**LAPORAN PRAKTIKUM 2**

**PENGOLAHAN CITRA DIGITAL**

**MANIPULASI GAMBAR/CITRA DIGITAL**



Oleh

|  |  |  |
| --- | --- | --- |
| Nama | : | Rizqillah |
| NIM | : | 1957301020 |
| Kelas | : | TI 3C |
| Dosen Pembimbing | : | Musta’inul Abdi, SST., M.Kom. |

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI DAN KOMPUTER**

**TAHUN 2021**

# LEMBAR PENGESAHAN

No. Praktikum : 02/PPCD/3C/TI/2021

Judul : Manipulasi Gambar/Citra Digital

Nama : Rizqillah

NIM / Kelas : 1957301020 / TI 3C

Jurusan : Teknologi Informasi Dan Komputer

Prodi : Teknik Informatika

Tanggal praktikum : 24 September 2021

Tanggal penyerahan : 29 September 2021

Nilai :

|  |  |
| --- | --- |
|  | Buketrata, 29 September 2021 |
|  | Dosen Pembimbing, |
|  |  |
|  | Musta’inul Abdi, SST., M.Kom.  NIP. 19911030 20190310 1 5 |

# DAFTAR ISI

[LEMBAR PENGESAHAN i](#_Toc83578973)

[DAFTAR ISI ii](#_Toc83578974)

[DAFTAR GAMBAR iii](#_Toc83578975)

[BAB I PENDAHULUAN 1](#_Toc83578976)

[1.1 Tujuan 1](#_Toc83578977)

[1.2 Dasar Teori 1](#_Toc83578978)

[1.3 Implementasi 7](#_Toc83578979)

[BAB II PEMBAHASAN 11](#_Toc83578980)

[2.1 Latihan 11](#_Toc83578981)

[BAB III PENUTUP 22](#_Toc83578982)

[3.1 Kesimpulan 22](#_Toc83578983)

[DAFTAR PUSTAKA 23](#_Toc83578984)

# DAFTAR GAMBAR

[Gambar 1 Pixel Warna 1](#_Toc83579035)

[Gambar 2 Kode Warna RGB 3](#_Toc83579036)

[Gambar 3 Gambar dengan warna RGB 3](#_Toc83579037)

[Gambar 4 Rumus *Grayscale* 3](#_Toc83579038)

[Gambar 5 Gambar dengan warna *Grayscale* 4](#_Toc83579039)

[Gambar 6 Rumus nilai xbw 5](#_Toc83579040)

[Gambar 7 Gambar dengan warna *Black* and *White* 5](#_Toc83579041)

[Gambar 8 Gambar dengan *Layer* *Red*, *Green* dan *Blue* 5](#_Toc83579042)

[Gambar 9 Perhitungan Citra Asli ke Negatif 6](#_Toc83579043)

[Gambar 10 Contoh Gambar *Invers* atau *Negative* 6](#_Toc83579044)

[Gambar 11 Tampilan GUI 7](#_Toc83579045)

[Gambar 12 *Grayscale Button* 7](#_Toc83579046)

[Gambar 13 Hasil *Grayscale* 8](#_Toc83579047)

[Gambar 14 *Black and White Button* 8](#_Toc83579048)

[Gambar 15 Hasil *Black and White* 8](#_Toc83579049)

[Gambar 16 Layer Red 9](#_Toc83579050)

[Gambar 17 Layer Green 9](#_Toc83579051)

[Gambar 18 Layer Blue 10](#_Toc83579052)

[Gambar 19 Layer Mix 10](#_Toc83579053)

[Gambar 20 Invers/Negative 10](#_Toc83579054)

[Gambar 21 Output soal 1 11](#_Toc83579055)

[Gambar 22 Output Threshold 12](#_Toc83579056)

[Gambar 23 Output Threshold Otsu 13](#_Toc83579057)

[Gambar 24 Output Cyan 14](#_Toc83579058)

[Gambar 25 Output Sephia 14](#_Toc83579059)

[Gambar 26 Output Hot 15](#_Toc83579060)

[Gambar 27 Output Cool 15](#_Toc83579061)

[Gambar 28 Output False 15](#_Toc83579062)

[Gambar 29 Cyan 18](#_Toc83579063)

[Gambar 30 Magenta 18](#_Toc83579064)

[Gambar 31 Yellow 18](#_Toc83579065)

[Gambar 32 Black 19](#_Toc83579066)

[Gambar 33 HSL (Hue Saturation Lightness) 19](#_Toc83579067)

[Gambar 34 HSI (Hue Saturation Intesity) 19](#_Toc83579068)

