

# PETUNJUK PRAKTIKUM

---

## PENGOLAHAN CITRA DAN POLA

INSTITUT BISNIS DAN TEKNOLOGI INDONESIA  
FAKULTAS TEKNOLOGI DAN INFORMATIKA  
PROGRAM STUDI TEKNIK INFORMATIKA

# DAFTAR ISI

<b>Petunjuk Praktikum #1</b>	
Pengolahan Citra dan Pola	1
Instalasi Python	6
<b>Petunjuk Praktikum #2</b>	
Dasar Pengolahan Citra Digital	15
Latihan Code Python Sederhana	23
<b>Petunjuk Praktikum #3</b>	
Operasi Dasar Pengolahan Citra (Operasi Aritmatika)	33
<b>Petunjuk Praktikum #4</b>	
Operasi Dasar Pengolahan Citra (Operasi Geometri)	47
<b>Petunjuk Praktikum #5</b>	
Operasi Pengolahan Citra: Operasi Titik Bagian 1	47
<b>Petunjuk Praktikum #6</b>	
Operasi Pengolahan Citra: Operasi Titik Bagian 2	74
<b>Petunjuk Praktikum #7</b>	
Operasi Pengolahan Citra: Operasi Spasial	88
<b>Petunjuk Praktikum #8</b>	
Image Morphology	105
<b>Petunjuk Praktikum #9</b>	
Image Morphology (Bagian 2)	120
<b>Petunjuk Praktikum #10</b>	
Image Segmentasi	132
<b>Petunjuk Praktikum #11</b>	
Image Segmentasi	146

## **PETUNJUK PRAKTIKUM #1**

**PENGOLAHAN CITRA DAN POLA,  
INSTALASI PYTHON**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGOLAHAN CITRA DAN POLA</h1>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 1 dari 183

**A. Capaian Pembelajaran Mata Kuliah**

1. Mampu menjunjung tinggi kebenaran, kebaikan dan keindahan;
2. Menunjukkan sikap jujur, luhur dan setia dalam menjalankan profesi dan pekerjaanya.

**B. Sub Capaian Pembelajaran Mata Kuliah**

1. Mahasiswa mampu menjelaskan konsep dasar citra digital, representasi citra digital dan penerapannya dalam berbagai bidang.

**C. Tujuan**

1. Mahasiswa mampu menjelaskan konsep dasar citra digital;
2. Mahasiswa memahami penerapan pengolahan citra digital dan pola dalam berbagai bidang;
3. Mahasiswa mampu melakukan instalasi python dan cara penggunaannya.

**D. Dasar Teori**

**Pengolahan Citra Digital**

Pengolahan citra (image processing) merupakan suatu sistem dimana proses dilakukan dengan masukan berupa citra (image) (Basuki, 2007) dan hasilnya juga berupa citra (image). Sesuai dengan perkembangan komputer itu sendiri, pengolahan citra mempunyai dua tujuan utama. Memperbaiki kualitas citra, merupakan tujuan awal pengolahan citra dimana citra yang dihasilkan dapat menampilkan informasi secara jelas atau dengan kata lain manusia dapat melihat informasi secara jelas atau dapat menginterpretasikan citra yang ada. Selanjutnya dengan pengolahan citra dapat juga mengekstraksi informasi ciri yang menonjol pada suatu citra di mana hasilnya adalah informasi citra dimana manusia mendapatkan informasi ciri dari citra secara numerik atau dengan kata lain komputer (mesin) melakukan interpretasi terhadap informasi yang ada pada citra melalui besaran-besaran data yang dapat dibedakan secara jelas (besaran-besaran ini berupa besaran numerik).

Pengolahan citra adalah pemrosesan citra, khususnya menggunakan computer, menjadi citra yang kualitasnya lebih baik. Pengolahan citra dilakukan, jika perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau menonjolkan beberapa aspek informasi yang terkandung di dalam citra.

1. Elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur.
2. Sebagian citra perlu digabung dengan bagian citra yang lain.

Ada beberapa bidang studi yang terkait dengan data citra tetapi mempunyai tujuan berbeda, yaitu:

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGOLAHAN CITRA DAN POLA</b></p>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 2 dari 183

- a. Grafika Komputer → menghasilkan citra Data Deskriptif → Grafika Komputer → Citra
- b. Pengolahan Citra → memperbaiki citra Citra → Pengolahan Citra → Citra
- c. Pengenalan Pola → mengenali suatu objek Citra → Pengenalan Pola → Deskripsi Objek

Operasi-operasi pada pengolahan citra diterapkan pada citra bila :

1. Perbaikan atau memodifikasi citra dilakukan untuk meningkatkan kualitas penampakan citra/menonjolkan beberapa aspek informasi yang terkandung dalam citra (image enhancement). Contoh : perbaikan kontras gelap/terang, perbaikan tepian objek, penajaman, pemberian warna semu, dll
2. Adanya cacat pada citra sehingga perlu dihilangkan/diminimumkan (image restoration). Contoh : penghilangan kesamaran (debluring) citra tampak kabur karena pengaturan fokus lensa tidak tepat / kamera goyang, penghilangan noise
3. Elemen dalam citra perlu dikelompokkan, dicocokan atau diukur (image segmentation). Operasi ini berkaitan erat dengan pengenalan pola.
4. Diperlukannya ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu dalam pengidentifikasi objek (image analysis).
5. Proses segementasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh : pendektsian tepi objek. Sebagian citra perlu digabung dengan bagian citra yang lain (image reconstruction). Contoh : beberapa foto rontgen digunakan untuk membentuk ulang gambar organ tubuh. Citra perlu dimampatkan (image compression); suatu file citra berbentuk BMP berukuran 258 KB dimampatkan dengan metode JPEG menjadi berukuran 49 KB.
6. Menyembunyikan data rahasia (berupa teks/citra) pada citra sehingga keberadaan data rahasia tersebut tidak diketahui orang (steganografi & watermarking)

### Metodologi Pengolahan Citra

Menurut (Hermawati 2013), metode-metode dalam pengolahan citra digital dibagi menjadi 8, yaitu:

1. Pembentukan Citra (Data Acquisition): Menentukan data yang diperlukan dan memilih metode perekaman citra digital.
2. Pengolahan Citra Tingkat Awal (Image Preprocessing): Meningkatkan kontras, menghilangkan gangguan geometrik/ radiometrik, menentukan bagian citra yang akan diobservasi.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGOLAHAN CITRA DAN POLA</b></p>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 3 dari 183

3. Segmentasi Citra (Image Segmentation) dan deteksi sisi (Edge Detection): Melakukan partisi citra menjadi wilayah-wilayah obyek (internal properties) atau menentukan garis batas wilayah obyek (external shape characteristics).
4. Seleksi dan Ekstraksi Ciri (Feature Extraction and Selection): Seleksi ciri memilih informasi kuantitatif dari ciri yang ada, yang dapat membedakan kelas-kelas obyek secara baik. Ekstraksi ciri mengukur besaran kuantitatif ciri setiap piksel.
5. Representasi dan Deskripsi: Suatu wilayah dapat direpresentasi sebagai suatu list titik-titik koordinat dalam loop yang tertutup, dengan deskripsi luasan/parameternya.
6. Pengenalan Pola (Pattern Recognition): Memberikan label kategori obyek pada setiap piksel citra berdasarkan informasi yang diberikan oleh deskriptor atau ciri piksel bersangkutan (pewilayahan jaringan keras dan pewilayahan berbagai jaringan lunak pada citra biomedik).
7. Interpretasi Citra (Image Interpretation): Memberikan arti pada obyek yang sudah berhasil dikenali (dari citra klasifikasi biomedik dapat dilihat adanya penyakit tumor).
8. Penyusunan Basis Pengetahuan: Basis pengetahuan ini digunakan sebagai referensi pada proses template matching/ object recognition.

### Operasi-Operasi dalam Citra

Operasi-operasi yang dilakukan dalam pengolahan citra banyak ragamnya. Namun, secara umum, operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis sebagai berikut:

1. Perbaikan kualitas citra (image enhancement).  
Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Dengan operasi ini, ciri-ciri khusus yang terdapat di dalam citra lebih ditonjolkan. Contoh-contoh operasi perbaikan citra: perbaikan kontras gelap/terang, perbaikan tepian objek (edge enhancement), penajaman (sharpening), pemberian warna semu (pseudocoloring), penapisan derau (noise filtering).
2. Pemugaran citra (image restoration).  
Operasi ini bertujuan menghilangkan/memminimumkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra. Bedanya, pada pemugaran citra penyebab degradasi gambar diketahui. Contoh-contoh operasi pemugaran citra: penghilangan kesamaran (deblurring), penghilangan derau (noise).
3. Pemampatan citra (image compression).

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGOLAHAN CITRA DAN POLA</b></p>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 4 dari 183

Jenis operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal penting yang harus diperhatikan dalam pemampatan adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus. Contoh metode pemampatan citra adalah metode JPEG.

4. Segmentasi citra (image segmentation).

Jenis operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini berkaitan erat dengan pengenalan pola.

5. Pengorakan citra (image analysis).

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik pengorakan citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh-contoh operasi pengorakan citra: pendekripsi tepi objek (edge detection), ekstraksi batas (boundary), representasi daerah (region).

6. Rekonstruksi citra (image reconstruction).

Jenis operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis. Misalnya beberapa foto rontgen dengan sinar X digunakan untuk membentuk ulang gambar organ tubuh.

### Computer Vision dan Hubungannya dengan Pengolahan Citra Digital

Menurut (Munir 2019), terminologi lain yang berkaitan erat dengan pengolahan citra adalah computer vision atau machine vision. Pada hakikatnya, computer vision mencoba meniru cara kerja sistem visual manusia (human vision). Human vision sesungguhnya sangat kompleks. Manusia melihat objek dengan indera penglihatan (mata), lalu citra objek diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti objek apa yang tampak dalam pandangan matanya. Hasil interpretasi ini mungkin digunakan untuk pengambilan keputusan (misalnya menghindar kalau melihat mobil melaju di depan). Proses-proses di dalam computer vision dapat dibagi menjadi tiga aktivitas:

1. Memperoleh atau mengakuisisi citra digital.
2. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra (operasi-operasi pengolahan citra).
3. Menganalisis dan menginterpretasi citra dan menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya memandu robot, mengontrol peralatan, memantau proses manufaktur, dan lain-lain.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGOLAHAN CITRA DAN POLA</b></p>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 5 dari 183

Computer vision merupakan proses otomatis yang mengintegrasikan sejumlah besar proses untuk persepsi visual, seperti akuisisi citra, pengolahan citra, klasifikasi, pengenalan (recognition), dan membuat keputusan. Pengolahan citra merupakan proses awal (preprocessing) pada computer vision, sedangkan pengenalan pola merupakan proses untuk menginterpretasi citra. Teknik-teknik di dalam pengenalan pola memainkan peranan penting dalam computer vision untuk mengenali objek.

