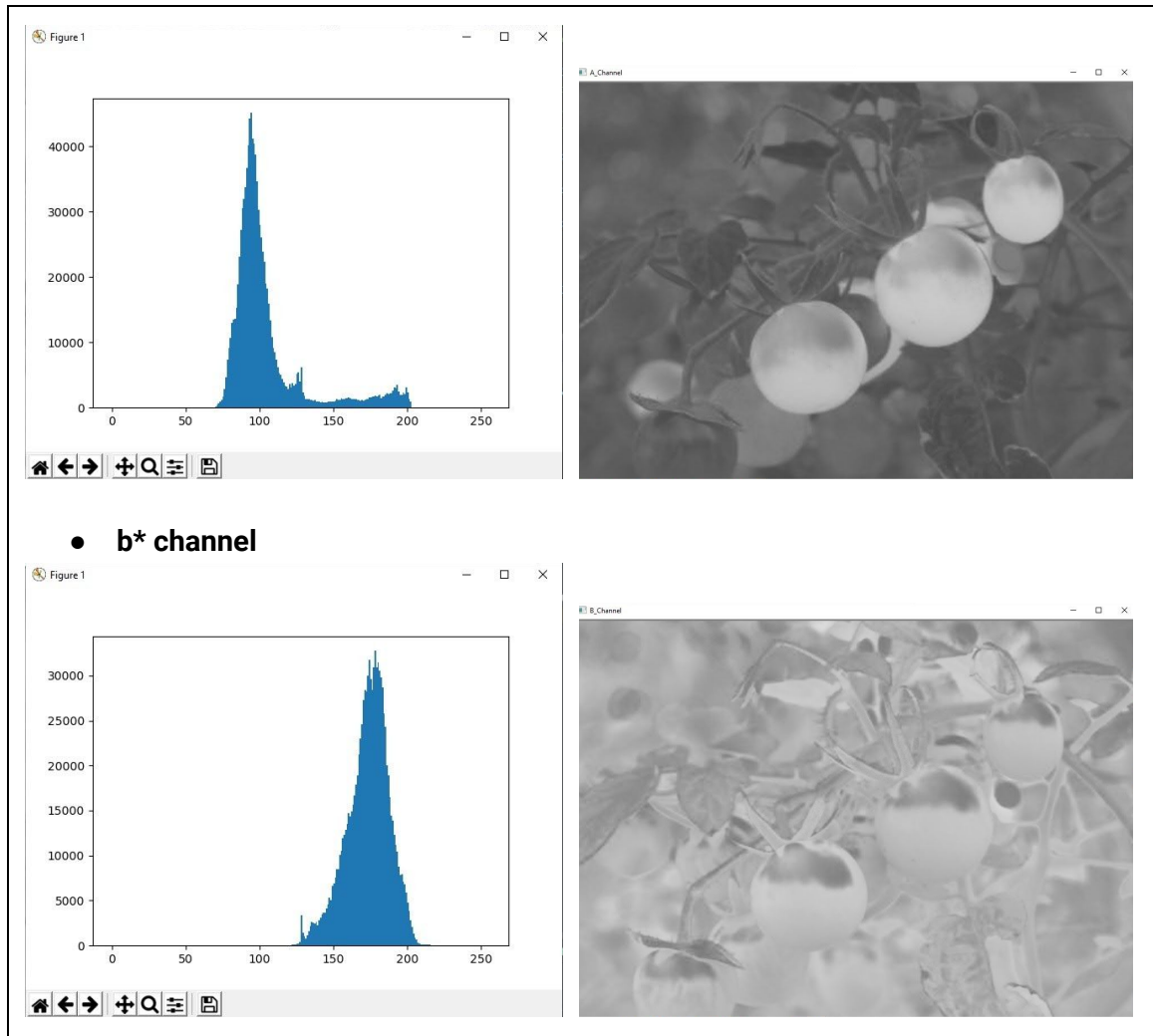
Dapat dilihat setelah mengubah color space gambar tersebut dari RGB ke beberapa color space (HSV, YCrCb, L*a*b*, dan HLS), L*a*b* merupakan color space yang dapat dengan jelas memberikan highlight tomat-tomat yang sudah matang dan persebaran warnanya homogen sehingga dapat dilihat dengan jelas perbedaan antara tomat yang sudah matang, tomat yang belum matang, dahan, dan dedaunan. Selain itu, L*a*b* memiliki sebuah color channel yaitu a* yang merupakan color channel dengan rentang nilai *green to red* dimana menurut saya tingkat warna kematangan tomat sendiri adalah dari warna hijau (belum matang) sampai merah (matang).



Color channel yang lain seperti L* (luminance) juga baik digunakan untuk bagian pencahayaan pada gambar dimana dapat dilihat untuk tomat sendiri, karena tekstur kulitnya yang licin dan mengkilap sehingga ketika terkena cahaya warna aslinya menjadi berubah akibat pantulan cahaya

3. Buat sebuah histogram terhadap color space yang dipilih dan grayscale. Tunjukkan kemungkinan objek yang ditunjukkan pada gambar tsb terhadap histogram.