# BAB I

**PENDAHULUAN**

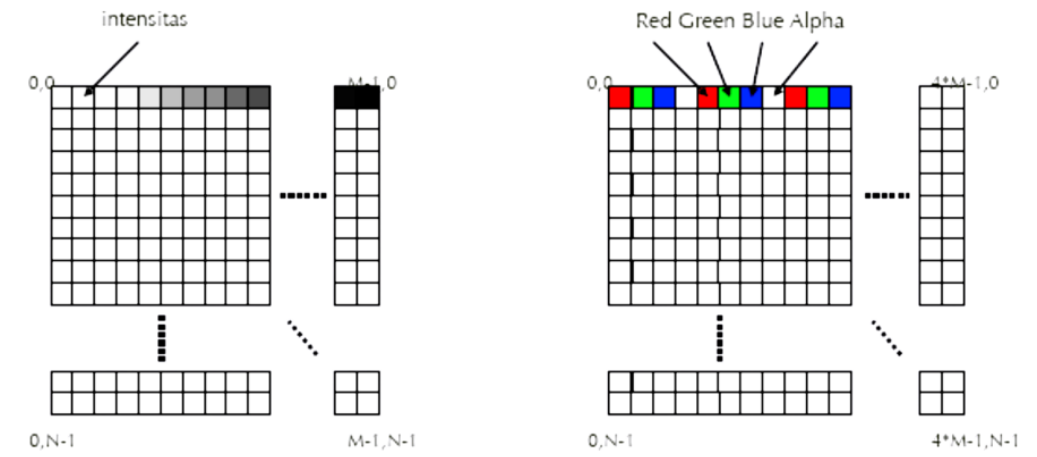
## Tujuan

1. Mahasiswa mampu memahami konsep dasar representasi citra digital
2. Mahasiswa mampu memanipulasi citra digital

## Dasar Teori

**Citra sebagai matrik 2D**

Citra/gambar secara konsep dapat dinyatakan dalam bentuk 2D atau array 2D, dimana setiap *pixel*-nya pada posisi (x,y) dinyatakan dalam P(x,y). *Pixel* adalah representasi dari titik terkecil citra digital atau gambar grafis dan dihitung per Inci(satuan matrik), *Pixel* merupakan singkatan akronim dari bahasa inggris *Picture Element.* Setiap *pixel* pada data gambar mempunyai nilai sesuai dengan format gambar yang digunakan:



Gambar 1 Pixel Warna

**Pixel Density**

*Pixel Density* adalah istilah untuk menunjukkan ketajaman serta kecerahan pada layar sebuah perangkat elektronik. *Pixel Density* atau Kepadatan Piksel biasanya dinyatakan dalam satuan *pixel per inch* (PPI) adalah indikasi penting yang memberitahu tentang seberapa jelas layar dari sebuah perangkat elekronik (Akbar, 2018). Contohnya seperti *smartphone*, komputer/laptop hingga televisi. Dengan kata lain *Pixel Density* adalah kerapatan jumlah *pixel* dalam satu inch.

**PPI (Pixel per Inch)**

PPI atau *Pixel per Inch* adalah tolak ukur ketajaman resolusi layar melalui banyaknya jumlah *pixel* yang menempati area gambar dalam 1 inch, satuan *pixel per inch* digunakan hanya untuk layar perangkat digital. Ketentuan standar resolusi dalam ruang digital adalah 72, namun tingginya angka menunjukkan kualitas gambar. PPI merupakan satuan untuk digital/tidak di kertas fisik melainkan layar monitor (Pranata, t.thn.).

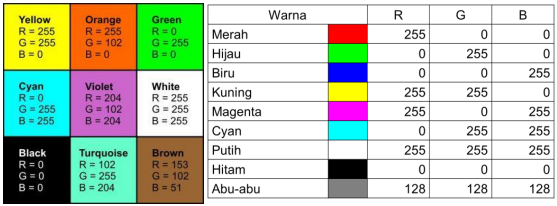
**LPI (Line per Inch)**

LPI merupakan singkatan *Lines Per Inch* yang artinya LPI adalah sebuah satuan yang berdasarkan banyaknya garis baris tiap inci-nya (Ibnu, 2021). LPI adalah tolak ukur ketajaman resolusi mesin cetak *offset* berdasar pada banyaknya jumlah *line*/garis yang menempati area gambar dalam 1 inch, umumnya satuan *line per inch* biasa digunakan pada cetak koran dan lain-lain.

**DPI (Dot per Inch)**

DPI atau *Dot per Inch* adalah tolak ukur ketajaman resolusi ukuran cetak melalui banyaknya jumlah *dot* (titik lingkaran terkecil) yang menempati area gambar dalam 1 inch. Semakin besar satuan DPI maka semakin tajam hasil gambar cetak. DPI juga bisa disebut ukuran jumlah titik sepanjang garis satu inci atau 2,54 cm. DPI bukan untuk mengukur titik per inci persegi, namun hanya dalam satu garis. Misalnya, pada printer tertulis bisa mencetak 600 DPI. Artinya printer tersebut dapat mencetak gambar secara horizontal maupun vertikal. Printer 600 DPI dapat mencetak gambar yang memiliki 360.000 titik per inci persegi (Sains, 2021).

1. RGB(*Red, Green, Blue*) adalah sebuah metode penggambaran warna. Warna yang digunakan untuk citra biasanya terdiri dari tiga warna, yaitu merah, hijau dan biru (Hidayati, 2021). Pada gambar berwarna, setiap titik mempunyai nilai 24bit RGB, dimana masing-masing komponen warna *Red*, *Green* dan *Blue* mempunyai nilai 8bit, atau dengan kata lain setiap komponen warna memiliki nilai 0 s/d 255. Secara default warna R = 255,0,0 G = 0,255,0 B = 0,0,255. Berikut beberapa kombinasi dari ketiga warna tersebut dan gambar yang RGB atau berwarna



Gambar 2 Kode Warna RGB

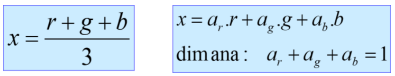


Gambar 3 Gambar dengan warna RGB

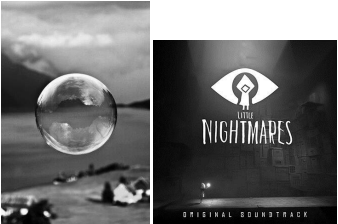
1. *Grayscale* adalah berbagai nuansa warna monokromatik dari hitam menjadi puti. Oleh karena itu, gambar *grayscale* hanya memiliki warna abu-abu dan tidak berwarna (Jejaring, 2019). Adapun menurut website glosarium.org, *grayscale* adala suatu istilah untuk menyebutkan satu citra yang memiliki warna abu-abu, hitam dan putih (Pann, 2019). Pada gambar *grayscale* atau derajat keabuan, setiap titik mempunyai nilai 8bit, atau 0 s/d 255. Citra RGB dapat dikonversi menjadi citra *grayscale* sehingga dihasilkan hanya satu kanal warna. Persamaan yang umumnya digunakan untuk mengkonversi citra RGB *truecolor* 24-bit menjadi citra *grayscale* 8-bit adalah :

*Grayscale = 0.2989\*R + 0.5870\*G + 0.1140\*B*

Setiap *pixel* mempunyai nilai *red*(r), *green*(g) dan *blue*(b) dengan nilai masing-masing 0-255. Setiap *pixel* mempunyai nilai derajat keabuan x dengan nilai 0- 255



