



DIGITAL IMAGE PROCESSING

Septian Cahyadi
Isnain Mulia
Edi Nurachmad
Febri Damatraseta

Digital Image Processing

Septian Cahyadi
Isnan Mulia
Edi Nurachmad
Febri Damatraseta



Digital Image Processing

Oleh:

Septian Cahyadi

Isnan Mulia

Edi Nurachmad

Febri Damatraseta

Edisi Pertama

Cetakan 1, Maret 2024

Anggota IKAPI No. 366

Editor : Team Kesatuan Press

ISBN :



© 2024. Kesatuan Press

Jl. Ranggagading No. 1, Bogor, Jawa Barat

Telp: (0251) 8381112

Faks: (0251) 8319925

e-mail: kesatuanpress@ibik.ac.id

Hak Cipta @2024 Pada Penulis.

Hak cipta dilindungi undang-undang. Dilarang memperbanyak sebagian atau seluruh isi buku ini dalam bentuk apa pun, baik secara elektronik maupun mekanis, termasuk tidak terbatas pada memfotokopi, merekam, atau dengan menggunakan sistem penyimpanan lainnya, tanpa izin tertulis dari Penerbit Kesatuan Press.

KATA PENGANTAR

Puji syukur ke hadirat Allah Yang Maha Kuasa, yang telah memberikan rahmat-Nya sehingga Buku Pengolahan Citra ini dapat diselesaikan dengan sebaik-baiknya. Tak lupa juga kami sampaikan terima kasih kepada seluruh pihak yang telah turut membantu dalam penyusunan buku ini. Rekan sejawat maupun mahasiswa yang telah berkenan untuk menyumbangkan *footage* untuk digunakan dalam pembelajaran melalui buku ini.

Buku ini secara khusus dibuat sebagai pedoman dalam melakukan kegiatan pembelajaran Pengolahan Citra ditingkat perguruan tinggi. Meski demikian, buku ini juga dapat digunakan sebagai referensi bagi yang mendalami dunia pengolahan citra digital. Buku ini diharapkan dapat membantu dalam mempersiapkan dan melaksanakan kegiatan pembelajaran dengan lebih baik, terarah, dan terencana. Pada setiap topik telah ditetapkan tujuan pelaksanaan kegiatan pembelajaran dan teori singkat untuk memperdalam pemahaman pembaca mengenai materi yang dibahas.

Penyusun meyakini bahwa dalam pembuatan Buku Pengolahan Citra ini masih jauh dari sempurna. Oleh karena itu penyusun mengharapkan kritik dan saran yang membangun guna penyempurnaan buku ajar ini dimasa yang akan datang. Akhir kata, penyusun mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung.

Bogor, September 2023

Penyusun

DAFTAR ISI

COVER	
KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	viii
SESI I PENGENALAN PENGOLAHAN CITRA	1
1.1. Tujuan Pembelajaran	1
1.2. Dasar Teori	1
1.2.1. Pengolahan Citra Digital	1
1.2.2. OpenCV.....	1
1.2.3. Google Colab	1
1.3. <i>Software</i>	2
1.4. Tahapan Kerja.....	2
1.4.1. Menyiapkan Ruang Kerja dan Mengimpor Library	2
1.4.2. Dasar Python	5
1.4.3. Membaca Citra	9
1.4.4. Menampilkan Citra pada Subplot.....	10
1.4.5. Jenis-Jenis Citra.....	12
1.4.6. Mengonversi Jenis Citra.....	16
1.5. Tugas dan Latihan	18
SESI II OPERASI PIKSEL DAN HISTOGRAM	19
2.1. Tujuan Pembelajaran	19
2.2. Dasar Teori	19
2.3. <i>Software</i>	19
2.4. Tahapan Kerja.....	19

2.4.1. Menyiapkan Ruang Kerja dan Mengimpor Library	19
2.4.2. <i>Spatial Resolution</i>	22
2.4.3. Histogram Citra	28
2.5. Tugas dan Latihan	29
SESI III PERBAIKAN KUALITAS CITRA PADA DOMAIN SPASIAL	31
3.1. Tujuan Pembelajaran	31
3.2. Dasar Teori	31
3.2.1. <i>Single Pixel Methods</i>	31
3.2.2. <i>Multiple Pixel Methods</i>	32
3.3. <i>Software</i>	33
3.4. Tahapan Kerja	33
3.4.1. Menyiapkan Ruang Kerja dan Mengimpor Library	33
3.4.2. Gray Level Transformation	35
3.4.3. Meningkatkan Kecerahan Citra (<i>Brightness</i>)	41
3.4.4. Meningkatkan Kontras Citra (<i>Contrast</i>)	42
3.4.5. <i>Multiple Value Pixel, Basic Spatial Filtering</i>	43
3.5. Tugas dan Latihan	45
SESI IV PERBAIKAN KUALITAS CITRA PADA DOMAIN FREKUENSI	46
4.1. Tujuan Pembelajaran	46
4.2. Dasar Teori	46
4.3. <i>Software</i>	46
4.4. Tahapan Kerja	47
4.4.1. <i>Fourier Transform</i>	47
4.4.2. Optimasi DFT	50
4.4.3. Operasi Filter dalam domain Frekuensi	51
4.5. Tugas dan Latihan	56

SESI V IMAGE RESTORATION	57
5.1. Tujuan Pembelajaran	57
5.2. Dasar Teori	57
5.2.1. Degradasi citra	57
5.2.2. Pemulihan Citra.....	58
5.3. <i>Software</i>	58
5.4. Tahapan Kerja	59
5.4.1. Filter Batas	59
5.4.2. Filter Pererataan	60
5.4.3. Filter Median	62
5.4.4. Menggeser Citra	63
5.4.5. Memutar Citra	64
5.4.6. Mendegradasi dan memperbaiki Citra	64
5.5. Tugas dan Latihan	68
SESI VI COLOR IMAGE PROCESSING	69
6.1. Tujuan Pembelajaran	69
6.2. Dasar Teori	69
6.3. <i>Software</i>	70
6.4. Tahapan Kerja.....	70
6.4.1. <i>Intensity Slicing</i>	70
6.4.2. <i>Multilevel Intensity Slicing</i>	72
6.4.3. <i>Color Coding</i>	73
6.4.4. <i>Grayscale To Color Transformation</i>	74
6.4.5. <i>Psudo Color</i>	76
6.4.6. <i>Full-Color Image and Variouis Color Space Components</i>	77
6.4.7. <i>Color Transformation</i>	80