Jika dihubungkan dengan grafika komputer, maka computer vision merupakan kebalikannya. Grafika komputer membentuk (sintesis) citra, sedangkan computer vision mengoraknya (analisis). Pada masa awal kedua bidang ini, tidak ada hubungan antara keduanya, tetapi beberapa tahun belakangan kedua bidang tersebut berkembang semakin dekat. Computer vision menggunakan representasi kurva dan permukaan dan beberapa teknik lain dari grafika komputer, sedangkan grafika komputer menggunakan teknik-teknik di dalam computer vision untuk memuat citra realistik (virtual reality).

### **Pengenalan Pola**

Pengenalan pola adalah mengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (komputer). Tujuan pengelompokan adalah untuk mengenali suatu objek di dalam citra. Manusia bisa mengenali objek yang dilihatnya karena otak manusia telah belajar mengklasifikasi objek-objek di alam sehingga mampu membedakan suatu objek dengan objek lainnya. Kemampuan sistem visual manusia yang dicoba ditiru oleh mesin. Komputer menerima masukan berupa citra objek yang akan diidentifikasi, memproses citra tersebut dan memberikan keluaran berupa informasi/deskripsi objek di dalam citra.

### **Aplikasi Pengolahan Citra dan Pengenalan Pola**

Saat ini aplikasi pengolahan citra sudah banyak diterapkan di berbagai bidang seperti:

1. Photo Editing
2. Otomasi perkantoran
3. Media
4. Biometrik
5. Kedokteran (Medis)
6. Entertainment

Berikut aplikasi Pengolahan Citra dan Pengenalan Pola:

- Bidang Perdagangan (pembacaan bar code)
- Bidang Militer (mengidentifikasi jenis pesawat musuh)
- Kedokteran ( Rekonstruksi foto janin bayi hasil USG)

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGOLAHAN CITRA DAN POLA</b></p>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 6 dari 183

- Biologi (pengenalan kromosom)
- Komunikasi Data dan Hiburan (Pemampatan video)
- Geografi dan Geologi (mengenali jenis batu-batuan melalui foto udara)
- Hukum (pengenalan sidikjari)
- Content Based Image Retrieval
- Pengenalan Wajah
- Tracking Wajah secara Real Time
- Pengenalan Tulisan dan Tanda Tangan Untuk Cek Bank
- Mesin Absensi Dengan Sidik Jari
- Deteksi dan Pengenalan Rambu-Rambu Lalu-Lintas
- Deteksi Gerakan Badan Untuk Kendali Game
- Kendali Game Dengan Gerakan Mata
- Filter Gambar Porno

### Mengenal Python

Python adalah bahasa pemrograman yang berdaya guna dan mudah dipelajari. Python memiliki struktur data tingkat tinggi yang efisien dengan pendekatan sederhana tetapi efektif untuk pemrograman berorientasi objek. Sintaksis Python yang elegan dengan pengetikan dinamis dan sifat tafsirannya alami, menjadikannya bahasa yang ideal untuk skrip dan pengembangan aplikasi yang cepat di banyak bidang dalam berbagai platform.

Phyton juga memiliki kinerja tinggi untuk komputasi berbagai masalah teknik yang mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu modul-modul yang sangat mudah untuk pakai. Penggunaan Matlab meliputi bidang-bidang:

- Matematika dan Komputasi
- Pembentukan Algoritm
- Akusisi Data
- Pemodelan, simulasi, dan pembuatan prototipe
- Analisa data, explorasi, dan visualisasi
- Grafika dan Rekayasa

Sistem Python bersifat interaktif dan memiliki elemen data dalam larik/array, sehingga pengguna tidak lagi dipusingkan dengan masalah dimensi. Hal ini memungkinkan pemecahan banyak masalah teknis yang terkait dengan komputasi, khususnya yang berhubungan dengan matrix dan formulasi vektor, yang mana masalah tersebut merupakan momok apabila kita harus menyelesaiakannya dengan menggunakan bahasa level rendah seperti Pascal, C dan Visual Basic.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGOLAHAN CITRA DAN POLA</b></p>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 7 dari 183

Penerjemah dan pustaka standar Python tersedia secara bebas dalam bentuk kode asal atau biner untuk semua platform dan dapat didistribusikan secara bebas yang dapat diunduh dari situs Web Python

Python dirancang agar mudah dibaca, karenanya sering menggunakan kata kunci bahasa Inggris yang memiliki konstruksi sintaksis yang lebih sedikit daripada bahasa lain. Beberapa keunggulan Python antara lain:

Python is Interpreted - Python diproses pada saat runtime oleh penerjemah, sehingga tidak perlu mengkompilasi program sebelum menjalankannya. Ini mirip dengan PERL dan PHP.

- Python is Interactive – Pengguna sebenarnya bisa duduk di prompt Python dan berinteraksi dengan penerjemah secara langsung saat menulis program.
- Python is Object-Oriented - Python mendukung gaya atau teknik pemrograman Berorientasi Objek yang merangkum kode di dalam objek.
- Python is a Beginner's Language - Python adalah bahasa yang bagus untuk pemrogram tingkat pemula dan mendukung pengembangan berbagai aplikasi mulai dari pemrosesan teks sederhana hingga peramban www hingga permainan/games.

Python tersedia di berbagai platform dari windows, Linux hingga Mac OS X. Untuk menggunakan Python perlu mengerti cara mengatur lingkungan Python seperti yang dijelaskan bagian praktikum.

Python merupakan salah satu bahasa pemrograman yang paling banyak digunakan di dunia. Python merupakan bahasa yang dikembangkan untuk keperluan umum dan dilengkapi dengan berbagai package atau library yang mendukung banyak fungsi yang spesifik, seperti *numpy* (pengolahan matriks seperti Matlab), *scipy* (pengolahan data statistik), *bokeh* (visualisasi dan data analitik), *flask* (framework untuk web berbasis Python), dan lain sebagainya. OpenCV sendiri merupakan salah satu *library* yang juga dibuat dalam bahasa Python sehingga memudahkan penggunaannya untuk keperluan pemrosesan citra dalam lingkungan Python. Saat ini terdapat dua versi besar Python yang beredar, yaitu versi 2.x dan versi 3.x.

OpenCV merupakan sebuah framework yang banyak digunakan untuk keperluan pemrosesan image dan video. OpenCV bersifat gratis dan tersedia untuk berbagai bahasa pemrograman, seperti C++, Python, Java dan lain-lain pada platform Windows, Linux, iOS, Android dan seterusnya. OpenCV dapat digunakan untuk melakukan berbagai operasi terhadap citra dan video dengan berbagai algoritma terbaru serta banyak digunakan pada berbagai sistem pengolahan citra yang populer.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGOLAHAN CITRA DAN POLA</b>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 8 dari 183

#### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

#### F. Keselamatan Kerja

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporan ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik
9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

#### G. Lembar Kerja

Pada kegiatan praktek Pengolahan Citra dan Pola ini, mahasiswa akan melakukan praktikum dari mulai mendesain dan mengimplementasikan algoritma pengolahan citra digital menggunakan bahasa pemrograman Python dengan didukung oleh library computer vision ‘opencv2’ dan library lainnya yang mendukung.

Untuk menggunakan bahasa python diperlukan instalasi python dan text editor. Beberapa text editor yang bisa digunakan adalah visual studi code (vs code), pycharm, jupyter atau spider atau IDE lainnya yang mendukung bahasa python.

Berikut ini adalah proses persiapan kegiatan praktikum mulai dari proses instalasi Python, open-source software untuk editor pemrograman menggunakan VS Code dan modul lib lainnya.

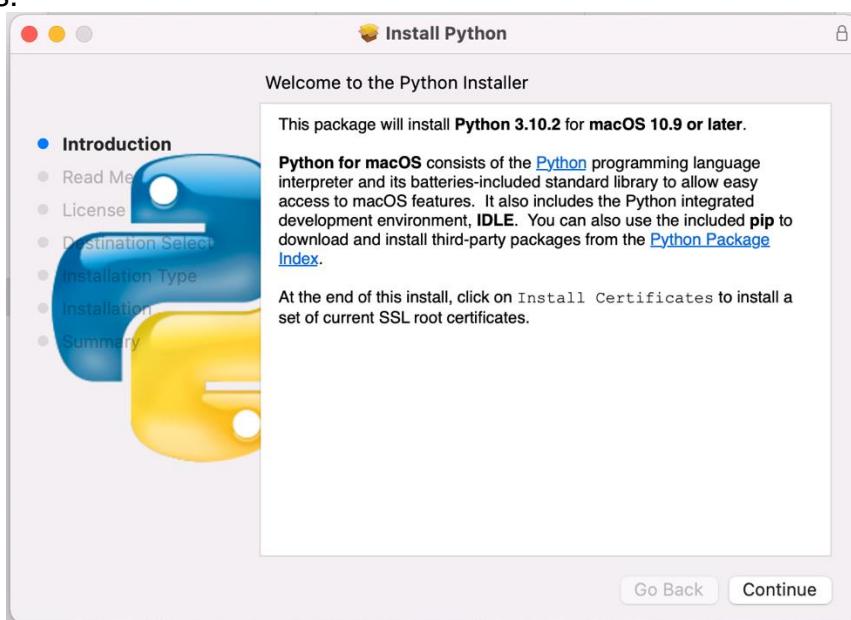
#### Kegiatan Praktikum 1: Instalasi Python

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGOLAHAN CITRA DAN POLA</h1>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 9 dari 183

Tahapan pertama yang dilakukan adalah menginstall Python terlebih dahulu, berikut langkah-langkah yang perlu dilakukan:

1. Download python pada link <https://www.python.org/downloads/>. Instalasi Python tersedia pada OS windows, mac OS dan Linux. Silahkan download installer sesuaikan dengan OS yang dimiliki
2. Setelah selesai download, klik dua kali pada installer. Berikut tampilan pada mac os.



3. Untuk windows agar lebih memudahkan pencarian silahkan lakukan customized untuk menentukan lokasi direktori instalasi. Sebaiknya instal pada direktori C dan buat folder baru dengan nama 'PythonXX' (XX menandakan versi python)
4. Ikuti perintah instalasi hingga akhir
5. Untuk pengguna windows, pastikan path sudah ditambahkan pada system variabel

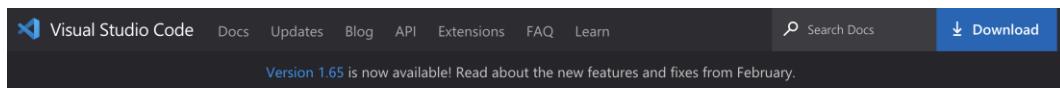
### Kegiatan Praktikum 2: Instalasi Code Editor

Code editor yang digunakan adalah vs code, berikut tahapan untuk download.