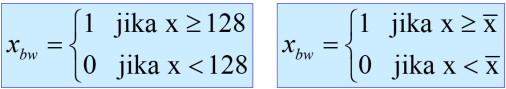
Gambar 4 Rumus *Grayscale*



Gambar 5 Gambar dengan warna *Grayscale*

1. *Threshold*/*Black* and *White*. *Thresholding* citra adala suatu metode yang digunakan untuk memisahkan antara objek dan *background*-nya. *Thresholding* merupakah teknik yang sederhana dan efektif untuk segmentasi citra (Fauzi et al., 2010). Adapun menurut website pemrogramanmatlab.com, *Thresholding* adala salah satu metode segmentasi citra di mana prosesnya didasarkan pada perbedaan derajat keabuan citra (Adi Pamungkas, Thresholding | Pemrograman Matlab, 2016). Pada gambar hitam putih (BW), setiap titik mempunyai nilai 0 atau 1. Jenis citra yang ketiga adalah citra biner. Citra biner adalah citra yang pikselnya memiliki kedalaman bit sebesar 1bit sehingga hanya memiliki dua nilai intensitas warna yaitu 0(hitam) dan 1(putih). Citra *grayscale* dapat dikonversi menjadi citra biner melalui proses *thresholding*. Dalam proses *thresholding*, dibutuhkan suatu nilai *threshold* sebagai nilai pembatas konversi. Nilai intensitas piksel yang lebih besar atau sama dengan nilai *threshold* akan dikonversi menjadi 1. Sedangkan nilai intensitas piksel yang kurang dari nilai *threshold* akan dikonversi menjadi 0. Misalnya nilai *threshold* yang digunakan adalah 128, maka piksel yang mempunyai intensitas kurang dari 128 akan diubah menjadi 0(hitam) dan yang lebih dari atau sama dengan 128 akan diubah menjadi 1(putih)

Setiap *pixel* mempunyai nilai derajat keabuan x dengan nilai 0-255. Setiap pixel mempunyai nilai warna xbw dengan nilai 0 dan 1

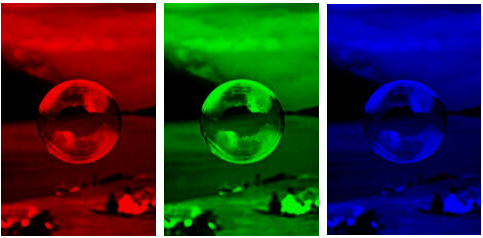


Gambar 6 Rumus nilai xbw



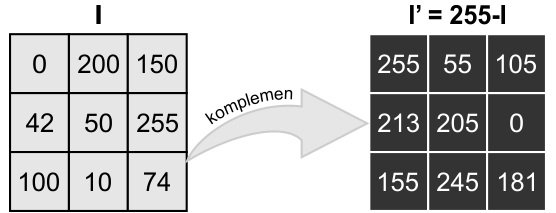
Gambar 7 Gambar dengan warna *Black* and *White*

1. Berdasarkan jenis warnanya, citra digital dapat dibedakan menjadi tiga jenis yaitu citra RGB, Grayscale, dan Biner. Citra RGB(*Red*, *Green*, *Blue*) adalah citra yang nilai intensitas *pixel*-nya tersusun oleh tiga layer, yaitu merah, hijau dan biru (Adi Pamungkas, Pengolahan Citra Digital, 2015). *Layer Red*, *Layer Green*, *Layer Blue* adalah menampilkan gambar dengan dominasi warna Merah, Hijau, atau Biru saja. Pada akhirnya hasil yang ditampilkan hanya salah satu warna yang dominan tersebut. Namun, jika digabungkan, maka terjadi gambar RGB yang utuh seperti sediakala



Gambar 8 Gambar dengan *Layer* *Red*, *Green* dan *Blue*

1. *Invers* adalah manipulasi gambar dengan menampilkan warna gambar berlawanan dengan gambar asli. Citra Negatif didapat dengan membalikkan nilai setiap *pixel* atau sering disebut dengan operasi negasi (operasi *invers*). Proses ini cocok untuk memperbaiki detail level putih atau keabuan pada latar belakang yang gelap. Proses ini banyak digunakan pada citra medis seperti USG dan foto sinar X. Misalkan pada citra *gray* 8-bit (256 derajat keabuan), citra negatif yang diperoleh dengan persamaan : fo(x,y) = 255 – fi(x,y) (Dyantama, 2016). Menurut website pemrogramanmatlab.com, Citra Negatif adalah citra yang nilai *pixel*-nya berkebalikan dengan citra aslinya. Apabila citra asli disimbolkan dengan I, maka negatif dari citra tersebut adalah I’ = 255-I (Adi Pamungkas, Complement Image (Negative Image), 2015). Contoh perhitungan nilai pixel dari citra negatif ditunjukkan pada gambar berikut :



Gambar 9 Perhitungan Citra Asli ke Negatif

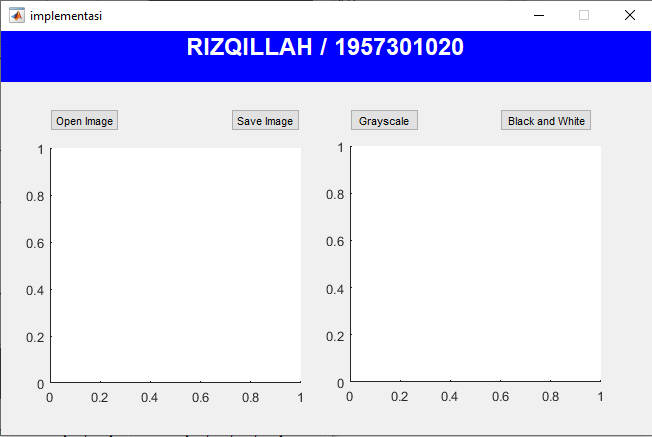
Hasil gambar terlihat seperti *negative* film pada kamera analog.