6.4.8. <i>Color Complement</i>	80
6.4.9. <i>Color Slicing Transformation</i>	82
6.4.10. <i>Tonal Correction</i>	83
6.4.11. <i>Color Balancing Correction</i>	83
6.4.12. <i>Histogram Equalization Of A Full-Color</i>	84
6.4.13. <i>Color Image Smoothing</i>	85
6.4.14. <i>Color Segmentation In HSI</i>	85
6.4.15. <i>Color Image Compression</i>	86
6.5. Tugas dan latihan.....	87
SESI VII WAVELET DAN MULTIREOLUTION PROCESSING.....	88
7.1. Tujuan Pembelajaran	88
7.2. Dasar Teori	88
7.2.1 Wavelet	88
7.2.2 <i>Multiresolution Analysis</i>	88
7.2.3 Penerapan dalam Pengolahan Citra:.....	88
7.3. <i>Software</i>	89
7.4. Tahapan Kerja.....	89
7.4.1. Wavelet Transform.....	89
7.4.2. Wavelet Denoising	94
7.4.3. Wavelet Compression	95
7.4.4. Wavelet Denoising	96
7.5. Tugas Pembelajaran.....	99
SESI VIII IMAGE COMPRESSION	100
8.1. Tujuan Pembelajaran	100
8.2. Dasar Teori	100
8.3. <i>Software</i>	102

8.4. Tahapan Kerja	102
8.4.1. Loosless Image Compression.....	102
8.4.2. Loosless Image Compression.....	104
8.4.3. Discreet Cosine Transform	104
8.4.4. Discreet Cosine Transform dengan Quantization	106
8.4.5. Discreet Cosine Transform dengan Quantization	109
8.5. Tugas dan Latihan	110
SESI IX MORFOLOGI CITRA.....	112
9.1. Tujuan Pembelajaran	112
9.2. Dasar Teori	112
9.3. <i>Software</i>	112
9.4. Tahapan Kerja	112
9.4.1. Operator Nalar	112
9.4.2. Dilasi	115
9.4.3. Erosi	115
9.4.4. Elemen Penstrukturan	116
SESI X SEGMENTASI CITRA & EKSTRAKSI FITUR.....	118
10.1. Tujuan Pembelajaran.....	118
10.2. Dasar Teori.....	118
10.2.1. Segmentasi Citra	118
10.2.2. Ekstraksi Fitur	118
10.2.3. Segmentasi dan Ekstraksi Fitur dalam Kombinasi	118
10.3. <i>Software</i>	119
10.4. Tahapan Kerja	119
10.4.1. Deteksi Tepi.....	119
10.4.3. Deteksi Garis.....	122

10.4.4. Deteksi Kontur	123
10.4.5. Gray Co-Level Occurence Matrix.....	124
10.4.6. Jarak Citra	125
SESI XI APLIKASI TEMU KEMBALI CITRA	126
11.1. Tujuan Pembelajaran.....	126
11.2. Dasar Teori.....	126
11.3. <i>Software</i>	126
11.4. Tahapan Kerja	126
11.4.1. Menggunakan Google Colaboratory.....	126
11.4.2. Menggunakan Tkinter pada Replit.....	129
DAFTAR PUSAKA	135

DAFTAR GAMBAR

Gambar 1. Buat folder Pengolahan Citra.....	3
Gambar 2. Pilih Colaboratory	3
Gambar 3. Area kerja google colab	3
Gambar 4. Sidepanel data pada google colab	4
Gambar 5. Console google colab	4
Gambar 6. Buat folder Pengolahan Citra.....	20
Gambar 7. Pilih Colaboratory	20
Gambar 8. Area kerja google colab	21
Gambar 9. Sidepanel data pada google colab	21
Gambar 10. Console google colab	21
Gambar 11. Pilih Colaboratory	33
Gambar 12. Pilih Colaboratory	34
Gambar 13. Area kerja google colab	34
Gambar 14. Sidepanel data pada google colab	34
Gambar 15. Console google colab	35
Gambar 16. File-File pada Ruang Kerja Replit	134
Gambar 17. Aplikasi Temu Kembali Citra menggunakan Ekstraksi Fitur Tekstur GLCM	134

SESI I

PENGENALAN PENGOLAHAN CITRA

1.1. Tujuan Pembelajaran

1. Mampu memahami konsep pengolahan citra.
2. Mampu memahami penggunaan dasar OpenCV.
3. Mampu memahami jenis-jenis citra.
4. Mampu memahami cara konversi jenis citra.

1.2. Dasar Teori

1.2.1. Pengolahan Citra Digital

Pengolahan Citra Digital (*Digital Image Processing*) merupakan bidang ilmu yang mempelajari tentang bagaimana suatu citra itu dibentuk, diolah, dan dianalisis sehingga menghasilkan informasi yang dapat dipahami oleh manusia. Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra tersusun oleh sekumpulan piksel (*picture element*) yang memiliki koordinat (x, y) dan amplitudo $f(x, y)$. Koordinat (x, y) menunjukkan letak/posisi piksel dalam suatu citra, sedangkan amplitudo $f(x, y)$ menunjukkan nilai intensitas warna citra.

1.2.2. OpenCV

OpenCV (*Open Source Computer Vision Library*), adalah sebuah *library open source* yang dikembangkan oleh Intel yang berfokus untuk menyederhanakan *programming* terkait citra digital. Di dalamnya terdiri dari banyak fitur, antara lain : pengenalan wajah, pelacakan wajah, deteksi wajah, Kalman *filtering*, dan berbagai jenis metode AI (*Artificial Intellegence*) dan menyediakan berbagai algoritma sederhana terkait *Computer Vision* untuk *low level API*. OpenCV merupakan *open source computer vision library* untuk bahasa pemrograman C/C++, dan telah dikembangkan ke phyton, java, matlab.

1.2.3. Google Colab

Google Colab atau Google Colaboratory adalah sebuah eksekusi dokumen yang memungkinkan Anda untuk menyimpan, menulis, dan berbagi program melalui Google Drive. Ini mirip dengan Jupyter Notebook yang gratis dan berbasis cloud yang dapat diakses melalui peramban web seperti Mozilla Firefox dan Google Chrome. Dengan Google Colab, Anda dapat menjalankan kode Python tanpa perlu melakukan instalasi atau konfigurasi tambahan karena semua pengaturan dan penyesuaian dilakukan secara otomatis di cloud. Ini menjadikannya

tempat yang ideal bagi para pengembang yang ingin meningkatkan pemahaman mereka tentang Python.

Selain itu, Google Colaboratory juga dikenal karena memfasilitasi kolaborasi tim. Anda dapat mengedit notebook bersama dengan anggota tim lainnya, mirip dengan cara mengedit dokumen di Google Documents. Salah satu keunggulan utama dari Google Colaboratory adalah kehadiran koleksi pustaka *machine learning* yang populer, yang dapat dengan mudah dimuat ke dalam notebook Anda.