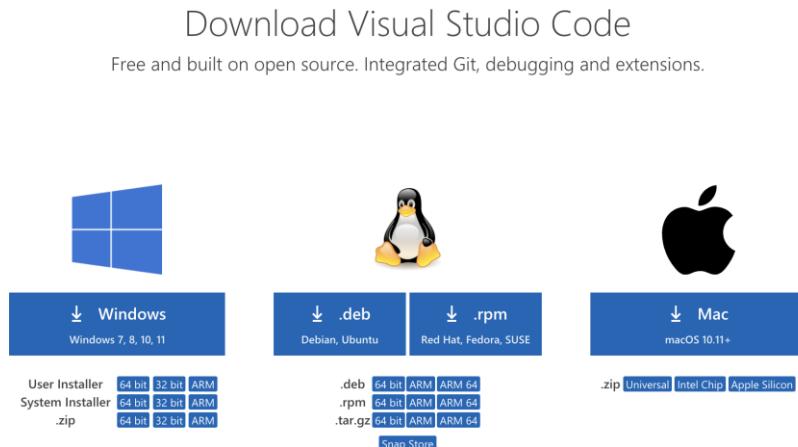
1. Download installer vs code di <https://code.visualstudio.com/download>

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

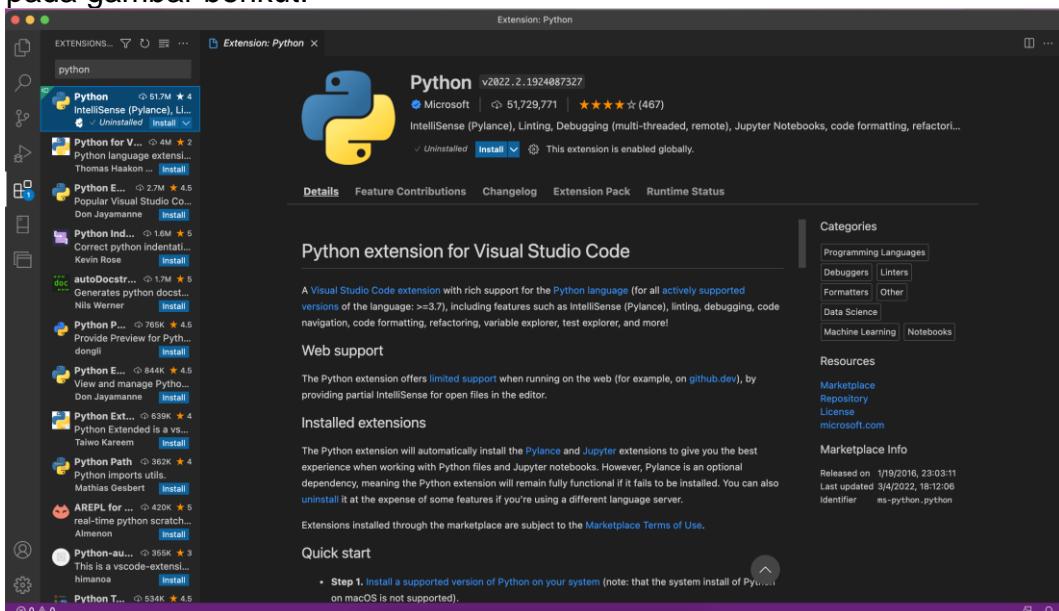
 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p><b>Fakultas Teknologi dan Informatika</b></p> <p><b>Program Studi Teknik Informatika</b></p> <h1 style="margin: 0;">PENGOLAHAN CITRA DAN POLA</h1>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 10 dari 183



The screenshot shows the Visual Studio Code download page. At the top, there's a navigation bar with links to Docs, Updates, Blog, API, Extensions, FAQ, and Learn. To the right is a search bar labeled 'Search Docs' and a blue 'Download' button. Below the navigation is a message: 'Version 1.65 is now available! Read about the new features and fixes from February.' The main content area has a heading 'Download Visual Studio Code' and a sub-heading 'Free and built on open source. Integrated Git, debugging and extensions.' Below this are download links for Windows, Linux (.deb and .rpm), and Mac OS. Each link includes options for User Installer, System Installer, and .zip files, with ARM support indicated.



2. Pada halaman download sesuaikan versi OS yang digunakan
3. Setelah selesai download, klik dua kali pada installer
4. Ikuti petunjuk instalasi sampai selesai
5. Setelah sukses terinstal, buka aplikasi vs code dan buka menu ‘Extensions’ pada sidebar kiri.
6. Pada kolom pencarian ketikkan kata kunci ‘Python’, extensions akan terlihat pada gambar berikut.



This screenshot shows the Visual Studio Code Extension Marketplace. A search bar at the top is set to 'Extension: Python'. On the left, a sidebar lists several Python-related extensions, including 'Python' by Microsoft, 'Python for V...' by Thomas Haakon, 'Python E...' by Popular Visual Studio Co., 'Python Ind...' by Kevin Rose, 'autoDocstr...' by Nils Werner, 'Python P...' by Don Jayamane, 'Python Ext...' by Taivo Karem, 'Python Path' by Mathias Gesbert, and 'AREPL for ...' by Almenon. The central panel displays the details for the 'Python' extension by Microsoft, version v2022.2.1924087327. It shows a Python logo icon, the extension name, developer information, and a brief description: 'IntelliSense (Pylance), Linting, Debugging (multi-threaded, remote), Jupyter Notebooks, code formatting, refactoring...'. Below this are tabs for Details, Feature Contributions, Changelog, Extension Pack, and Runtime Status. To the right, there are sections for Categories (Programming Languages, Debuggers, Linters, Formatters, Other, Data Science, Machine Learning, Notebooks), Resources (Marketplace, Repository, License, microsoft.com), and Marketplace Info (Released on 1/19/2016, Last updated 3/4/2022, Identifier ms-python.python).

## PENGOLAHAN CITRA DAN POLA 10

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

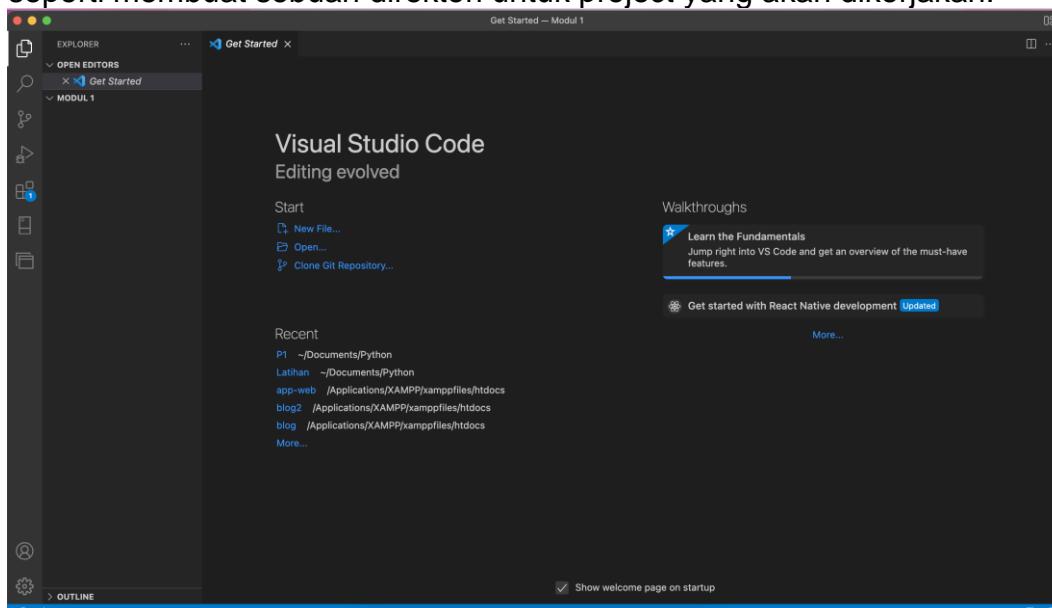
 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p><b>Fakultas Teknologi dan Informatika</b></p> <p><b>Program Studi Teknik Informatika</b></p> <h1 style="margin: 0;">PENGOLAHAN CITRA DAN POLA</h1>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 11 dari 183

7. Klik instal pada extension dan tunggu beberapa saat sampai selesai, jika memungkinkan diminta untuk restart vs code.
8. Untuk menginstal extensions lainnya seperti menjalankan 'jupyter' pada vs code bisa dilakukan dengan cara yang sama.

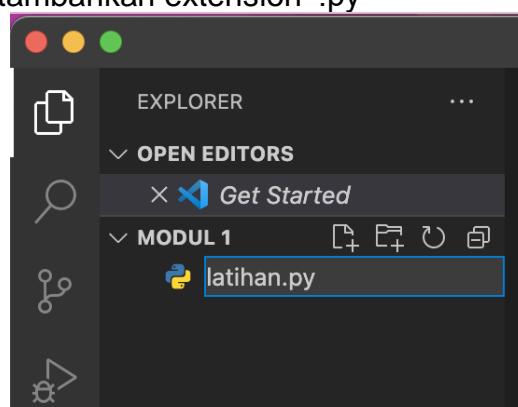
### Kegiatan Praktikum 3: Membuat file python

Untuk menjalankan program python berikut langkah-langkahnya:

1. Buka aplikasi vs code
2. Pada menu sidebar kiri, pilih menu 'explorer' dan pilih 'Open Folder'. Tentukan letak file dan sebaiknya dibuatkan dalam satu folder. Open folder seperti membuat sebuah direktori untuk project yang akan dikerjakan.



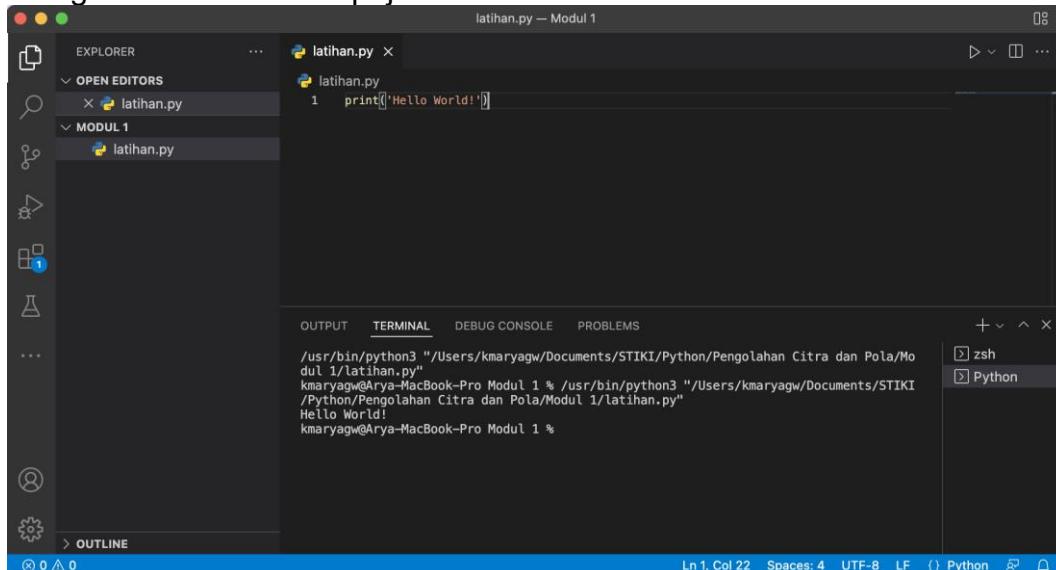
3. Arahkan cursor pada nama folder, lalu pilih 'new file'. Ketikkan nama file dan pada akhir tambahkan extensiion '.py'



Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p><b>Fakultas Teknologi dan Informatika</b></p> <p><b>Program Studi Teknik Informatika</b></p> <h1 style="margin: 0;">PENGOLAHAN CITRA DAN POLA</h1>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 12 dari 183

4. Ketikkan code python sederhana dan jalankan dengan menggunakan mengklik tombol ‘run’ di pojok kanan atas.



```
latihan.py — Modul 1
EXPLORER OPEN EDITORS latihan.py
MODUL 1 latihan.py
latihan.py
1 print('Hello World!')

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS
/usr/bin/python3 "/Users/kmaryagw/Documents/STIKI/Python/Pengolahan Citra dan Pola/Modul 1/latihan.py"
kmaryagw@Arya-MacBook-Pro Modul 1 % /usr/bin/python3 "/Users/kmaryagw/Documents/STIKI/Python/Pengolahan Citra dan Pola/Modul 1/latihan.py"
Hello World!
kmaryagw@Arya-MacBook-Pro Modul 1 %

Ln 1, Col 22 Spaces: 4 UTF-8 LF () Python ⌂ ⌂
```

5. Hasil akan ditunjukkan pada terminal  
 6. Sukses menjalankan program python

#### Kegiatan Praktikum 4: Membuat Virtual Environment

Secara default, setiap bahasa Python yang Anda instal berjalan di lingkungan global, yang tidak spesifik untuk satu proyek. Misalnya, jika Anda baru saja menjalankan python (Windows) atau python3 (macOS/Linux) pada command prompt baru, Anda menjalankannya di lingkungan global. Dengan demikian, paket apa pun yang Anda instal atau copot akan memengaruhi lingkungan global dan semua program yang dijalankan dalam konteks global. Meskipun bekerja di lingkungan global adalah cara mudah untuk memulai, seiring waktu, akan menjadi berantakan dengan banyak paket berbeda yang telah Anda instal untuk berbagai proyek. Kekacauan seperti itu mempersulit pengujian aplikasi secara menyeluruh terhadap kumpulan paket tertentu dengan versi yang diketahui, yang persis seperti lingkungan yang Anda siapkan di server build atau server web.

Untuk alasan ini, pengembang sering membuat lingkungan virtual untuk sebuah proyek. Virtual Environment adalah subfolder dalam proyek yang berisi salinan penerjemah tertentu. Saat mengaktifkan lingkungan virtual, paket apa pun yang terinstal hanya diinstal di subfolder lingkungan tersebut. Ketika kemudian menjalankan program Python di dalam lingkungan itu, yang perlu tahu bahwa itu hanya menjalankan paket-paket tertentu. Jika kita tidak menggunakan lingkungan virtual, dan memiliki beberapa versi Python yang

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p><b>Fakultas Teknologi dan Informatika</b></p> <p><b>Program Studi Teknik Informatika</b></p> <h1 style="margin: 0;">PENGOLAHAN CITRA DAN POLA</h1>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 13 dari 183

diinstal dan disetel dalam variabel lingkungan, mungkin perlu menentukan bahasa Python untuk digunakan di terminal untuk menginstal library ke lingkungan global.

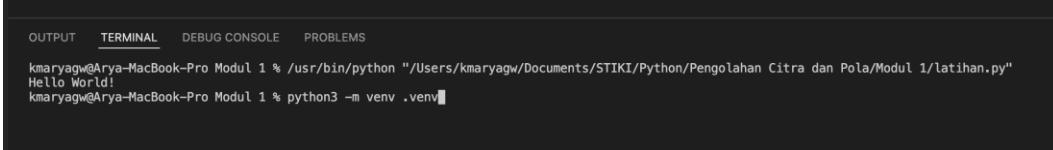
Untuk membuat virtual environment gunakan perintah berikut:

```
# macOS/Linux
# You may need to run sudo apt-get install python3-venv first
python3 -m venv .venv

# Windows
# You can also use py -3 -m venv .venv
python -m venv .venv
```

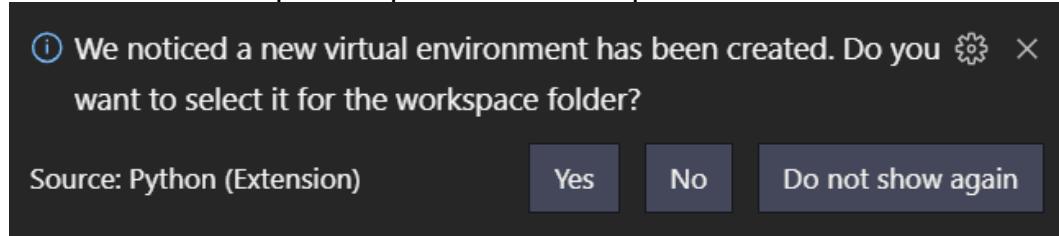
Sumber: <https://code.visualstudio.com/docs/python/python-tutorial>

Perintah dijalankan pada terminal:

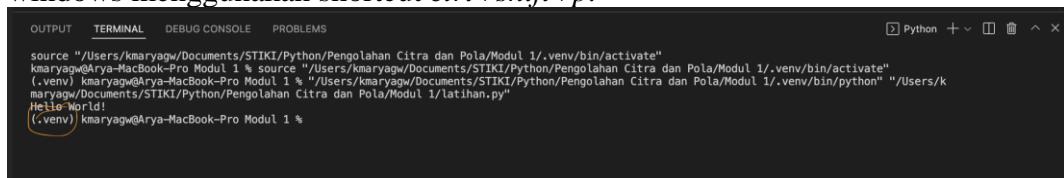


```
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS
kmaryagw@Arya-MacBook-Pro Modul 1 % /usr/bin/python "/Users/kmaryagw/Documents/STIKI/Python/Pengolahan Citra dan Pola/Modul 1/latihan.py"
Hello World!
kmaryagw@Arya-MacBook-Pro Modul 1 % python3 -m venv .venv
```

Setelah sukses maka pada explorer akan terdapat notifikasi.



Untuk mengaktifkan interpreter pada direktori .venv yang perlu dilakukan adalah menutup terminal sebelumnya dan jalankan pada terminal baru. Penanda bahwa virtual environment telah berjalan dapat dilihat pada bagian kiri. Jika belum terlihat (.venv) maka interpreter yang digunakan masih belum sesuai. Untuk melakukan *select interpreter* dapat menggunakan shortcut *cmd+shift+p* untuk mac os. Untuk pengguna windows menggunakan shortcut *ctrl+shift+p*.



```
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS
source "/Users/kmaryagw/Documents/STIKI/Python/Pengolahan Citra dan Pola/Modul 1/.venv/bin/activate"
kmaryagw@Arya-MacBook-Pro Modul 1 % source "/Users/kmaryagw/Documents/STIKI/Python/Pengolahan Citra dan Pola/Modul 1/.venv/bin/activate"
(.venv) kmaryagw@Arya-MacBook-Pro Modul 1 % "/Users/kmaryagw/Documents/STIKI/Python/Pengolahan Citra dan Pola/Modul 1/.venv/bin/python" "/Users/kmaryagw/Documents/STIKI/Python/Pengolahan Citra dan Pola/Modul 1/latihan.py"
Hello World!
(.venv) kmaryagw@Arya-MacBook-Pro Modul 1 %
```

Untuk mengaktifkan melalui perintah dapat mengikuti perintah berikut.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1>PENGOLAHAN CITRA DAN POLA</h1>		
Pertemuan : 1	<b>Pengolahan Citra dan Pola, Instalasi Python</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 1/03/2022	Hal 14 dari 183

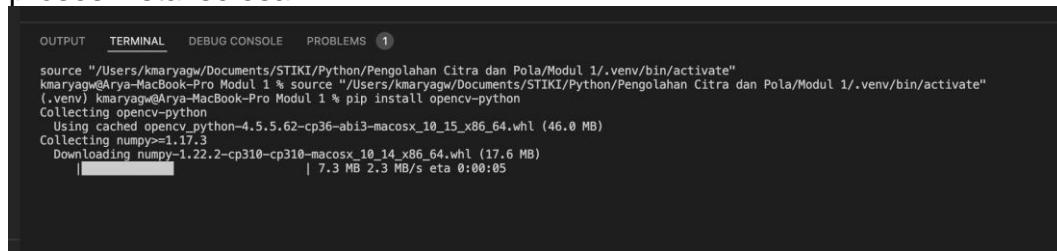
```
# For Windows
.venv\scripts\activate
```

```
# For macOS/Linux
source .venv/bin/activate
```

### Kegiatan Praktikum 5: Instalasi Library Tambahan

Dalam praktikum pengolahan citra dan pola akan menggunakan beberapa library tambahan seperti OpenCv atau numpy. Secara default library tersebut belum terinstal sehingga perlu melakukan instalasi terlebih dahulu. Untuk melakukan instalasi OpenCV berikut langkahnya:

1. Untuk menginstal library tambahan menggunakan perintah *pip install <nama library>*
2. Jalankan perintah *pip install opencv-python* pada terminal. Tunggu hingga proses instal selesai.



```
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 1
source "/Users/kmaryagw/Documents/STIKI/Python/Pengolahan Citra dan Pola/Modul 1/.venv/bin/activate"
kmaryagw@Arya-MacBook-Pro Modul 1 % source "/Users/kmaryagw/Documents/STIKI/Python/Pengolahan Citra dan Pola/Modul 1/.venv/bin/activate"
(.venv) kmaryagw@Arya-MacBook-Pro Modul 1 % pip install opencv-python
Collecting opencv-python
  Using cached opencv_python-4.5.5.62-cp36-ab13-macosx_10_15_x86_64.whl (46.0 MB)
Collecting numpy>=1.17.3
  Downloading numpy-1.22.2-cp310-cp310-macosx_10_14_x86_64.whl (17.6 MB)
|██████████| 7.3 MB 2.3 MB/s eta 0:00:05
```

3. Untuk menguji apakah sudah sukses terinstal maka silahkan jalankan code berikut

```
import cv2

print("Hello World")
```

4. Jalankan (run) dan jika terjadi error seperti "module cv2 not found" berarti proses instalasi masih belum berhasil. Silahkan untuk mengulang kembali dan pastikan terdapat koneksi internet saat melakukan instalasi.

### H. Tugas Praktikum

1. Cobalah masing-masing untuk melakukan instalasi python dan code editor (visual studio code)
2. Buatlah virtual environment dan instal library tambahan seperti OpenCV . Lalu jalankan sebuah code untuk menampilkan "Hello World".

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

## **PETUNJUK PRAKTIKUM #2**

**DASAR PENGOLAHAN CITRA; CODE PYTHON  
SEDERHANA; PENGGUNAAN OPENCV**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 16 dari 183

#### A. Capaian Pembelajaran Mata Kuliah

1. Mampu menjunjung tinggi kebenaran, kebaikan dan keindahan;
2. Menunjukkan sikap jujur, luhur dan setia dalam menjalankan profesi dan pekerjaanya.

#### B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu menjelaskan konsep dasar citra digital, representasi citra digital dan penerapannya dalam berbagai bidang.