Gambar 10 Contoh Gambar *Invers* atau *Negative*

## Implementasi

Untuk menampilkan gambar tersebut perlu dibuat dalam bentuk GUI supaya mempermudah pengolahan gambar tersebut. Pertama kita buat GUI seperti gambar berikut



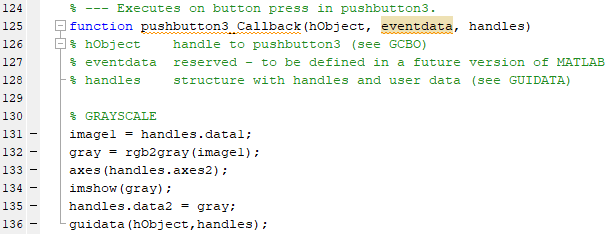
Gambar 11 Tampilan GUI

Setelah dibuat GUI seperti gambar tersebut. Masukkan kode pada MATLAB sebagai berikut:

**Grayscale**

|  |
| --- |
| image1 = handles.data1;  gray = rgb2gray(image1);  axes(handles.axes2);  imshow(gray);  handles.data2 = gray;  guidata(hObject,handles); |

Dan hasilnya,



Gambar 12 *Grayscale Button*

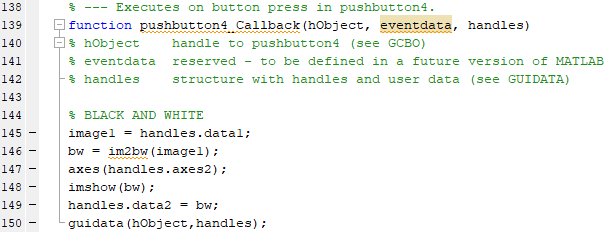


Gambar 13 Hasil *Grayscale*

**Black and White**

|  |
| --- |
| image1 = handles.data1;  bw = im2bw(image1);  axes(handles.axes2);  imshow(bw);  handles.data2 = bw;  guidata(hObject,handles); |

Dan hasilnya,



Gambar 14 *Black and White Button*



Gambar 15 Hasil *Black and White*

**Layer Red, Green, Blue**

|  |
| --- |
| image1 = handles.data1;  allBlack = zeros(size(image1, 1), size(image1, 2));  %red  red = image1(:,:,1);  ired = cat(3, red, allBlack, allBlack);  %green  green = image1(:,:,1);  igreen = cat(3, allBlack, green, allBlack);  %blue  blue = image1(:,:,1);  iblue = cat(3, allBlack, allBlack, blue);  %mix  mix = cat (3,255-red,120-green,20-allBlack);  axes(handles.axes2);  imshow(iblue);  handles.data2 = iblue;  guidata(hObject,handles); |

Dan hasilnya,

RED :

****

Gambar 16 Layer Red

GREEN :

****

Gambar 17 Layer Green

BLUE :

****

Gambar 18 Layer Blue

MIX :

****

Gambar 19 Layer Mix

**Invers**

|  |
| --- |
| image1 = handles.data1;  inverse=imcomplement(image1);  handles.data2 = inverse;  axes(handles.axes2);  imshow(inverse);  handles.data2 = inverse;  guidata(hObject,handles); |

Dan hasilnya,



Gambar 20 Invers/Negative

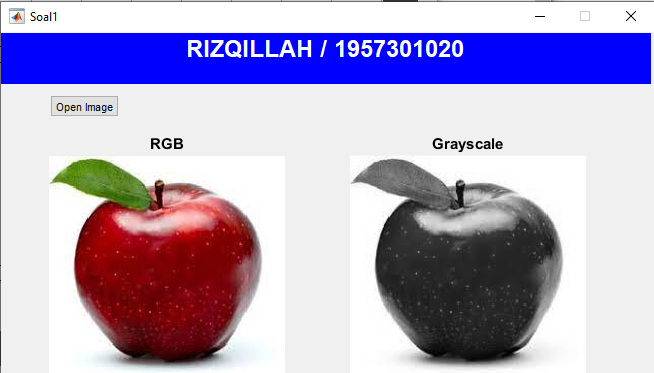
# BAB II

**PEMBAHASAN**

## Latihan

1. **Coba dengan menampilkan 2 gambar yang berbeda sekaligus pada MATLAB dengan menggunakan GUI**

**Hasil :**

****

Gambar 21 Output soal 1

**Program :**

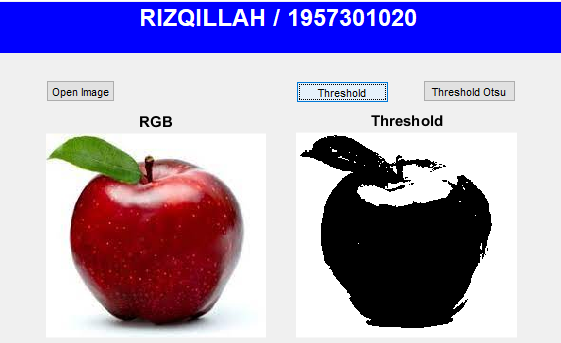
|  |
| --- |
| % --- Executes on button press in pushbutton1.  function pushbutton1\_Callback(hObject, eventdata, handles)  % hObject handle to pushbutton1 (see GCBO)  % handles structure with handles and user data (see GUIDATA)    % OPEN IMAGE  [FileName,PathName]= uigetfile(...  {'\*.bmp;\*.jpg;\*.jpeg;\*.png;\*.tiff;\*.gif', 'File of Type (\*.bmp,\*.jpg,\*.jpeg,\*.png,\*.tiff,\*.gif)';  '\*.bmp', 'File Bitmap(\*.bmp)';...  '\*.jpg', 'File Jpg(\*.jpg)';...  '\*.jpeg', 'File Jpeg(\*.jpeg)';...  '\*.png', 'File PNG(\*.png)';...  '\*.tiff', 'File TIF(\*.tif)';...  '\*.gif', 'File GIF(\*.gif)';...  '\*.\*', 'All type(\*.\*)'},...  'Open Image');  if ~isequal (FileName, 0)  handles.data1 = imread(fullfile(PathName,FileName));  guidata (hObject, handles);  axes(handles.axes1);  imshow(handles.data1);  title('RGB');  else  return  end  % GRAYSCALE  image1 = handles.data1;  gray = rgb2gray(image1);  axes(handles.axes2);  imshow(gray);  title('Grayscale');  handles.data2 = gray;  guidata(hObject,handles); |