1.3. *Software*

1. Python 3, bahasa pemrograman untuk melakukan pengolahan citra. Anda bisa menggunakan bahasa lain untuk bereksperimen dengan pengolahan citra seperti Matlab/Octave.
2. OpenCV 4, *library* untuk melakukan pengolahan citra.
3. Scikit-Image, *library* untuk melakukan pengolahan citra seperti OpenCV tetapi memiliki beberapa fungsi tambahan untuk melengkapi proses dari OpenCV.
4. Matplotlib, *library* untuk melakukan *plotting* atau membuat grafik.
5. NumPy, *library* untuk melakukan manipulasi *array* dan operasi numerik.
6. Tkinter, *library* untuk membuat GUI pada *Python*.
7. Google Colaboratory, layanan *Jupyter Notebook* daring untuk menjalankan kode Python.
8. Replit, mirip seperti *Google Colab*, tetapi kode yang dibuat tidak dalam bentuk *notebook* dan memiliki dukungan untuk membuat GUI menggunakan *Tkinter*.

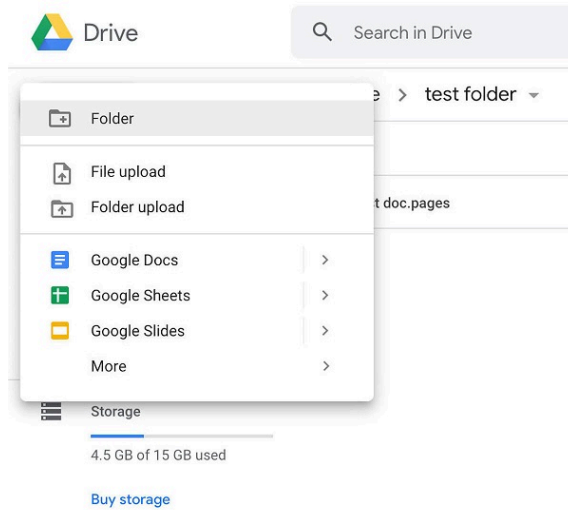
1.4. Tahapan Kerja

Pada materi pertama ini Anda akan menggunakan Jupyter Notebook yang disediakan oleh Google Colaboratory untuk mengeksekusi kode Python untuk menjalankan proses pengolahan citra.

1.4.1. Menyiapkan Ruang Kerja dan Mengimpor *Library*

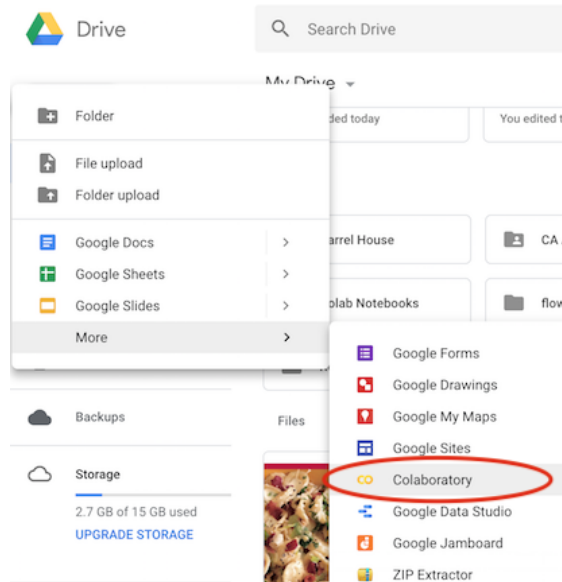
Sebelum melakukan kegiatan pembelajaran, Anda perlu menyiapkan ruang kerja terlebih dahulu. Anda perlu menginstall *library* yang akan digunakan pada kegiatan pembelajaran ini dan juga mengunduh beberapa sampel gambar yang akan digunakan untuk melakukan pengolahan citra. Ikuti langkah-langkah di bawah ini.

1. Pertama-tama, untuk menggunakan Google Colab, kamu harus memiliki akun Google lalu kemudian akses fitur Colaboratory.



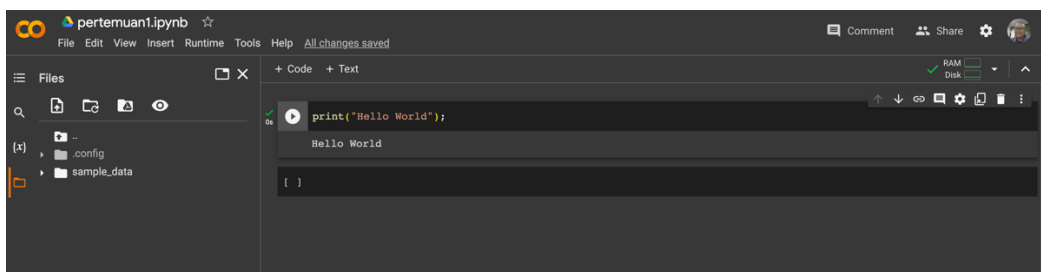
Gambar 1. Buat folder Pengolahan Citra

2. Membuat *notebook*

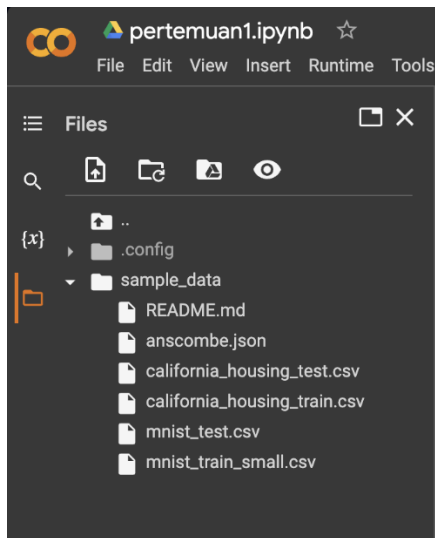


Gambar 2. Pilih Colaboratory

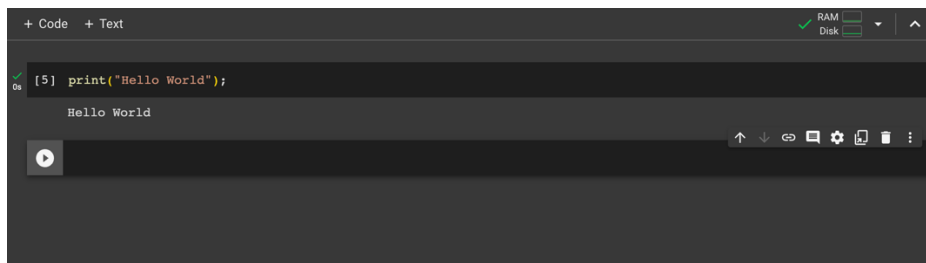
3. Tampilan <https://colab.research.google.com/>



Gambar 3. Area kerja google colab



Gambar 4. Sidepanel data pada google colab



Gambar 5. Console google colab

4. Download lalu upload atau salin semua footage yang ada pada folder drive berikut <https://bit.ly/footagepc> ke google drive kalian masing-masing.
5. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

<pre> 1 import numpy as np 2 import matplotlib.pyplot as plt 3 4 import cv2 5 from google.colab.patches import cv2_imshow </pre>	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1-5, mengimpor <i>library</i> yang akan digunakan untuk melakukan pengolahan citra.