#### C. Tujuan

Mahasiswa mampu menjelaskan definisi citra digital, menjelaskan proses akuisisi citra dan pembentukan, memberikan contoh citra digital

#### D. Dasar Teori

##### Definisi Dasar Pengolahan Citra

Citra adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses sampling. Gambar analog dibagi menjadi N baris dan M kolom sehingga menjadi gambar diskrit. Persilangan antara baris dan kolom tertentu disebut dengan piksel. Contohnya adalah gambar/titik diskrit pada baris n dan kolom m disebut dengan piksel [n,m].

Sampling adalah proses untuk menentukan warna pada piksel tertentu pada citra dari sebuah gambar yang kontinu. Pada proses sampling biasanya dicari warna rata-rata dari gambar analog yang kemudian dibulatkan. Proses sampling sering juga disebut proses digitisasi. Sampling merupakan bagian dari metodologi statistika.

Ada kalanya, dalam proses sampling, warna rata-rata yang didapat direlasikan ke level warna tertentu. Contohnya apabila dalam citra hanya terdapat 16 tingkatan warna abu-abu, maka nilai rata-rata yang didapat dari proses sampling harus diasosiasikan ke 16 tingkatan tersebut. Proses mengasosiasikan warna rata-rata dengan tingkatan warna tertentu disebut dengan kuantisasi.

Derau (noise) adalah gambar atau piksel yang mengganggu kualitas citra. Derau dapat disebabkan oleh gangguan fisis (optik) pada alat akuisisi maupun secara disengaja akibat proses pengolahan yang tidak sesuai. Contohnya adalah bintik hitam atau putih yang muncul secara acak yang tidak diinginkan di dalam citra. bintik acak ini disebut dengan derau salt & pepper.

##### Pengertian Citra Digital

Citra atau gambar didefinisikan sebagai fungsi dari dua variabel misalnya  $a$  ( $x,y$ ) dimana variabel  $a$  sendiri sebagai amplitudo (seperti : kecerahan), citra pada

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 17 dari 183

koordinat  $(x,y)$ . Citra digital  $a[m,n]$  merupakan citra dalam ruang diskrit dua dimensi yang berasal dari citra analog  $a(x,y)$  dalam sebuah ruang kontinyu melalui proses sampling yang biasa disebut sebagai digitalisasi.

Citra digital adalah citra  $f(x,y)$  yang telah didiskritkan pada koordinat spasial dan kecerahan. Citra digital direpresentasikan oleh array dua dimensi atau sekumpulan array dua dimensi dimana setiap array merepresentasikan satu kanal warna. Nilai kecerahan yang di digitalkan dinamakan nilai tingkat keabuan (Hidayatullah 2017).

### Tipe Citra

Ada beberapa tipe citra yang sering digunakan, di antaranya adalah:

1. Citra Biner

Di dalam citra biner, tiap-tiap piksel hanya membutuhkan 1 bit memori. Maka, dengan demikian setiap piksel hanya mempunyai 2 buah kemungkinan intensitas, yaitu 1 atau 0.

2. Citra Grayscale

Citra grayscale adalah matrik data yang nilainya mewakili intensitas setiap piksel berkisar antara 0 sampai dengan 255. Setiap piksel membutuhkan 8 bit memori.

3. Citra Warna

Citra warna adalah citra yang masing-masing piksel mempunyai 3 komponen warna yang spesifik, yaituk komponen merah, hijau, dan biru. Warna setiap piksel ditentukan oleh kombinasi dari intensitas warna merah, hijau, dan biru yang disimpan pada bidang warna di lokasi piksel. Format file grafis menyimpan citra warna sebagai citra 24 bit, yang berasal dari komponen merah, hijau, dan biru yang masing-masing 8 bit. Hal ini menyebabkan citra warna mempunyai 24 juta kemungkinan warna.

### Elemen Dasar Citra

Citra digital mengandung sejumlah elemen-elemen dasar. Elemen-elemen dasar inilah yang dimanipulasi dalam pengolahan citra. Elemen-elemen dasar yang penting diantaranya adalah sebagai berikut:

1. Kecerahan (brightness)

Kecerahan disebut juga sebagai intensitas cahaya. Kecerahan pada suatu titik (piksel) di dalam suatu citra sebenarnya adalah intensitas rata-rata dari suatu area yang melingkapinya.

2. Kontras (contrast)

Kontras menyatakan sebaran terang (lightness) dan gelap (darkness) dalam suatu citra. Citra dengan kontras rendah dicirikan oleh sebagian besar

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 18 dari 183

komposisi citranya adalah terang atau sebagian besar gelap. Citra dengan kontras yang baik, komposisi gelap dan terangnya tersebar secara merata.

### 3. Kontur (contour)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada pixel-pixel yang bertetangga. Adanya perubahan intensitas inilah maka tepi-tepi (edge) objek pada citra dapat dideteksi.

### 4. Warna (color)

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang yang berbeda-beda. Warna yang diterima oleh sistem visual manusia (mata) merupakan hasil kombinasi cahaya dengan panjang gelombang yang berbeda-beda. Kombinasi warna yang memberikan rentang warna yang paling lebar adalah red (R), green (G), dan blue (B).

### 5. Bentuk (shape)

Bentuk adalah properti intrinsik dari objek tiga dimensi. Bentuk merupakan properti intrinsik utama untuk sistem visual manusia karena manusia lebih sering menginterpretasikan suatu objek berdasarkan bentuknya daripada elemen lainnya.

### 6. Tekstur (texture)

Tekstur dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan pixelpixel yang bertetangga. Sehingga, tekstur tidak dapat didefinisikan untuk sebuah pixel. Tekstur merupakan karakteristik untuk menganalisa permukaan berbagai jenis citra objek.

## Pembentukan Citra

Suatu objek mendapatkan pencahayaan dari sumber cahaya tertentu. Posisi sumber cahaya menentukan juga hasil citra yang didapat. Ada dua teknik pembentukkan citra, yaitu:

### 1. Metode Sampling

Metode sampling merupakan metode untuk mendigitalisasi suatu fungsi citra yang diambil dari citra nyata yang direpresentasikan dari sebuah fungsi kontinyu  $f(x,y)$  yang kemudian diubah kedalam fungsi diskrit. Pada umumnya perangkat pembentuk citra, seperti kamera digital, piksel diwakili oleh rata-rata dari sinyal di area tertentu dari sebuah geometri dari elemen sensor pada perangkat pembentuk citra, jumlah piksel atau resolusi yang digunakan. Semakin bagus resolusinya, maka semakin presisi suatu citra yang didigitalisasikan.

### 2. Metode Kuantisasi

Metode kuantisasi dilakukan dengan menurunkan jumlah variasi tingkat keabuan dalam rangka untuk menyimpan dan memproses nilai-nilai

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 19 dari 183

gambar di komputer. Umumnya, nilai-nilai tersebut dikonversi atau diubah ke rentang nilai bilangan bulat (misalnya  $256=28$  atau  $4096=224$ ). Terkadang nilai di ubah ke dalam bentuk skala bilangan desimal seperti dalam aplikasi pencitraan medis. Konversi dilakukan dengan menggunakan pengubah sinyal analog ke digital, yang biasanya tertanam dalam sensor elektronik.

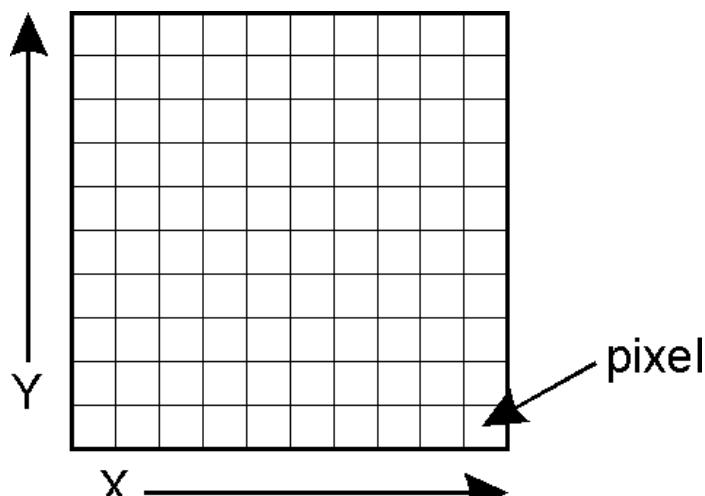
### Resolusi Citra

Resolusi citra merupakan tingkat detail suatu citra. Semakin tinggi resolusi citra maka akan semakin tinggi pula detail dari citra tersebut. Satuan dalam pengukuran resolusi citra dapat berupa ukuran fisik (jumlah garis per mm/ jumlah garis per inci). Cara pengukuran resolusi citra ada 3 jenis, yaitu:

1. Resolusi pixel

Resolusi pixel merupakan perhitungan jumlah pixel dalam sebuah citra digital. Sebuah dengan tinggi N pixel dan lebar M pixel berarti memiliki resolusi sebesar  $M \times N$ . Resolusi pixel akan memberikan dua buah angka integer yang secara berurutan akan mewakili jumlah pixel lebar dan jumlah pixel tinggi dari citra tersebut.

Pengertian lainnya dari resolusi pixel adalah merupakan hasil perkalian jumlah pixel lebar dan tingginya kemudian dibagi dengan 1 juta. Jenis resolusi pixel seperti ini sering kali dijumpai dalam kamera digital. Suatu citra yang memiliki lebar 2.048 pixel dan tinggi 1.536 pixel maka akan memiliki total pixel sebanyak  $2048 \times 1536 = 3145728$  pixel atau 3,1 mega pixel.

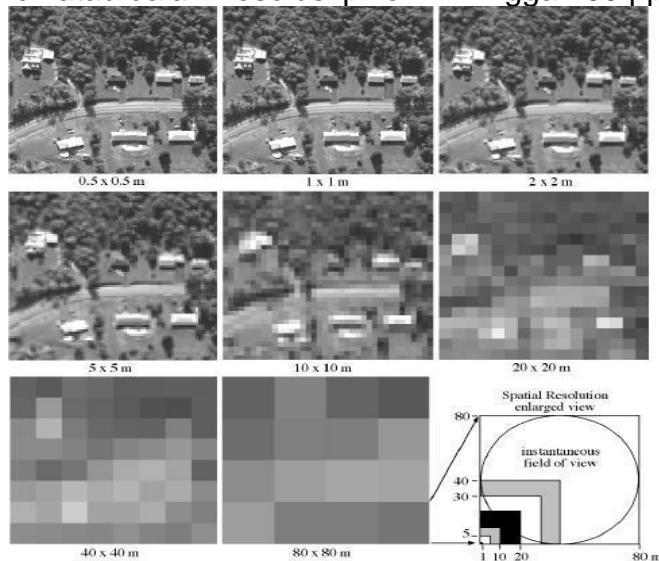


Gambar 2.1. Ilustrasi Resolusi Pixel  
Sumber: Google Images

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 20 dari 183

## 2. Resolusi Spasial

Resolusi spasial menunjukkan seberapa dekat jarak setiap garis pada citra. Jarak tersebut tergantung dari sistem yang menciptakan citra tersebut. Resolusi spasial menghasilkan jumlah pixel per satuan panjang. Resolusi spasial dari sebuah monitor komputer adalah 72 hingga 100 garis per inch atau dalam resolusi pixel 72 hingga 100 ppi.