**Analisis :**

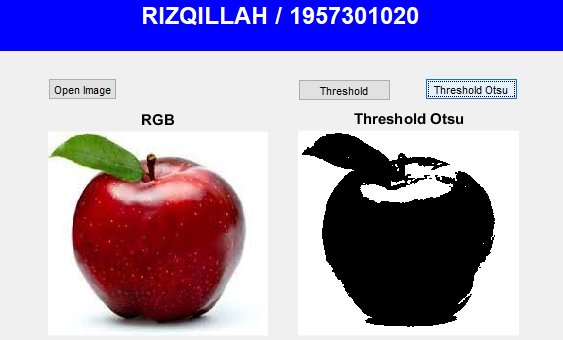
Program diatas dapat menghasilkan gambar seperti tersebut dikarenakan setelah melakukan operasi open image, program langsung melakukan eksekusi untuk mengubah gambar menjadi grayscale, dan gambar yang telah diubah ditampilkan pada axes2. Pengubahan gambar menjadi grayscale dapat dilakukan dikarenakan menggunakan fungsi rgb2gray() dan pada fungsi tersebut diisi dengan gambar yang diambil. Kemudian memilih axes yang akan ditampilkan beserta melakukan fungsi imshow() untuk menampilkan. Dan gambar grayscale yang telah dibuat disimpan dalam variabel handles.data2 jika nantinya akan digunakan untuk menyimpan gambar, maka variabel ini dapat digunakan.

1. **Cari perbedaan antara Threshold yang sudah dilakukan dengan Threshold Otsu dan apa kegunaan utama dari Threshold Otsu. Sertakan kodingnya**

**Hasil :**

****

Gambar 22 Output Threshold

****

Gambar 23 Output Threshold Otsu

**Program :**

**Threshold**

|  |
| --- |
| % --- Executes on button press in pushbutton2.  function pushbutton2\_Callback(hObject, eventdata, handles)  % hObject handle to pushbutton2 (see GCBO)  % handles structure with handles and user data (see GUIDATA)    % BLACK AND WHITE /Threshold  image1 = handles.data1;  bw = im2bw(image1);  axes(handles.axes2);  imshow(bw);  handles.data2 = bw;  guidata(hObject,handles);  title('Threshold'); |

**Threshold Otsu**

|  |
| --- |
| % --- Executes on button press in pushbutton3.  function pushbutton3\_Callback(hObject, eventdata, handles)  % hObject handle to pushbutton3 (see GCBO)  % handles structure with handles and user data (see GUIDATA)    % Threshold Otsu  image1 = handles.data1;  level = graythresh(image1);  bw = im2bw(image1,level);  axes(handles.axes2);  imshow(bw);  handles.data2 = bw;  guidata(hObject,handles);  title('Threshold Otsu'); |

**Analisis :**

Untuk menampilkan gambar threshold, hal pertama yang dilakukan adalah mengubah gambar menggunakan fungsi im2bw(). Kemudian mencetak gambar yang telah diubah tersebut.

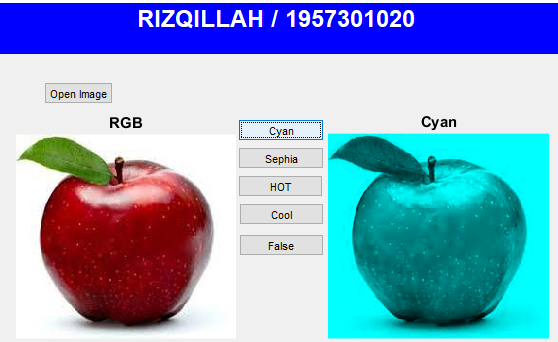
Dan untuk membuat gambar threshold otsu, pertama melakukan pengambilan nilai level dari gambar menggunakan fungsi graythresh(), setelah didapat level dari gambar kemudian melakukan pengubahan gambar dengan fungsi im2bw() dengan nilai dari gambar yang akan diubah beserta level yang diinginkan.

Adapun perbedaan threshold dengan threshold otsu adalah threshold mempunyai nilai yang lebih sedikit sehingga ada beberapa warna putih yang terdapat di beberapa area pada citra biner. Sementara threshold otsu memiliki nilai threshold yang lebih banyak, sehingga bagian berwarna putih ditutupi. Fungsi dari threshold otsu sendiri adalah membagi histogram gray level ke dalam dua daerah yang berbeda secara otomatis tanpa membutuhkan user untuk menginput nilai ambang.

1. **Memanipulasi gambar dengan warna Cyan, Sephia, Hot, Cool, dan False (warna dari gambar berbeda/berlawanan dengan warna aslinya)**