4. Lakukan segmentasi buah tomat yang matang pada gambar dengan thresholding. Pada tahap ini, ujikan gambar dengan berbagai nilai threshold. Tuliskan apa dampak dari menurunkan dan menaikkan nilai low threshold dan high threshold.

Untuk melakukan segmentasi buah tomat yang matang, hal yang harus dilakukan adalah mengeset range nilai piksel sebagai threshold untuk masing-masing channel yaitu L^* , a^* , dan b^* . Lower (min) range merupakan sebuah array yang menyimpan batas bawah dari masing-masing channel. Upper (max) range merupakan sebuah array yang menyimpan batas atas dari masing-masing channel. Pada OpenCV, range value untuk $L^*a^*b^*$ untuk 8 bit-image adalah :

- $0 > L^* > 255$
- $0 > a^* > 255$
- $0 > b^* > 255$

Bersamaan dengan aturan tersebut dengan menggunakan beberapa website rujukan dan beberapa tes empiris serta evaluasi visual, saya akan melakukan percobaan

dengan menggunakan beberapa range threshold untuk L^* , a^* , dan b^* secara berurutan yaitu:

1. (30 - 200, 150 - 255, 54 - 200)
2. (20 - 170, 120 - 255, 160 - 255)
3. (10 - 255, 132 - 255, 100 - 255)

Range ini nantinya akan menjadi kriteria untuk nilai-nilai piksel yang termasuk ke dalam tomat yang sudah matang.

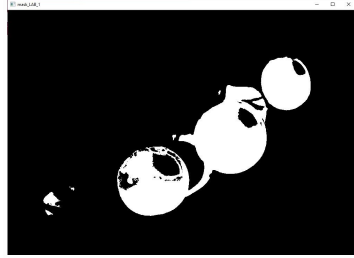
```
10  # menerapkan threshold ke gambar LAB untuk menentukan range atas dan bawah
11  # batas bawah lab1
12  min_LAB1 = np.array([30,150,54],np.uint8)
13  # batas atas lab1
14  max_LAB1 = np.array([200,255,200],np.uint8)
15  # batas bawah lab2
16  min_LAB2 = np.array([20,120,160],np.uint8)
17  # batas atas lab2
18  max_LAB2 = np.array([170,255,255],np.uint8)
19  # batas bawah lab3
20  min_LAB3 = np.array([10,132,100],np.uint8)
21  # batas atas lab3
22  max_LAB3 = np.array([255,255,255],np.uint8)
```

Gambar yang telah diubah color space nya menjadi $L^*a^*b^*$, akan digunakan untuk membuat mask bersama dengan nilai range (min dan max) dari masing-masing channel yang telah diset sebelumnya. Mask untuk image didapat menggunakan fungsi dari openCV yaitu fungsi `cv2.inRange()` dimana fungsi tersebut hanya mengembalikan MASK bernilai biner. Piksel putih (255) mewakili piksel yang termasuk ke dalam range antara min dan max dan begitu pula piksel hitam (0) mewakili piksel yang tidak termasuk ke dalam range.

```
24  # membuat mask dengan
25  # mengembalikan nilai biner image menggunakan range bawah dan atas (min, max)
26  ripeMaskLAB1 = cv2.inRange(imageLAB,min_LAB1,max_LAB1)
27  ripeMaskLAB2 = cv2.inRange(imageLAB,min_LAB2,max_LAB2)
28  ripeMaskLAB3 = cv2.inRange(imageLAB,min_LAB3,max_LAB3)
```

Untuk melihat hasil mask, dapat dengan menampilkannya menggunakan fungsi `cv2.imshow()`. Untuk masing-masing range threshold, hasil outputnya adalah sebagai berikut :

1. (30 - 200, 150 - 255, 54 - 200)



2. (20 - 170, 120 - 255, 160 - 255)



3. (10 - 255, 132 - 255, 100 - 255)



Dampak dari menaikkan atau menurunkan nilai low threshold dan high threshold adalah tentu saja pada seberapa banyak pixel yang dapat direpresentasikan melalui threshold tersebut. Nilai threshold pada range pertama belum dapat dengan sempurna merepresentasikan buah tomat yang matang karena bagian tomat yang terkena cahaya tidak sempurna masuk. Kemudian pada threshold range kedua saya mencoba untuk menurunkan nilai pada low threshold channel L* dan a* kemudian menaikkan low dan high threshold pada channel b*, hasilnya semakin buruk yang mungkin disebabkan oleh value pada low threshold channel b* yang terlalu tinggi, ditambah lagi dengan value pada low threshold channel a* yang terlalu rendah yaitu 120 mengakibatkan tomat yang belum matang ikut masuk. Selanjutnya, dengan nilai threshold pada range ketiga akhirnya didapatkan mask yang hampir sempurna merepresentasikan tomat-tomat yang sudah matang. Untuk value pada high threshold channel a* saya sengaja untuk tetap memberikan value 255 (totally red).