**Gambar 2.2 Ilustrasi Resolusi Spasial**  
Sumber: Google Images

## 3. Resolusi Spektral

Sebuah citra digital membedakan intensitas kedalam beberapa spektrum. Citra multi spektrum akan memberikan panjang gelombang yang lebih baik yang akan digunakan untuk menampilkan warna (Putra 2010).

### Jenis Citra

Citra digital secara umum terbagi dalam tiga jenis, yaitu:

#### 1. Citra Berwarna

Citra berwarna adalah citra yang memiliki 3 buah kanal warna di dalamnya. Pada umumnya jenis citra ini terbentuk dari komponen warna merah/red (R), hijau/green (G), dan biru/blue (B) yang dimodelkan kedalam ruang warna RGB. RGB adalah standar yang digunakan untuk menampilkan citra berwarna pada layar televisi maupun komputer. Namun terdapat juga citra berwarna yang menggunakan ruang warna yang berbeda, seperti CMYK

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 21 dari 183

(Cyan, Magenta, Yellow, Black), HSV (Hue, Saturation, Value ), YCbCr (Luma, Chroma blue, Chroma red ) dan LAB ( $L^* a^* b^*$ ).

Pada masa ini, citra berwarna dikembangkan lagi dengan penambahan fitur transparansi. Citra berwarna dengan transparansi biasanya digunakan untuk menghilangkan bagian background dari objek dalam sebuah citra. Jenis citra ini terbentuk dari komponen RGB dan Alpha (A) yang dimodelkan kedalam ruang warna RGBA. Alpha atau transparansi akan disimpan kedalam sebuah kanal tambahan pada citra berwarna, sehingga total kanal yang digunakan sebanyak 4 kanal, yakni 3 kanal warna dan 1 kanal transparansi. Oleh karena itu, citra berwarna dengan transparansi memiliki ukuran file yang lebih besar. Citra berwarna dengan transparansi disebut juga sebagai 32-bit color image karena untuk setiap nilai pikselnya memerlukan penyimpanan sebesar 32-bit.

Tidak semua format file citra mendukung adanya transparansi. Sebagai contoh JPG tidak mendukung transparansi sehingga jika sebuah citra digital yang mengandung transparansi yang disimpan ke format JPG maka transparansinya akan hilang. Beberapa format yang mendukung transparansi antara lain PNG, TIFF dan GIF (Hidayatullah 2017).

## 2. Citra Grayscale

Citra grayscale adalah citra yang hanya memiliki 1 buah kanal sehingga yang ditampilkan hanyalah nilai intensitas atau dikenal juga dengan istilah derajat keabuan. Karena jenis citra ini hanya memiliki 1 kanal saja, maka citra grayscale memiliki tempat penyimpanan yang lebih hemat. Jenis citra ini disebut juga sebagai 8-bit image. Foto hitam putih maupun gambar yang ditampilkan oleh televisi hitam putih sebenarnya menggunakan citra grayscale, bukan dalam warna hitam dan warna putih. Secara teori ada beberapa cara dalam mengonversi citra berwarna RGB ke dalam citra grayscale. Cara paling mudah adalah dengan merata-ratakan semua nilai pixel RGB sesuai dengan persamaan berikut:

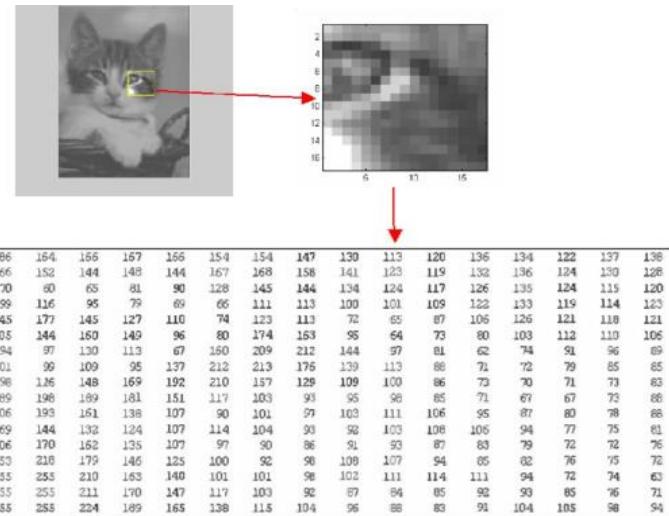
$$y = \frac{(R + G + B)}{3}$$

Namun nilai yang dihasilkan tidak terlalu bagus. Untuk mendapatkan hasil konversi yang lebih baik, dapat digunakan persamaan berikut:

$$y = 0.299R + 0.578G + 0.114B$$

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code</b> <b>Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 22 dari 183



Gambar 2.3 Representasi Citra Digital Keabuan (Gray Scale) dengan Matriks

### 3. Citra Biner

Citra biner atau citra hitam putih (black and white image) adalah citra yang hanya memiliki 2 kemungkinan nilai untuk setiap pixelnya, yaitu 0 atau 1. Nol (0) akan tampil sebagai warna hitam sedangkan 1 sebagai warna putih. Citra ini hanya membutuhkan 1-bit untuk menyimpan nilai pada setiap pixelnya. Jenis citra ini biasanya digunakan untuk proses masking atau proses segmentasi citra.

Untuk mendapatkan suatu citra biner, maka kita membutuhkan citra grayscale yang dilakukan thresholding terhadapnya berdasarkan nilai ambang batas (threshold) yang ditentukan. Jika nilai pixel pada citra grayscale melebihi atau menyamai nilai threshold, maka nilai piksel tersebut dikonversi menjadi 1. Namun jika nilai pixel kurang dari nilai threshold, maka nilai pixel dikonversi menjadi 0. Proses konversi dapat dituliskan menggunakan persamaan berikut:

$$y' = \begin{cases} 1, & y \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases}$$

Menemukan nilai threshold untuk menghasilkan citra biner yang baik bukanlah hal yang sederhana. Bisa saja kita memilih nilai threshold sebesar 128 yang merupakan nilai tengah dari 0 – 255. Namun belum tentu citra biner yang didapatkan merupakan hasil terbaik untuk semua citra. Threshold memiliki nilai yang dinamis bergantung terhadap nilai-nilai pixel dalam suatu citra.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 23 dari 183

#### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

#### F. Keselamatan Kerja

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporan ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik
9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

#### G. Lembar Kerja

Untuk mengerjakan praktikum pertemuan 2 pastikan sudah sukses dalam instalasi python pada pertemuan 1. Pastikan juga untuk menginstall library yang dibutuhkan seperti Open CV. Jika sudah sukses maka silahkan untuk membuat file baru dengan format .py pada folder yang sama dengan latihan pada pertemuan 1.

#### Kegiatan Praktikum: Latihan Python Sederhana

Pada file yang telah dibuat silahkan ketikkan baris perintah berikut.

```
# ini adalah komentar pada baris python
# mencetak ke layar menggunakan perintah 'print'
print("hello world!")

# mendefinisikan variabel
x = 5
y = 6
```

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 24 dari 183

```
# operasi aritmatika sederhana
hasil = x*y+2*x-2*y

# mencetak hasil ke keluaran
print('hasil dari operasi adalah', hasil)

# variabel juga dapat didefinisikan sekaligus, dipisahkan oleh tanda koma
var1, var2 = 23.5, "ini adalah isi var2"

print("isi var1: ", var1, "\nnisi var2: ", var2)
```

Jalankan dengan menggunakan perintah run atau klik kanan dan pilih “Run Python File in Terminal”.

### Tipe Data

Pada Python, tipe data tidak perlu didefinisikan secara explisit. Python akan mengenali secara otomatis tipe data tiap kali sebuah variabel didefinisikan. Apabila operasi aritmatika dilakukan pada dua variabel dengan tipe data yang tidak sesuai, Python akan memunculkan error. Cell-cell berikut menunjukkan beberapa jenis pendefinisian tipe data pada Python.

```
# Tipe data pada Python didefinisikan pada saat variabel dibuat
x = 5          # int
y = 6.5        # float
z = "bahasa ular"  # str

hasil = x + y    # penjumlahan antara int dan float
print(hasil)      # mencetak hasil
print(type(hasil)) # mencetak tipe data hasil
```

Selanjutnya ketikkan perintah berikut untuk mengetahui error jika tipe data tidak sesuai.

```
# cell berikut hanya akan bisa dijalankan
# apabila cell di atas sudah dijalankan sebelumnya

hasil = x + z  # akan muncul error akibat penjumlahan tipe data yang tidak sesuai
```

Hasil run pada terminal.

```
Traceback (most recent call last):
File "/Users/kmaryagw/Documents/STIKI/Python/Pengolahan Citra dan Pola/Modul 1/latihan2.py", line
34, in <module>
    hasil = x + z  # akan muncul error akibat penjumlahan tipe data yang tidak sesuai
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

### Tipe Data Lainnya

Seperti pada bahasa pemrograman lainnya pada python juga ada tipe data seperti array, dictionary, bolean dan lainnya. Silahkan untuk mencoba perintah berikut.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 25 dari 183

```
# Beberapa tipe data lain pada Python
# tipe data boolean
trueOrFalse = True
print(trueOrFalse)

# tipe data array/list
ar = ['blue', 'green', 'red']
print(ar[0]) # array pada Python dimulai dari urutan nol (0)

# tipe data dictionary
dic = {'nama buah': 'durian',
       'nama ilmiah': 'Durio Zibethinus',
       'asal': 'Malaysia',
       'persebaran': 'Asia Tenggara'
      }
print(dic['nama buah'])
```

## Kondisi dan Perulangan

Hal penting yang perlu diperhatikan pada saat membuat pernyataan kondisi adalah **indentasi**, yaitu seberapa banyak spasi yang dibuat pada suatu blok kode pernyataan kondisi maupun perulangan/iterasi (seperti for). Contoh Cell-cell berikut menunjukkan kode untuk pernyataan kondisi maupun iterasi:

```
# pernyataan kondisional
if 5 > 2:
    print("lima lebih dari dua lho")

# operand (operasi pengujian) untuk persamaan menggunakan dua tanda sama dengan ('==')
x = 4
y = (8-x)
if x == y:
    print("x dan y sama")
else:
    print("x dan y beda")

# contoh iterasi. cell di atas (no. 9) harus dijalankan terlebih dahulu
# agar variabel 'ar' didefinisikan
for isi in ar:
    print(isi)
```

## Fungsi (Function)

Fungsi pada Python digunakan antara lain untuk menyederhanakan struktur kode pada Python, sekaligus memungkinkan fungsi untuk dapat dipanggil beberapa kali tanpa harus menuliskan kode yang sama berulang-ulang.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

## **PENGENALAN CITRA DAN POLA**

Pertemuan : 2

**Dasar Pengolahan Citra dan Code  
Python Sederhana**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 26 dari 183

```
# mendefinisikan fungsi menggunakan 'def'
def ini_fungsi(siapaNamanya):
    print(siapaNamanya + " belajar Bahasa Ular")