Sekarang Anda sudah memahami bagaimana cara menyiapkan *notebook* pada Google Colab untuk melakukan pengolahan citra. Anda akan selalu menggunakan *template* di atas untuk semua sesi pada kegiatan pembelajaran pengolahan citra.

1.4.2. Dasar Python

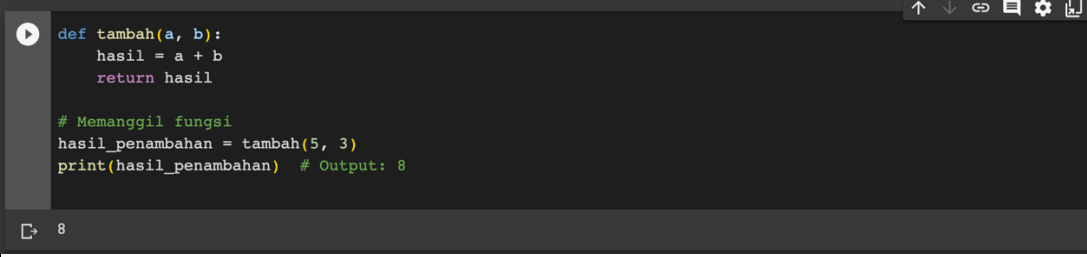
A. Tipe data Python

<pre> 1 a = 1 #integer 2 b = 0.5 # float 3 c = "Satu" #String 4 d = [1,"Dua",3,4.5] #Variabel list pada python 5 e = (1,2,3,4) #tuple, mirip list tapi tak bisa diubah nilainya 6 f = {1:"first name",2:"last name", "age":33} #Dictionary 7 8 print(a) 9 print(b) 10 print(c) 11 print(d) 12 print(e) 13 print(f) </pre>	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1 deklarasi tipe data integer • Baris 2 deklarasi tipe data float • Baris 3 deklarasi tipe data string • Baris 4 deklarasi struktur data list • Baris 5 deklarasi struktur data tuple • Baris 6 deklarasi struktur data Dictionary
<p>Output</p> <pre> 1 0.5 Satu [1, 'Dua', 3, 4.5] (1, 2, 3, 4) {'1': 'first name', 2: 'last name', 'age': 33} </pre>	

Secara umum tipe data pada Python tidak berbeda dengan Bahasa pemrograman lain, seperti diantaranya integer, float, dan string. Dalam bahasa pemrograman Python, tuple adalah salah satu jenis struktur data yang digunakan untuk menyimpan kumpulan elemen. Namun, ada beberapa perbedaan utama antara tuple dan list. Pertama, tidak dapat diubah (*Immutable*): Tuple bersifat tidak dapat diubah, artinya setelah sebuah tuple dibuat, elemennya tidak dapat dimodifikasi, ditambahkan, atau dihapus. Ini berbeda dengan list, yang dapat diubah (*mutable*). Kedua, deklarasi: Tuple dideklarasikan dengan menggunakan tanda kurung (), sedangkan list dideklarasikan dengan tanda kurung siku [].

B. Mendefinisikan fungsi

Di Python, Anda dapat mendefinisikan fungsi menggunakan kata kunci `def`. Fungsi adalah blok kode yang digunakan untuk mengeksekusi tugas tertentu. Fungsi digunakan untuk mengelompokkan sejumlah pernyataan agar dapat digunakan kembali dengan cara yang mudah. Berikut adalah contoh cara mendefinisikan fungsi:

<pre> 1 def tambah(a, b): 2 hasil = a + b 3 return hasil 4 5 # Memanggil fungsi 6 hasil_penambahan = tambah(5, 3) 7 print(hasil_penambahan) # Output: 8 </pre>	
	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1-3, deklarasi fungsi pada python • Baris 6, memanggil fungsi tambah • Baris 7, mencetak fungsi
	

C. Logic Control dan Looping

Percabangan digunakan untuk membuat sebuah kode yang cara berjalannya kode ditentukan oleh pengetesan pernyataan bersyarat. Sama seperti fungsi pengecekan pada boolean, pada percabangan ini juga dilakukan dengan memanfaatkan operator pembandingan. Ada tiga jenis pernyataan yang akan kita pelajari pada kesempatan kali ini, yaitu pernyataan if, pernyataan if - else, dan pernyataan if - elif - else.

<pre> 1 planets = ['Mercury', 'Venus', 'Earth', 'Mars', 'Jupiter', 'Saturn', 'Uranus', 2 'Neptune'] 3 for planet in planets: 4 print(planet, end=' ') # cetak dalam 1 baris 5 print("\n") 6 for i in range(5): 7 print("Doing important work. i =", i) 8 9 areas = [11.25, 18.0, 20.0, 10.75, 9.50] 10 print("\n") 11 for index,area in enumerate(areas) : 12 print("room"+str(index)+" : "+str(area)) </pre>	
	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, deklarasi list • Baris 2-3, perintah for untuk memanggil variable list • Baris 5-6, perintah for dengan menggunakan range() • Baris 11-12, perintah for dengan menggunakan fungsi enumerate()

```

☐ Mercury Venus Earth Mars Jupiter Saturn Uranus Neptune

Doing important work. i = 0
Doing important work. i = 1
Doing important work. i = 2
Doing important work. i = 3
Doing important work. i = 4

room0: 11.25
room1: 18.0
room2: 20.0
room3: 10.75
room4: 9.5

```

```

1 i = 0
2 while i < 10:
3     print(i, end=' ')
4     i += 1
5
6 print("\n")
7 while True:
8     a = input('input string: ')
9     if a == 'quit':
10        break
11    print('length of a is {}'.format(len(a)))
12
13 fruits = ['apple', 'banana', 'orange', 'grape', 'pineapple']
14 for fruit in fruits:
15     if fruit == 'orange':
16         continue
17     print('my fruit is {}'.format(fruit))

```

Penjelasan Kode

- Baris 2-4, menggunakan perintah while, argumen dari while loop dievaluasi setiap looping, dan loop dijalankan sampai hasil evaluasi dari conditional operator bernilai False.
- Baris 7-11, perintah while dengan break, ketika input berupa 'quit' program akan mengakhiri looping.
- Baris 14-17, perintah for looping dengan continue, iterasi akan berhenti ketika memenuhi kondisi 'orange' dan melanjutkan ke iterasi selanjutnya.

```

0 1 2 3 4 5 6 7 8 9

input string: quit
my fruit is apple
my fruit is banana
my fruit is grape
my fruit is pineapple