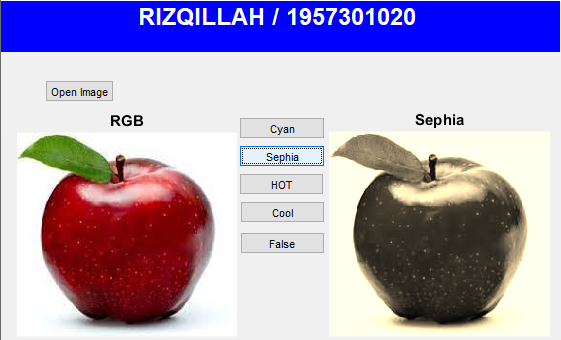
**Hasil :**

**Cyan**

****

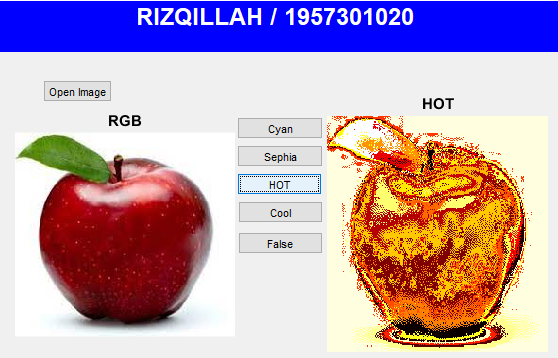
Gambar 24 Output Cyan

**Sephia**

****

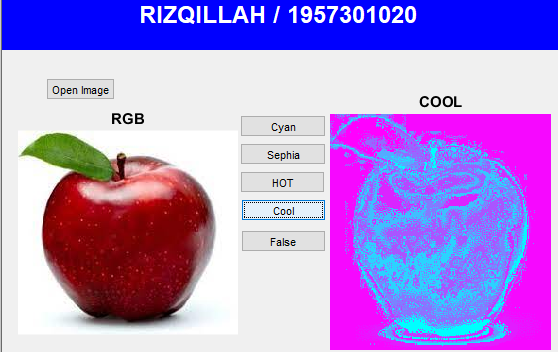
Gambar 25 Output Sephia

**Hot**

****

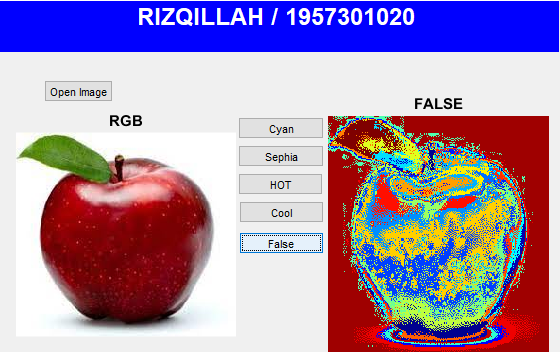
Gambar 26 Output Hot

**Cool**

****

Gambar 27 Output Cool

**False**

****

Gambar 28 Output False

**Program :**

|  |
| --- |
| % CYAN  image1 = handles.data1;  allBlack = zeros(size(image1,1),size(image1,2));    cyan = image1(:,:,1);  icyan = cat(3,allBlack,cyan,cyan);  axes(handles.axes2);  imshow(icyan);  handles.data2 = cyan;  title('Cyan');  guidata(hObject,handles);  % Sephia  d = handles.data1;  dR = d(:,:,1);  dG = d(:,:,2);  dB = d(:,:,3);    r = .393\*dR + .769\*dG + .189\*dB;  g = .349\*dR + .686\*dG + .168\*dB;  b = .272\*dR + .534\*dG + .131\*dB;    dSephia = cat(3,r,g,b);  axes(handles.axes2);  imshow(dSephia);  title('Sephia');  guidata(hObject,handles);  % HOT  image1 = handles.data1;  [Y] = rgb2ind(image1,28);  axes(handles.axes2);  imagesc(Y);  colormap(hot)  title('HOT');  axis off  % COOL  image1 = handles.data1;  [Y] = rgb2ind(image1,28);  axes(handles.axes2);  imagesc(Y);  colormap(cool);  title('COOL');  axis off  % FALSE  image1 = handles.data1;  [Y] = rgb2ind(image1,28);  axes(handles.axes2);  imagesc(Y);  colormap(jet);  title('FALSE');  axis off |

**Analisis :**

Untuk dapat membuat citra menjadi warna cyan adalah dengan cara mengatur pada variabel cyan dengan nilai dari image1 yang diambil. Kemudian pada variabel icyan melakukan fungsi cat(concatenate) dengan nilai dim=3, dan A1=allBlack dan A2 dan A3 dengan variabel cyan. Kemudian menampilkan citra cyan yang telah diubah tersebut ke axes2. Adapun allBlack disini bernilai 0 atau hitam. Dan variabel cyan diisi dengan warna dengan nilai 255. Maka hasil manipulasi akan muncul di axes2.

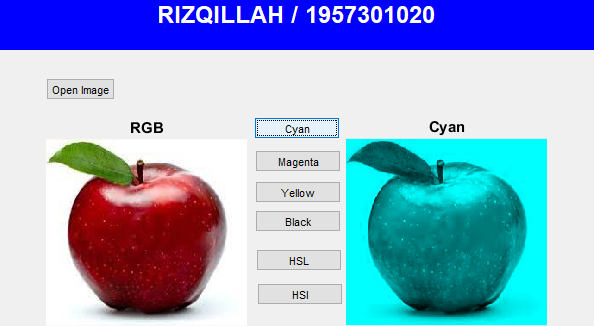
Adapun untuk membuat citra menjadi sephia adalah dengan cara membuat 3 variabel yang akan digunakan sebagai warna sephia, yaitu variabel dR, dG dan dB. Dan warna tersebut akan dilakukan perhitungan untuk warna RGB dari sephia. Setelah mendapat warna RGB sephia, kemudian melakukan concatenate dengan nilai dim=3, beserta A1,A2,dan A3 dengan nilai dari RGB yang telah didapat. Setelah kemudian mencetak variabel dSephia yang berisi citra yang telah diubah menjadi sephia.

Untuk mengubah citra menjadi hot adalah dengan cara mengambil nilai dari fungsi rgb2ind() yaitu nilai pertama adalah gambar yang akan diubah, dan nilai kedua adalah inmap yang diperlukan yaitu dengan nilai 28. Kemudian nilai yang telah didapat dari fungsi rgb2ind() atau convert rgb ke indexed image disimpan dalam variabel Y. Kemudian menampilkan gambar dengan fungsi imagesc() yaitu gambar ditampilkan dengan warna scale. Dan mengatur colormap yang diinginkan, yaitu dengan hot. Dan pada gambar cool juga dilakukan hal yang sama, hanya berbeda pada bagian colormap yaitu dengan nilai cool. Dan untuk citra false juga sama, hanya berbeda pemilihan colormap yaitu dengan nilai jet.