# memanggil fungsi
ini_fungsi('Wayan Harry')

# memanggil kembali fungsi dengan nilai parameter berbeda
ini_fungsi('Kak Budi')

# lagi
ini_fungsi('Semua orang')
```

### Modul pada Python

Python merupakan bahasa yang sangat kaya. Terdapat ratusan bahkan ribuan *module* yang sudah dibuat oleh berbagai komunitas Python untuk memudahkan fungsi-fungsi tertentu. Berikut adalah sedikit contoh dari beberapa Modul Python:

- [PyTide](#), untuk analisis pasut harmonik
- [Laika](#), untuk pengolahan data GNSS
- [Python Photogrammetry Toolbox](#)
- [SPy](#), untuk pengolahan citra Hyperspectral

Sebuah modul Python dapat berisi ratusan fungsi berbeda. Sebagai contoh, modul '**Datetime**' berisi fungsi untuk menampilkan tanggal, yaitu fungsi *date()*. Demikian juga, terdapat fungsi untuk menampilkan seluruh waktu saat ini sampai ke satuan detik, yaitu *now()*. Keseluruhan modul dapat dipanggil menggunakan perintah:

```
import <module>
```

atau dapat juga dipanggil hanya fungsi tertentu pada satu modul menggunakan perintah:

```
from <module> import <fungsi>
```

Pada praktikum Pengolahan Citra dan Pola ini akan digunakan beberapa modul Python untuk keperluan pengolahan citra, salah satunya bernama [OpenCV](#) yang memiliki fungsi untuk mengolah citra digital dengan fungsi Computer Vision. OpenCV dapat dipanggil menggunakan perintah:

```
import cv2
```

berikut adalah contoh pemanggilan modul Python

```
# memanggil modul 'datetime' untuk menampilkan waktu
# fungsi datetime dinamakan sebagai 'dt' untuk menyingkat pemanggilan
import datetime as dt

# memanggil fungsi datetime.now pada modul datetime
```

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 27 dari 183

```
x = dt.datetime.now()
print(x)

# memanggil hanya fungsi tertentu pada satu modul
from datetime import date

harilni = date.today()

print("Hari ini tanggal: ", harilni)
```

## Library Numpy

Numpy ([numpy.org](https://numpy.org)) merupakan modul Python yang berfungsi untuk menangani operasi Matriks. Numpy dapat digunakan untuk berbagai operasi terkait matriks, dan dapat disebut sebagai 'Matlabnya Python' karena kelengkapan fungsi yang disediakan.

Dalam pengolahan citra digital, sebuah gambar atau foto dapat dianggap sebagai sebuah matriks. oleh karenanya, Numpy dapat membantu beberapa fungsi yang digunakan pada saat pengolahan citra.

```
import numpy as np
a = np.array([1,2,3])

# mencetak dimensi dan tipe data matriks a
print(a.shape)
print(a.dtype)
```

Pada contoh di bawah ini, kita juga akan menggunakan modul **Matplotlib**. [Matplotlib](https://matplotlib.org) merupakan modul Python yang digunakan untuk keperluan mencetak keluaran pada layar: grafik, plot, gambar, dan lain sebagainya. Kita dapat mengkombinasikan Numpy dan Matplotlib untuk mencetak berbagai grafik yang berguna untuk keperluan tertentu, misalnya menampilkan suatu histogram.

Contoh di bawah ini menunjukkan bagaimana Matplotlib digunakan untuk menampilkan sebuah histogram distribusi normal dari serangkaian angka acak yang dibuat menggunakan Numpy.

Jika terjadi error ketika menjalankan coding maka kemungkinan adalah library belum terinstall. Silahkan install menggunakan perintah berikut pada **terminal bukan pada file python**:

```
pip install matplotlib
```

Tunggu hingga proses instalasi selesai dan coba coding berikut:

```
# memanggil numpy dan matplotlib
import numpy
import matplotlib.pyplot as plt
```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1>PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 28 dari 183

```
# membuat distribusi normal dengan mean=5, stdev=1 dan jumlah data 100.000
x = numpy.random.normal(5.0, 1.0, 100000)

# plotting dengan matplotlib
plt.hist(x, 100)
plt.show()
```

## Menggunakan OpenCV

Untuk menggunakan OpenCV, kita akan memanggil fungsi **cv2** dari library OpenCV. Untuk itu, kita tidak perlu melakukan instalasi modul ini menggunakan perintah pip seperti yang kita lakukan pada latihan sebelumnya. Kita langsung dapat memanggil modul-modul **cv2** menggunakan perintah:

```
# memanggil library opencv
import cv2
from cv2 import waitKey

# menyimpan gambar dengan fungsi imread dari OpenCV
img = cv2.imread("gambar/lena.jpg")
# sesuaikan dengan nama file yang diunggah pada cell sebelumnya

# menampilkan gambar dengan fungsi cv2_imshow
cv2.imshow('Lena', img)
waitKey(0)

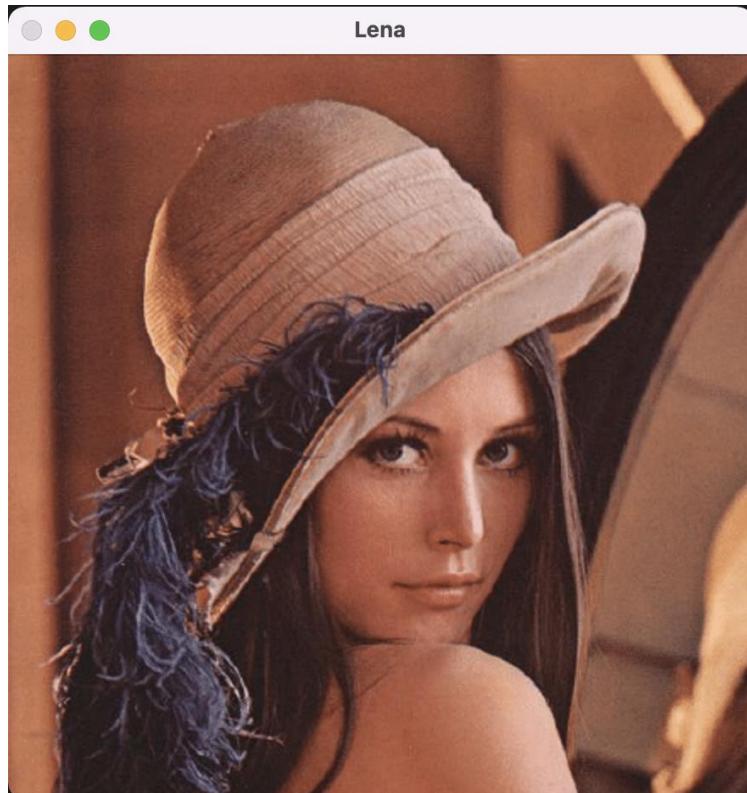
# lihat tipe data img. disimpan sebagai apa?
print(type(img))
```

### Keterangan:

- *import* → fungsi Python untuk memanggil modul di luar fungsi dasar yang disediakan
- *cv2* → nama modul Python yang digunakan untuk memanggil fungsi-fungsi OpenCV
- *numpy* → modul Python untuk pengolahan matriks (ingat bahwa citra pada OpenCV dianggap sebagai matrix) *matplotlib* modul Python untuk melakukan fungsi plotting
- *cv2.imread* → perintah yang digunakan pada modul cv2 untuk memanggil citra
- *cv2.imshow* → perintah yang digunakan pada modul cv2 untuk menampilkan citra. Argumen yang diperlukan adalah nama citra yang dipanggil
- *waitKey(0)* → perintah untuk menunggu sembarang masukan keyboard untuk membatalkan script

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 29 dari 183



Hasil tampilan image

Beberapa perintah lain yang dapat digunakan untuk mengakses properti citra img adalah:

```
print(img.shape) #menampilkan resolusi
print(img.size) #menampilkan ukuran data pada media penyimpan
print(img.dtype) #image datatype (kedalaman bit)
```

### Ekstraksi Band

Sebuah citra dapat dipecah pada band penyusunnya dengan menggunakan perintah cv2.split().

```
# Band blue, green dan red masng-masing disimpan pada variabel b,g,r
b, g, r = cv2.split(img)
```

atau bisa juga dengan menggunakan operasi index pada matriks img:

```
b = img[... ,0] # blue channel
g = img[... ,1] # green channel
r = img[... ,2] # red channel
```

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 30 dari 183

Perhatikan bagaimana tiap band disusun pada variabel yang menyimpan gambar: *masing-masing band disusun sebagai sebuah array di dalam array*. Ingat bahwa pada Python, index sebuah array **dimulai dari angka nol**. Selanjutnya masing-masing band dapat ditampilkan dengan menggunakan cv2\_imshow atau plt.imshow:

```
cv2.imshow('hasil', b); # menampilkan band biru
```

### Transformasi Colorspace

Citra pada OpenCV disimpan dalam susunan BGR (*Blue, Green, Red*). Adakalanya untuk berbagai keperluan perlu dilakukan transformasi colorspace BGR menjadi format lain, misalnya HSV (*Hue Saturation Value*). Penggunaan Colorspace HSV bertujuan untuk mendekati bagaimana cara mata manusia mengenali warna. Contoh penggunaannya adalah untuk melakukan ekstraksi objek berdasarkan warna dan kecerahannya.

Fungsi yang digunakan untuk melakukan transformasi adalah cv2.cvtColor(). Fungsi di atas mentransformasi citra 'img' ke dalam kelas warna HSV menggunakan fungsi COLOR\_BGR2HSV. Untuk fungsi transformasi lain silahkan merujuk dokumentasi OpenCV. Hasil konversi juga dapat dipecah ke dalam masing-masing band.