```

D. Exceptional Handling

```

1 a = int(input())
2 b = int(input())
3
4 try:
5     c = a/b
6 except Exception as e:
7     print('raising error in:', e)
8 finally:
9     print('Finish')

```

	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 4, Try : suatu statement yang menguji kode error • Baris 6, Except : suatu statement yang handle apabila ada error • Baris 8, Finally : suatu statement yang mengeksekusi suatu kode • Ketika ada suatu error di sebuah kode, Python biasanya akan berhenti mengeksekusi kode selanjutnya dan menampilkan error.
	<pre> 2 0 raising error in: division by zero Finish </pre>

E. Numpy

<pre> 1 import numpy as np 2 3 my_numpy = np.array([4 [1,2,3,4,5], 5 [6,7,8,9,0] 6]) 7 print(my_numpy) 8 print(my_numpy.ndim) 9 print(my_numpy.shape) 10 print(my_numpy.dtype) 11 print(my_numpy.itemsize) 12 print(my_numpy.nbytes) </pre>	
	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, mengimport package numpy dengan nama np • Baris 3, deklarasi array dengan ordo 2x5 • Baris 7, mencetak variable numpy • Baris 8, Dimension, digunakan untuk melihat dimensi yang dimiliki suatu numpy array • Baris 9, Shape, digunakan untuk mengetahui berapa jumlah baris dan kolom dari suatu numpy array • Baris 10, type, digunakan untuk mengetahui tipe data yang dimiliki suatu numpy array • Baris 11, itemsize, untuk melihat ukuran memori yang dibutuhkan untuk menyimpan sebuah numpy array • Baris 12, nbytes, digunakan untuk menghitung total bytes yang dimiliki suatu numpy array.
	<pre> ☐→ [[1 2 3 4 5] [6 7 8 9 0]] 2 (2, 5) int64 8 80 </pre>

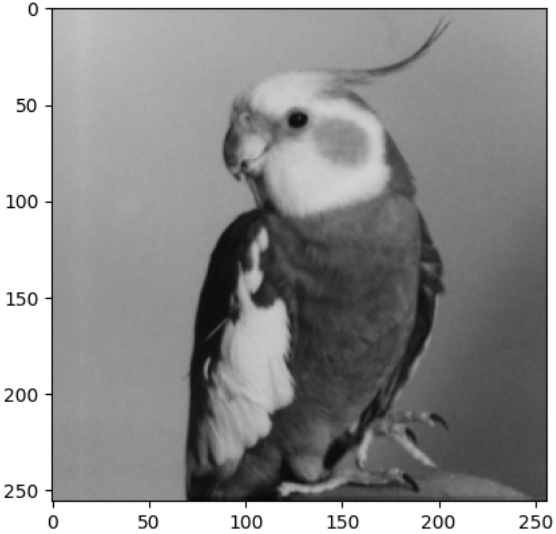
1.4.3. Membaca Citra

Salah satu proses yang paling penting dalam pengolahan citra adalah memuat file citra atau gambar yang akan Anda olah. Pada sesi ini Anda akan menggunakan *library* OpenCV untuk membaca file citra kemudian menampilkannya.

1. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.


1	<code>img = cv2.imread("/content/drive/MyDrive/Footage/bird.tif")</code>
2	<code>print(img.shape)</code>
	Output Sel (256, 256, 3)
	Penjelasan Kode <ul style="list-style-type: none">• Baris 1, membaca file <i>bird.tif</i>, file ini diunduh melalui <i>script</i> yang telah Anda eksekusi pada sub sesi sebelumnya. Kembalian dari fungsi ini adalah <i>array NumPy</i> yang berisi data piksel dari citra yang dimuat.• Baris 2, menampilkan bentuk atau dimensi <i>array</i> dari citra yang dimuat. Pada umumnya citra yang dimuat memiliki dimensi (tinggi, lebar, kanal) atau (<i>height, width, channels</i>). Artinya, citra yang dimuat pada sel ini memiliki ukuran 256x256 piksel dengan tiga kanal warna (BGR).

2. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1	<code>plt.imshow(img)</code>
	Output Sel 
	Penjelasan Kode <ul style="list-style-type: none">• Baris 1, menampilkan citra yang dimuat menggunakan <i>library Matplotlib</i>.

3. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1	<code>cv2_imshow(img)</code>
---	------------------------------

	<p>Output Sel</p> 
	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, menampilkan citra yang dimuat menggunakan <i>library OpenCV</i>.

Pada tahap ini Anda telah mempelajari bagaimana cara memuat dan menampilkan citra menggunakan *library OpenCV* dan *Matplotlib*. Anda dapat memuat citra menggunakan perintah `imread` dan menampilkan citra menggunakan `plt.show` dan `cv2_imshow`. Tidak ada perbedaan yang signifikan antara kedua fungsi tersebut, Anda bisa menggunakan salah satunya.

1.4.4. Menampilkan Citra pada Subplot

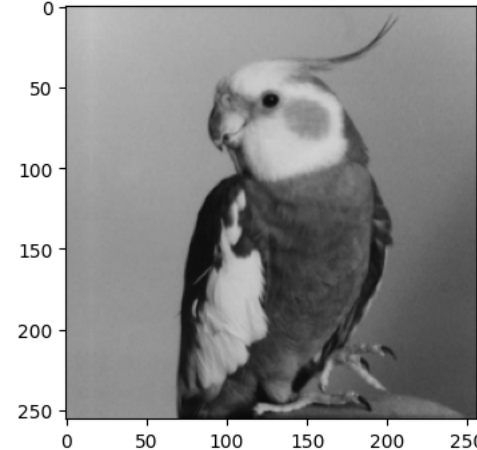
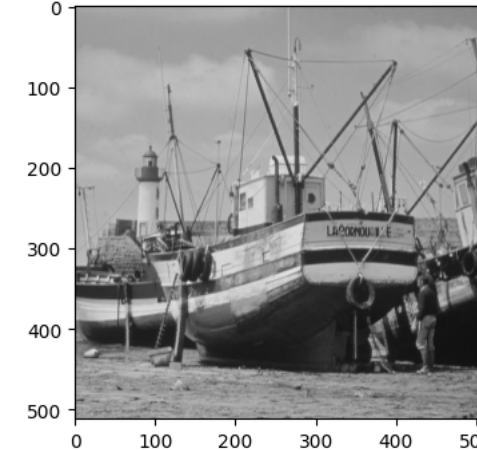
Misalnya Anda memiliki dua buah citra dan Anda ingin menampilkan dua citra tersebut secara berdampingan, maka Anda dapat menggunakan fitur *subplot* pada *Matplotlib* untuk membuat kisi dan menampilkan banyak citra sebagai satu figur.

1. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1 2 3 4 5	<pre>img1 = cv2.imread("/content/drive/MyDrive/Footage/bird.tif") print(img1.shape) img2 = cv2.imread("/content/drive/MyDrive/Footage/boat.tif") print(img2.shape)</pre>
	<p>Output Sel</p> <pre>(256, 256, 3) (512, 512, 3)</pre>
	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1-5, memuat dua citra dan menampilkan dimensinya.

2. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1 2 3 4 5 6	<pre>fig, axes = plt.subplots(1, 2, figsize=(10, 4)) axes[0].imshow(img1) axes[1].imshow(img2) plt.show()</pre>
	<p>Output Sel</p>

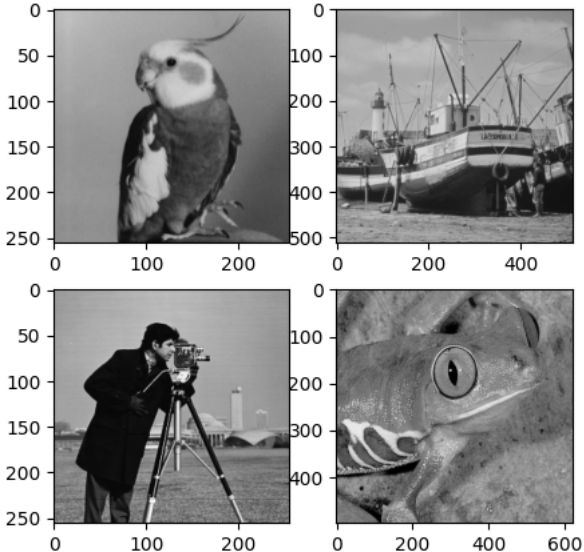
	
<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, membuat <i>subplot</i> dengan satu baris dan dua kolom kisi pada figur dengan ukuran 10x4. Pengembalian dari fungsi ini adalah objek <i>Figure</i> dan <i>array axes</i> yang berisi <i>axis</i> untuk membuat plot. • Baris 3, menampilkan citra pada <i>axis</i> pertama (baris 1, kolom 1). • Baris 4, menampilkan citra pada <i>axis</i> kedua (baris 1, kolom 2). • Baris 6, menampilkan figur. 	

3. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

<pre>1 2 3 4 5</pre>	<pre>img3 = cv2.imread("/content/drive/MyDrive/Footage/camera.tif") print(img3.shape) img4 = cv2.imread("/content/drive/MyDrive/Footage/frog.tif") print(img4.shape)</pre>
<p>Output Sel (256, 256, 3) (498, 621, 3)</p>	
<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1-5, memuat citra tambahan untuk membuat plot yang lebih besar. 	

4. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

<pre>1 2 3 4 5 6 7</pre>	<pre>fig, axes = plt.subplots(2, 2, figsize=(5, 5)) axes[0, 0].imshow(img1, aspect="auto") axes[0, 1].imshow(img2, aspect="auto") axes[1, 0].imshow(img3, aspect="auto") axes[1, 1].imshow(img4, aspect="auto") plt.show()</pre>
--------------------------	---

	<p>Output Sel</p> 
	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, membuat <i>subplot</i> dengan dua baris dan dua kolom kisi pada figur dengan ukuran 5x5. • Baris 3-7, menampilkan citra pada kisi <i>axis</i>. Apabila Anda hanya memiliki satu baris atau satu kolom, maka <i>array axes</i> akan memiliki satu dimensi, apabila Anda menggunakan dua atau lebih baris dan kolom, maka <i>array axes</i> akan memiliki dua dimensi yang menunjukkan baris dan kolom. Parameter <i>aspect</i> menyatakan bagaimana <i>Matplotlib</i> harus menampilkan citra apabila ukuran citra tidak sama dengan ukuran <i>axis</i>.

Pada tahap ini Anda seharusnya sudah memahami bagaimana cara membuat *subplots* untuk menampilkan banyak citra sebagai kisi. Baca kembali dokumentasi *Matplotlib* untuk mengetahui bagaimana cara membuat *subplots* yang lebih kompleks.

1.4.5. Jenis-Jenis Citra

Terdapat tiga jenis citra yang umum digunakan pada pengolahan citra, yaitu citra berwarna (*color*), citra berskala keabu-abuan (*grayscale*), dan citra biner (*binary*).

A. Citra Berwarna

Citra berwarna secara umum merupakan citra yang terdiri atas tiga komponen warna yaitu RGB (*red, green, blue*). Tiap-tiap komponen ini disebut sebagai kanal atau *channel*. Umumnya citra memiliki skala delapan bit yang berarti nilai intensitas piksel berada pada rentang 0-255. Secara *default*, *OpenCV* akan memuat citra dalam format BGR (bukan RGB).

1. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

<pre>1 img = cv2.imread("/content/drive/MyDrive/Footage/lena3.tif") 2 print(img.shape) 3 cv2_imshow(img)</pre>	
	<p>Output Sel</p>

	(512, 512, 3)
	
<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, memuat citra daun. • Baris 2, menampilkan dimensi citra. Dapat dilihat bahwa citra ini memiliki ukuran 255x191 piksel dengan tiga kanal warna. • Baris 3, menampilkan citra. 	

2. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.


1	<code>img[:3, :3, :]</code>
<p>Output Sel</p> <pre>array([[125, 137, 226], [125, 137, 226], [133, 137, 223]], [[125, 137, 226], [125, 137, 226], [133, 137, 223]], [[125, 137, 226], [125, 137, 226], [133, 137, 223]]], dtype=uint8)</pre>	
<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, menampilkan tiga elemen pertama pada setiap baris (<i>tinggi/height</i>) dan kolom (<i>lebar/width</i>) pada semua kanal pada citra. Dapat dilihat bahwa citra yang dimuat pada dasarnya adalah kumpulan piksel berupa angka dengan jangkauan 0-255 yang merepresentasikan intensitas warna pada kanal tersebut. Kelompok <i>array</i> pertama menyatakan intensitas warna pada kanal B, kelompok <i>array</i> kedua menyatakan intensitas warna pada kanal G, dan kelompok <i>array</i> ketiga menyatakan intensitas warna pada kanal R. 	

Sekarang Anda telah memahami bagaimana suatu citra dapat terbentuk. Citra berwarna memiliki tiga kanal yaitu RGB yang merepresentasikan warna untuk tiap-tiap piksel yang akhirnya bersatu menghasilkan berbagai warna dari tiga warna primer. Kanal warna RGB ini sering disebut sebagai ruang warna atau *color space*. Selain RGB, terdapat berbagai jenis ruang warna seperti HSV (*hue, saturation, value*), CMYK (*cyan, magenta, yellow, black*), dan HSL (*hue, saturation, lightness*).