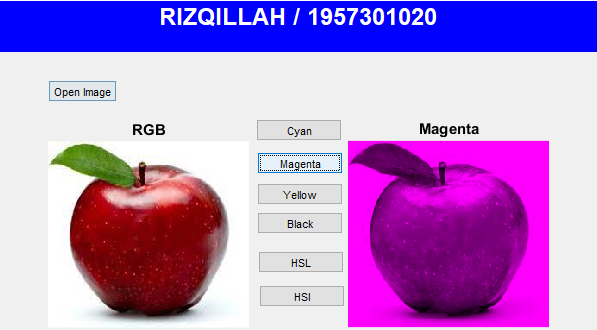
1. **Jelaskan apa itu warna CMYK, HSL, dan HSI. Berikan contohnya gambar serta koding untuk matlab!**

**Hasil :**

**CMYK**

****

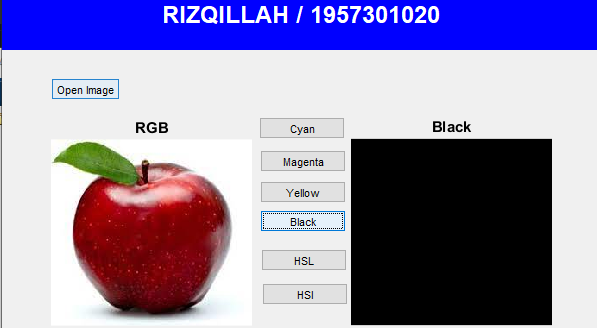
Gambar 29 Cyan



Gambar 30 Magenta

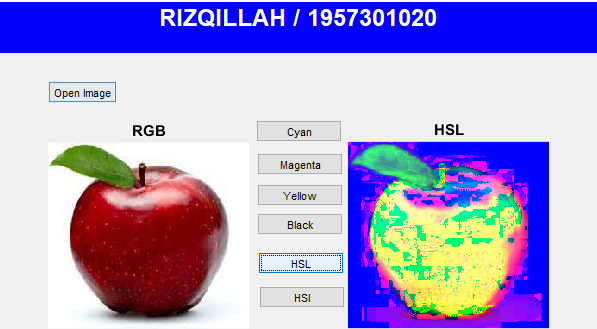


Gambar 31 Yellow



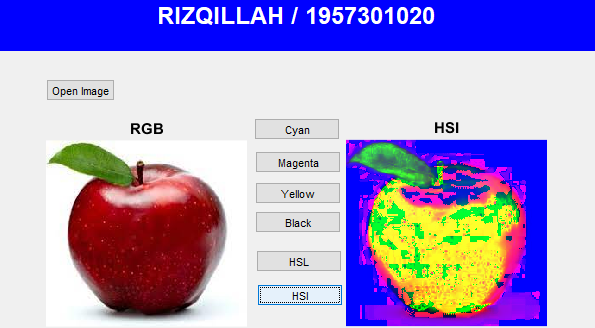
Gambar 32 Black

**HSL**

****

Gambar 33 HSL (Hue Saturation Lightness)

**HSI**

****

Gambar 34 HSI (Hue Saturation Intesity)

**Program :**

|  |
| --- |
| % CYAN  image1 = handles.data1;  allBlack = zeros(size(image1,1),size(image1,2));    cyan = image1(:,:,1);  icyan = cat(3,allBlack,cyan,cyan);  axes(handles.axes2);  imshow(icyan);  handles.data2 = cyan;  title('Cyan');  guidata(hObject,handles);  % MAGENTA  image1 = handles.data1;  allBlack = zeros(size(image1,1),size(image1,2));    magenta = image1(:,:,1);  imagenta = cat(3,magenta,allBlack,magenta);  axes(handles.axes2);  imshow(imagenta);  handles.data2 = imagenta;  title('Magenta');  guidata(hObject,handles);  % Yellow  image1 = handles.data1;  allBlack = zeros(size(image1,1),size(image1,2));    yellow = image1(:,:,1);  iyellow = cat(3,yellow,yellow,allBlack);  axes(handles.axes2);  imshow(iyellow);  handles.data2 = iyellow;  title('Yellow');  guidata(hObject,handles);  % Black  image1 = handles.data1;  allBlack = zeros(size(image1,1),size(image1,2));    iblack = cat(3,allBlack,allBlack,allBlack);  axes(handles.axes2);  imshow(iblack);  handles.data2 = iblack;  title('Black')  guidata(hObject,handles);  % HSL  image1 = handles.data1;  hsl = rgb2hsv(image1);  axes(handles.axes2);  imshow(hsl);  handles.data2 = hsl;  title('HSL')  guidata(hObject,handles);  % HSI  image1 = handles.data1;  hsi = rgb2hsi(image1);  axes(handles.axes2);  imshow(hsi);  handles.data2 = hsi;  title('HSI')  guidata(hObject,handles); |

**Analisis :**

CMYK (Cyan-Magenta-Yellow-blacK) atau biasanya disebut warna proses atau empat warna. CMYK adalah sebuah model warna berbasis pengurangan sebagian gelombang cahaya (substractive color model) dan yang umum dipergunakan dalam pencetakan berwarna. Alasan Cyan-Magenta-Yellow-Black disingkat menjadi CYMK dikarenakan K disini bermaksud key atau kunci yang berarti jika tidak ada warna hitam maka warna yang dihasilkan tidak akan lengkap.

Untuk mendapatkan citra Cyan, Magenta, Yellow dan Black adalah menggunakan metode yang sama, yang membedakannya hanya pada pengisian nilai dari fungsi cat() atau concatenate.

HSL atau Hue Saturation Lightness atau dapat juga disebut HSV adalah sebuah metode penamaan warna yang berdasarkan atas derajat roda warna “Hue” dengan pengaturan persentase kejenuhan warna “Saturation” dan persentase kepekatan cahaya “Lightness” sehingga akan menghasilkan warna baru yang disebut dengan warna HSL.