5. Berikan gambar hasil masking pada buah tomat tersebut (menghilangkan background/mengubah background menjadi hitam).

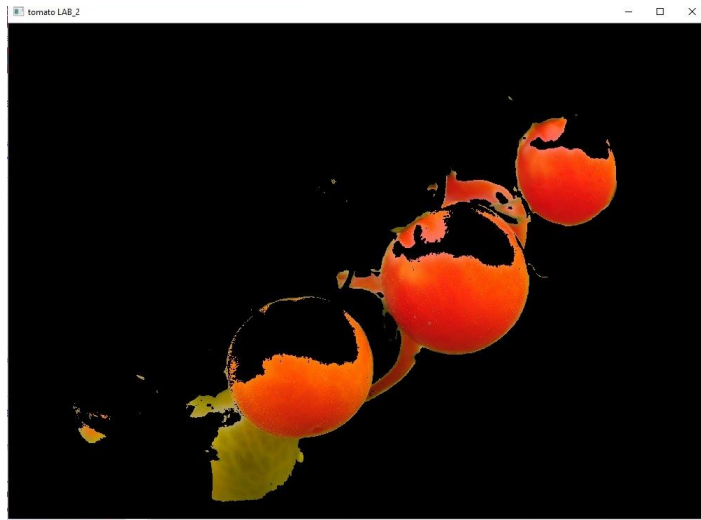
Setelah mendapatkan mask, maka langkah selanjutnya adalah melakukan segmentasi wajah pada gambar asli dengan menggunakan mask yang telah didapatkan pada step sebelumnya dengan memanfaatkan fungsi `cv2.bitwise_and()` yang menjaga setiap piksel dalam gambar yang diberikan jika nilai yang sesuai dalam mask adalah 1. Cara

bekerja fungsi ini mirip seperti subtraction, dimana gambar output merupakan hasil gambar hasil irisan dari dua buah gambar. Untuk menampilkan hasil akhir dapat dilakukan dengan menggunakan fungsi `cv2.imshow()` dan akan didapatkan output seperti berikut :

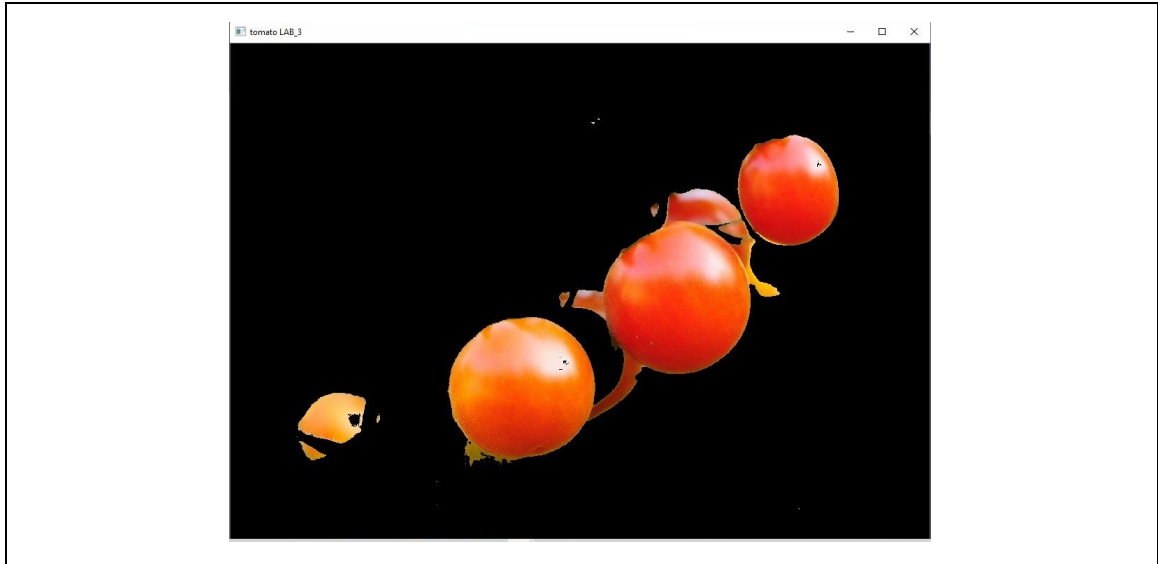
1. Segmentasi menggunakan mask pada range pertama



2. Segmentasi menggunakan mask pada range kedua



3. Segmentasi menggunakan mask pada range ketiga



Sumber referensi :

https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html

https://www.researchgate.net/post/What_would_be_a_good_colour_space_to_deal_with_images_taken_in_varying_lighting_conditions

GAWANDE, A. P.; DHANDE, S. S. Implementation of fruit Grading System by Image Processing and Data Classifier-A Review. *International Journal of Engineering Research and General Science*, 2014, 2.6: 411-413.