B. Citra Berskala Keabu-abuan

Citra berskala keabu-abuan hanya memiliki satu kanal untuk merepresentasikan intensitas cahaya pada tiap-tiap piksel. Citra ini berguna untuk melakukan koreksi citra dan mengonversi citra ke citra biner.

1. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1 2 3	<pre>img = cv2.imread("/content/drive/MyDrive/Footage/peppers3.tif", cv2.IMREAD_GRAYSCALE) print(img.shape) cv2_imshow(img)</pre>
<p>Output Sel (512, 512)</p> 	
<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, memuat citra daun menggunakan format keabu-abuan (<i>grayscale</i>). • Baris 2, menampilkan dimensi citra. Dapat dilihat bahwa citra ini memiliki ukuran (512 x 512) piksel dengan satu kanal warna. • Baris 3, menampilkan citra. 	

2. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1	<pre>img[:3, :3]</pre>
<p>Output Sel array([[0, 0, 0], [0, 116, 115], [0, 119, 115]], dtype=uint8)</p>	
<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, menampilkan tiga elemen pertama pada setiap baris (<i>tinggi/height</i>) dan kolom (<i>lebar/width</i>) pada citra. Dapat dilihat bahwa citra yang dimuat pada dasarnya adalah kumpulan piksel berupa angka dengan jangkauan 0-255 yang merepresentasikan intensitas warna pada kanal tersebut. Karena ini merupakan citra berskala keabu-abuan, kanal pada citra ini hanya ada satu. 	

Pada tahap ini Anda sudah memahami bahwa pada citra berskala keabu-abuan hanya terdapat satu kanal warna yang digunakan untuk merepresentasikan intensitas cahaya pada tiap-tiap piksel pada citra. Berdasarkan pengamatan ini dapat diambil simpulan bahwa untuk


membentuk citra berwarna komputer menggunakan gabungan dari warna-warna primer (RGB) untuk menghasilkan warna-warna baru.

C. Citra Biner

Citra biner merupakan citra yang hanya memiliki dua rentang nilai, yaitu 0 (hitam) dan 255 (putih). Citra ini dapat digunakan untuk berbagai proses pada pengolahan citra seperti deteksi tepi dan segmentasi citra.

1. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1	<code>img = cv2.imread("/content/drive/MyDrive/Footage/peppers3.tif", cv2.IMREAD_GRAYSCALE)</code>
2	<code>thresh, img_bw = cv2.threshold(img, 128, 255, cv2.THRESH_BINARY cv2.THRESH_OTSU)</code>
3	
4	<code>print(img_bw.shape)</code>
5	<code>cv2_imshow(img_bw)</code>
Output Sel	

	(512, 512)
	
<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, memuat citra daun menggunakan format keabu-abuan (<i>grayscale</i>). • Baris 2, melakukan proses <i>threshold</i> untuk mengonversi citra berskala abu-abu menjadi citra biner. Proses ini akan dijelaskan pada sub sesi berikutnya. • Baris 4, menampilkan dimensi citra. Dapat dilihat bahwa citra ini memiliki ukuran 512x512 piksel dengan satu kanal warna. • Baris 3, menampilkan citra. 	

2. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1	<code>img[:3, :3]</code>
Output Sel	
<pre>array([[0, 0, 0], [0, 116, 115], [0, 119, 115]], dtype=uint8)</pre>	
<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, menampilkan tiga elemen pertama pada setiap baris (<i>tinggi/height</i>) dan kolom (<i>lebar/width</i>) pada citra. Pada contoh ini hanya terdapat nilai 255 yang berarti warna putih. 	

Pada tahap ini Anda telah memahami tiga jenis citra yaitu citra berwarna, citra berskala keabu-abuan, dan citra biner. Pada citra biner, hanya terdapat dua nilai yang merepresentasikan piksel yaitu 0 (hitam) dan 255 (putih).

1.4.6. Mengonversi Jenis Citra


Sering kali pada proses pengolahan citra Anda perlu melakukan konversi antara beberapa jenis citra karena terdapat beberapa proses yang hanya bisa dilakukan pada jenis citra tertentu. Maka dari itu Anda perlu mempelajari bagaimana cara melakukan konversi jenis citra.

A. Konversi Citra Berwarna ke Citra Keabuan

1. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1	<code>img = cv2.imread("/content/drive/MyDrive/Footage/peppers3.tif")</code>
2	<code>print(img.shape)</code>
	Output Sel (512, 512, 3)
	Penjelasan Kode • Baris 1-2, memuat citra dan menampilkan dimensi citra.

2. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1	<code>img_gray = np.array(0.1140 * img[:, :, 0] + 0.5870 * img[:, :, 1] + 0.2989 * img[:, :, 2])</code>
2	<code>cv2_imshow(img_gray)</code>
	Kode Alternatif
1	<code>img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)</code>
	Output 

<ul style="list-style-type: none"> • Baris 1, melakukan proses penjumlahan menggunakan bobot untuk tiap-tiap <i>channel</i> pada citra untuk mendapatkan tingkat pencahayaan atau luminositas (<i>luminosity</i>). <p>Penjelasan Kode Alternatif</p> <ul style="list-style-type: none"> • Baris 1, melakukan konversi BGR ke <i>grayscale</i> menggunakan algoritma yang sama tetapi menggunakan fungsi bantuan dari <i>OpenCV</i>.
--

B. Konversi Citra Keabuan ke Citra Biner


1. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1	<code>img = cv2.imread("/content/drive/MyDrive/Footage/peppers3.tif",</code>
2	<code>cv2.IMREAD_GRAYSCALE)</code>
	<code>print(img.shape)</code>
	Output Sel (512, 512, 3)
	Penjelasan Kode • Baris 1-2, memuat citra dan menampilkan dimensi citra.

2. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.