Untuk membuat citra menjadi HSL (Hue Saturation Lightness) adalah dengan cara menggunakan fungsi bawaan dari matlab yaitu fungsi rgb2hsv(). Yaitu dalam fungsi tersebut diisi nilai dengan gambar yang ingin diubah.

HSI atau Hue Saturation Intesity adalah dimana Hue disini berarti warna murni, Saturation adalah untuk mengatur derajat banyaknya warna murni yang dilunakkan dengan warna putih, dan Intesity adalah untuk menggabungkan warna dari Hue dan Saturation.

Untuk membuat citra menjadi HSI adalah dengan cara menggunakan fungsi rgb2hsi(). Dimana nilai dalam fungsi tersebut diisi dengan gambar yang akan diubah menjadi HSI (Hue Saturation Intesity).

# BAB III

**PENUTUP**

## Kesimpulan

Manipulasi Citra (Gambar) adalah sebuah proses melalui tampilan visual dari gambar asli yang telah diubah. Teknik-teknik ini dapat digunakan untuk memproduksi efek yang lebih luas, dari perubahan halus dan koreksi untuk lebih intervensi dramatis. Memanipulasi sebuah citra sama seperti halnya seperti melakukan pengeditan gambar melalui sebuah *software* seperti photoshop dan lain sebagainya. Adapun dalam modul 3 ini telah belajar mengenai cara memanipulasi citra menjadi sebuah *Grayscale*, *Black and White* (Biner), dan *Invers* (*Negative*).

Adapun tahap-tahap yang dilakukan untuk mengubah gambar RGB ke bentuk *grayscale* adalah dengan cara melakukan perhitungan pada tiap-tiap *pixel*.

Contohnya :

Red : 100 Green : 80 Blue : 210

Grayscale = 0.2989\*red(100) + 0.5870\*green(80) + 0.1140\*blue(210)

= 29.89 + 46.96 + 23.94

= 100.79

Berarti nilai *grayscale* pada *pixel* tersebut adalah 100.79. hal ini bisa didapatkan dikarenakan warna RGB memiliki besaran 24-bit atau berjumlah 16.777.216 variasi warna. Sedangkan *grayscale* memiliki besaran 16-bit atau berjumlah 65.536 variasi warna. Oleh karenanya *graycale* hanya memiliki warna dari putih dan hitam.

Adapun untuk merubah gambar menjadi warna biner atau *black and white* adalah dengan cara mengambil nilai dari *grayscale* yang telah didapat sebelumnya, dengan mempertimbangkan jika nilai tersebut ≥ 128 maka warnanya adalah putih, dan jika < 128 maka warnanya adalah hitam.

# DAFTAR PUSTAKA

Adi Pamungkas, M. (2015, September 29). *Complement Image (Negative Image)*. Retrieved from Pemrograman Matlab: https://pemrogramanmatlab.com/tag/citra-negatif/

Adi Pamungkas, M. (2015, Desember 27). *Pengolahan Citra Digital*. Retrieved from Pemrograman Matlab: https://pemrogramanmatlab.com/2015/12/27/pengolahan-citra-digital-rgb-grayscale-biner-menggunakan-gui-matlab/

Adi Pamungkas, M. (2016, September). *Thresholding | Pemrograman Matlab*. Retrieved from Pemrograman Matlab: https://pemrogramanmatlab.com/pengolahan-citra-digital/segmentasi-citra/thresholding-menggunakan-matlab/

Akbar, J. (2018, April 5). *Apa itu Pixel per Inch (PPI) dan bagaimana cara menghitungnya?* Retrieved from PuGam: https://www.pugam.com/1691/apa-itu-pixel-per-inch-ppi-dan-bagaimana-cara-menghitungnya/

Dyantama, D. (2016, December 14). *Citra Negatif*. Retrieved from Dendy Dyantama's Blog: https://student.blog.dinus.ac.id/dendy/2016/12/14/pengolahan-citra-digital-citra-negatif/

Hidayati, K. F. (2021, April 19). *Ingin Mendesain Gambar Digital? Pahami dulu seluk-beluk RGB!* Retrieved from Glints: https://glints.com/id/lowongan/warna-rgb-adalah

Ibnu, Y. (2021, Agustus 13). *Ketahui Bedanya PPI, DPI dan LPI dalam Desain Grafis*. Retrieved from PortalDekave: https://www.portaldekave.com/artikel/yuk-ketahui-bedanya-ppi-dpi-dan-lpi-dalam-desain-grafis

IT, A. (n.d.). *Cara Membuat Flipping Vertikal Citra di Matlab*. Retrieved from anakit.com: https://www.anakit.id/2018/08/cara-membuat-flipping-vertikal-citra-di.html

Jejaring. (2019, February 20). *Pengertian Grayscale*. Retrieved from Jejaring - Portal Belajar Indonesia: https://www.jejaring.web.id/pengertian-grayscale/

Matlab, P. (n.d.). *Pengolahan Citra Digital, Pengolahan Video, Pengenalan Pola, dan Data Mining*. Retrieved from pemrogramanmatlab.com: https://pemrogramanmatlab.com/tag/dasar-dasar-pengolahan-citra-digital/

Pann. (2019, April 14). *Grayscale - Apa Itu Grayscale?* Retrieved from Glosarium Online: https://glosarium.org/arti-grayscale-di-komputer/

Pranata. (n.d.). *Perbedaan PPI, DPI dan LPI dalam Desain Grafif*. Retrieved from PRANATA Printing: https://www.pranataprinting.com/perbedaan-ppi-dpi-dan-lpi-dalam-desain-grafis/

Sains, T. &. (2021, Juli 7). *DPI: Istilah yang harus kamu ketahui dalam Dunia Fotografi*. Retrieved from Kumparan: https://kumparan.com/how-to-tekno/dpi-istilah-yang-harus-kamu-ketahui-dalam-dunia-fotografi-1w5MXefZCD2/1