1	<code>ambang = 160</code>
2	<code>img_biner = np.zeros(img_gray.shape)</code>
3	
4	<code>for h in range(img_gray.shape[0]):</code>
5	<code>for w in range(img_gray.shape[1]):</code>
6	<code>if img_gray[h, w] >= ambang:</code>
7	<code>img_biner[h, w] = 0</code>
8	<code>else:</code>
9	<code>img_biner[h, w] = 255</code>
	Kode Alternatif 1
1	<code>ambang = 220</code>
2	<code>_, img_biner = cv2.threshold(img_gray, ambang, 255, cv2.THRESH_BINARY)</code>
	Kode Alternatif 2
1	<code>thresh, img_biner = cv2.threshold(img_gray, 0, 255, cv2.THRESH_BINARY </code>
	<code>cv2.THRESH_OTSU)</code>
	Penjelasan Kode
	<ul style="list-style-type: none"> • Baris 1, menyatakan ambang batas intensitas piksel untuk menentukan titik potong, apabila nilai piksel di atas nilai <i>threshold</i> ini, maka piksel tersebut akan diubah menjadi 255 dan sebaliknya apabila di bawah nilai <i>threshold</i> maka piksel tersebut akan diubah menjadi 0. • Baris 2, menciptakan <i>array NumPy</i> baru dengan dimensi yang sama seperti citra asli dengan seluruh elemen bernilai nol. • Baris 4-5, melakukan iterasi untuk semua kolom dan baris pada <i>array</i> piksel. • Baris 6-9, kondisional apabila nilai pada piksel (<i>h, w</i>) lebih besar atau sama dengan nilai ambang, maka nilai pada piksel tersebut akan diubah menjadi 255, selain itu maka akan diubah menjadi 0.
	Penjelasan Kode Alternatif 1
	<ul style="list-style-type: none"> • Baris 1, menyatakan ambang atau <i>threshold</i>. • Baris 2, melakukan proses <i>threshold</i> menggunakan nilai ambang.
	Penjelasan Kode Alternatif 2
	<ul style="list-style-type: none"> • Baris 1, melakukan proses <i>threshold</i> dengan menggunakan metode <i>Otsu</i> untuk menentukan nilai <i>threshold</i> yang optimal untuk citra yang digunakan. Nilai <i>threshold</i> yang digunakan akan dikembalikan pada variabel <i>thresh</i>.

3. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1	<code>cv2_imshow(img_biner)</code>
Output Sel	
Penjelasan Kode	<ul style="list-style-type: none">• Baris 1, menampilkan citra

1.5. Tugas dan Latihan

1. Apa yang dimaksud dengan pengenalan citra?
2. Mengapa Python sering digunakan dalam pengenalan citra?
3. Apa itu OpenCV, dan mengapa itu penting dalam pengenalan citra?
4. Bagaimana cara membaca dan menampilkan sebuah citra menggunakan Python dan OpenCV?
5. Apa perbedaan antara citra berwarna (RGB) dan citra grayscale, dan bagaimana cara mengonversi citra dari satu format ke yang lain menggunakan OpenCV?

1 2 3 4 5 6 7 8 9 10 11	<pre> fig, axes = plt.subplots(1, len(contours), figsize=(10, 1)) for i, contour in enumerate(contours): x, y, w, h = cv2.boundingRect(contour) roi = img_thresh[y:y+h, x:x+w] axes[i].imshow(roi, cmap="gray") axes[i].set_aspect('auto') axes[i].set_yticklabels([]) axes[i].set_xticklabels([]) plt.show() </pre>
	<p>Output Sel</p> 
	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, membuat <i>subplots</i> dengan jumlah kolom sesuai dengan banyaknya kontur yang ditemukan. • Baris 3, melakukan iterasi untuk semua kontur yang ditemukan. • Baris 4, mencari titik-titik dan ukuran kontur pada citra. • Baris 6-9, menampilkan citra sesuai dengan lokasi kontur yang dideteksi. • Baris 11, menampilkan figur.

10.4.5. Gray Co-Level Occurrence Matrix

1. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1 2	<pre> img = cv2.imread("daun.png", cv2.IMREAD_GRAYSCALE) print(img.shape) </pre>
	<p>Output Sel (191, 255)</p>
	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, membaca dan menampilkan dimensi citra.

2. Buat sel baru, kemudian masukan dan eksekusi sel kode berikut.

1 2 3 4 5	<pre> glcm = graycomatrix(img, distances=[5], angles=[0], levels=256, symmetric=True, normed=True) print("Correlation: ", graycoprops(glcm, 'correlation').item(0)) print("Contrast: ", graycoprops(glcm, 'Contrast').item(0)) print("Homogeneity: ", graycoprops(glcm, 'homogeneity').item(0)) print("Energy: ", graycoprops(glcm, 'energy').item(0)) </pre>
	<p>Output Sel</p> <pre> Correlation: 0.9201844457414212 Contrast: 519.4916020942411 Homogeneity: 0.3104028468461741 Energy: 0.05201956182670323 </pre>
	<p>Penjelasan Kode</p> <ul style="list-style-type: none"> • Baris 1, menghitung matriks <i>Gray Co-occurrence Level Matrix</i>. • Baris 2-5, menghitung fitur tekstur dari matriks GLCM seperti <i>correlation</i>, <i>Contrast</i>, <i>homogeneity</i>, dan <i>energy</i>.

DAFTAR PUSAKA

- Gonzalez, Rafael C., Woods, Richard E., dan Eddins, Steven L. 2009. Digital Image Processing Using Matlab, 2nd edition. Gatesmark Publishing. United States.
- Marques, Oge. 2011. Practical Image and Video Processing Using Matlab. Wiley-Blackwell. Singapore.
- Solomon, Chris., Breckon, Toby. 2011. Fundamentals of Digital Image Processing; A Practical Approach with Example in Matlab. Wiley-Blackwell.

Website :

[https://colab.research.google.com/drive/1JW-](https://colab.research.google.com/drive/1JW-DI1PSsVzO51IXYYv8DPyZ6eh4RWx5?usp=sharing)

[DI1PSsVzO51IXYYv8DPyZ6eh4RWx5?usp=sharing](https://colab.research.google.com/drive/1JW-DI1PSsVzO51IXYYv8DPyZ6eh4RWx5?usp=sharing)

[https://colab.research.google.com/drive/1rVMCEVLXGdVRK5rAPqz-](https://colab.research.google.com/drive/1rVMCEVLXGdVRK5rAPqz-TrNKAQVSjUJC?usp=sharing)

[TrNKAQVSjUJC?usp=sharing](https://colab.research.google.com/drive/1rVMCEVLXGdVRK5rAPqz-TrNKAQVSjUJC?usp=sharing)

https://colab.research.google.com/github/goodboychan/goodboychan.github.io/blob/main/_notebooks/2020-08-02-02-Image-restoration-Noise-Segmentation-and-Contours.ipynb

<https://colab.research.google.com/github/tensorflow/compression/blob/master/models/hific/colab.ipynb>

<https://github.com/danielaczarref/Wavelet-Image-Processing/blob/master/main.ipynb>

https://github.com/mahmoudnafifi/WB_sRGB/blob/master/WB_s

<https://medium.com/@koushikc2000/2d-discrete-wavelet-transformation-and-its-applications-in-digital-image-processing-using-matlab-1f5c68672de3>

<https://pyimagesearch.com/2021/02/15/automatic-color-correction-with-opencv-and-python/>

<https://pywavelets.readthedocs.io/en/latest/>