```
import cv2
from cv2 import waitKey

img = cv2.imread("gambar/lena.jpg")

# konversi BGR dari variable img ke colorspace HSV
hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV);

# memisahkan hue, saturation dan value
h, s, v = cv2.split(hsv)

# menampilkan band hue
cv2.imshow('Hasil', h)
waitKey(0)
cv2.imshow('Hasil', img)
waitKey(0)
```

### H. Tugas Praktikum

Untuk lebih memahami materi pada praktikum 2 silahkan untuk mengerjakan soal latihan berikut

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 2	<b>Dasar Pengolahan Citra dan Code Python Sederhana</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 31 dari 183

- Buat sebuah variabel dengan tipe string, set nilai awal dengan “nama – NIM” (masing-masing mahasiswa). Lalu pisahkan menjadi variabel nama dan nim.

Temp = “I Komang Arya Ganda Wiguna – 1008605018”

Output: 2 variabel (nama dan nim). Nama berisi nama dan NIM berisi NIM.

- Buatlah sebuah fungsi yang dapat menampilkan bintang sejumlah inputan yang diberikan user. Misalkan diberikan inputan 3 akan dihasilkan berikut

\*

\*\*

\*\*\*

Jika diberikan inputkan 6 maka tampilan berikut:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

- Carilah sebuah gambar sembarang lalu letakan pada folder gambar. Baca gambar tersebut lalu tampilkan dengan menggunakan imshow dan matplotlib. Kemudian tampilkan gambar untuk menampilkan masing-masing band dari citra! Lihat perbedaanya

## I. Daftar Referensi

- Hidayatullah, Priyanto. 2017. Pengolahan Citra Digital - Teori dan Aplikasi Nyata. Bandung: Informatika.
- Putra, Darma. 2010. Pengolahan Citra Digital. 1 ed. ed. Westriningsih. Yogyakarta: CV. ANDI OFFSET.
- <https://github.com/danylaksono/OpenCV-PCD>

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

### **PETUNJUK PRAKTIKUM #3**

## **OPERASI DASAR PENGOLAHAN CITRA (OPERASI ARITMATIKA DAN BOOLEAN)**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 3	Operasi Dasar Pengolahan Citra	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 33 dari 183

### A. Capaian Pembelajaran Mata Kuliah

1. Mampu mendefinisikan kebutuhan pengguna atau pasar terhadap kinerja (menganalisis, mengevaluasi dan mengembangkan) algoritma/metode berbasis komputer
2. Mampu mengambil keputusan secara tepat dalam konteks penyelesaian masalah di bidang keahliannya, berdasarkan hasil analisis informasi dan data

### B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu menganalisis dan menguraikan operasi-operasi dasar pada Pengolahan Citra Digital

### C. Tujuan

1. Mahasiswa mampu menjelaskan operasi dasar aritmatika dan boolean pada pengolahan citra digital
2. Mahasiswa mampu menerapkan operasi aritmatika dan boolean pada citra digital

### D. Dasar Teori

#### Operasi Aritmatika

Operasi bisa dilakukan dengan nilai skalar atau dengan citra yang berbeda. Karena Citra digital adalah matriks, maka operasi-operasi aritmatika matriks juga berlaku pada citra. Operasi matriks yang dilakukan adalah:

#### Operasi Penjumlahan

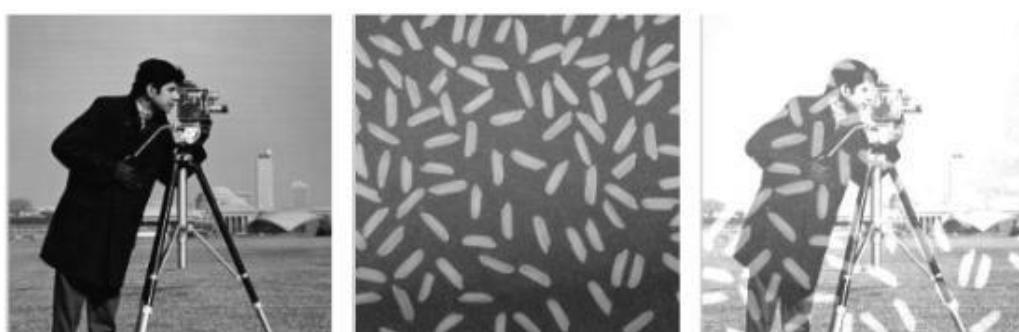
Operasi penjumlahan akan membuat citra lebih cerah. Semakin tinggi nilai piksel maka intensitas semakin cerah. Dua buah citra, A dan B, dengan resolusi spasial dan resolusi keabuan yang sama dapat dijumlahkan dengan operasi penjumlahan matriks.

Misalkan resolusi keabuan kedua citra adalah  $L = 256$ , maka hasil penjumlahan citra A dan B kemungkinan akan mempunyai range nilai  $[0, 2*256]$ . Sehingga pada operasi penjumlahan juga diperlukan proses normalisasi, untuk mengembalikan range nilai menjadi  $[0,256]$  (Hermawati 2013). Lalu jika nilai piksel pada suatu citra ditambahkan, maka sebetulnya efek yang terjadi adalah citra tersebut akan bertambah grayscale-nya sehingga tampak lebih terang. Untuk citra berwarna, penambahan nilai piksel berarti penambahan derajat untuk masing-masing komponen citra (misal R, G, dan B) (Hidayatullah 2017).

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 3	<b>Operasi Dasar Pengolahan Citra</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 34 dari 183



Gambar 3.1 Contoh Penjumlahan Skalar Citra  
Sumber: Google Images



Gambar 3.2 Contoh Penjumlahan Citra Satu dengan Citra Lainnya  
Sumber: Google Images

Namun, perlu diingat bahwa nilai maksimal piksel dari sebuah citra adalah 255 sehingga apabila hasil penambahan tersebut melebihi 255 maka akan diset menjadi 255, sesuai dengan persamaan berikut:

$$f(u) = \begin{cases} u + c, & \text{if } u + c \leq 255 \\ 255, & \text{else} \end{cases}$$

Sebagai contoh apabila terdapat sebuah matrik grayscale dengan ukuran 4x4 dan dilakukan operasi penjumlahan dengan besar nilai 50 maka akan hasil yang didapat adalah sebagai berikut:

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 3	<b>Operasi Dasar Pengolahan Citra</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 35 dari 183

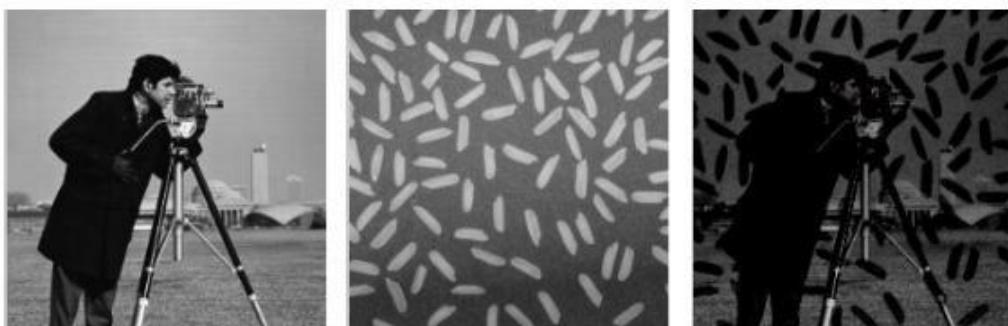
20	189	104	231	→	70	239	154	255
151	168	201	13		201	218	251	63
38	127	177	36		88	177	227	86
123	220	122	172		173	255	172	222

### Operasi Pengurangan

Pengurangan citra adalah operasi saling mengurangkan dua matriks yang berukuran sama. Jika nilai pixel pada suatu citra dikurangi, maka sebetulnya efek yang terjadi adalah citra tersebut akan berkurang grayscale-nya sehingga tampak lebih gelap. Untuk citra berwarna, maka pengurangan dilakukan pada kanal yang bersesuaian (Hidayatullah 2017).



Gambar 3.3 Contoh Pengurangan Skalar Citra  
Sumber: Google Images



Gambar 3.4 Contoh Pengurangan Citra Satu dengan Citra Lainnya  
Sumber: Google Images

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 3	<b>Operasi Dasar Pengolahan Citra</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 36 dari 183

Perlu diingat bahwa nilai minimum piksel lebih kecil dari 0, maka akan diset menjadi 0, sesuai dengan persamaan berikut ini (Hidayatullah 2017).

$$f(u) = \begin{cases} u - c, & \text{if } u - c \geq 0 \\ 0, & \text{else} \end{cases}$$

Untuk proses pengurangan antara dua citra  $f(x,y)$  dan  $h(x,y)$  dinyatakan sebagai:

$$g(x,y) = f(x,y) - h(x,y)$$

Jika hasil pengurangan,  $g(x,y)$ , bernilai negatif, maka harus dinormalkan ke skala grey level semula yaitu  $[0, L-1]$  dengan cara (misalkan  $L = 256$ ):

- A. Tambahkan  $L-1= 255$  ke setiap piksel hasil,  $g(x,y)$  kemudian dibagi 2. Atau.
- B. Tambahkan minimum difference (yang negatif) ke semua piksel  $g(x,y)$ . Lalu kalikan dengan  $(L-1=255)/\text{Max}$ . Di mana Max adalah nilai  $g(x,y)$  paling maksimum setelah dimodifikasi (ditambah minimum difference).

Contoh pertama: Dua buah citra berukuran  $5 \times 4$  dengan resolusi kebuan  $L=8$  sebagai berikut :

Citra A:

2	3	1	4	4
5	1	1	1	3
7	0	7	7	0
2	3	1	5	6

Citra B:

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>			
Pertemuan : 3	Operasi Dasar Pengolahan Citra		100 menit	
Kode MK: TIW-043	Rev : 00		Tgl : 01/03/2022	Hal 37 dari 183

1	0	2	3	1
4	7	6	1	1
7	5	5	5	5
1	4	3	7	2

Citra C = Citra A – Citra B

1	3	-1	1	3
1	-6	-5	0	2
0	-5	2	2	-5
1	-1	-2	-2	6

Hasil pengurangan terdapat nilai negatif, maka perlu dilakukan normalisasi. Hasil normalisasi menggunakan cara sebagai berikut (Hermawati 2013). Perlu diperhatikan nilai L adalah 8 maka untuk L-1 adalah 7.

$$\begin{array}{l}
 A-B = \begin{array}{|c|c|c|c|c|} \hline
 8/2 & 10/2 & 6/2 & 8/2 & 10/2 \\ \hline
 8/2 & 1/2 & 2/2 & 7/2 & 9/2 \\ \hline
 7/2 & 2/2 & 9/2 & 9/2 & 2/2 \\ \hline
 8/2 & 6/2 & 5/2 & 5/2 & 13/2 \\ \hline
 \end{array} \quad = \quad \begin{array}{|c|c|c|c|c|} \hline
 4 & 5 & 3 & 4 & 5 \\ \hline
 4 & 1 & 1 & 4 & 5 \\ \hline
 4 & 1 & 5 & 5 & 1 \\ \hline
 4 & 3 & 3 & 3 & 7 \\ \hline
 \end{array}
 \end{array}$$

Contoh kedua apabila terdapat sebuah matrik grayscale dengan ukuran 4x4 dan dilakukan operasi pengurangan dengan besar nilai 50 maka akan hasil yang didapat adalah sebagai berikut (Hidayatullah 2017):

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 3	Operasi Dasar Pengolahan Citra	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 38 dari 183

128	126	92	153
91	211	42	243
181	164	240	231
60	98	172	189

→ -50 →

78	76	42	103
41	161	0	193
131	114	190	181
10	48	122	139

### Operasi Perkalian

Membuat citra lebih cerah juga dapat dilakukan dengan fungsi perkalian. Perkalian citra A dengan skalar C akan menghasilkan citra baru B yang intensitasnya lebih terang dari semula. Kenaikan intensitas setiap piksel sebanding dengan C. Operasi perkalian citra dengan skalar digunakan untuk kalibrasi kecerahan.

$$B(x, y) = A(x, y) * c$$



Gambar 5. Contoh Perkalian Skalar Citra  
Sumber: Google Images

### Operasi Pembagian

Membuat citra lebih gelap juga dapat dilakukan dengan fungsi pembagian. Pembagian citra A dengan skalar C akan menghasilkan citra baru yang intensitasnya lebih gelap dari semula. Penurunan intensitas setiap piksel berbanding terbalik dengan C. Operasi pembagian citra dengan skalar digunakan untuk normalisasi kecerahan.

$$B(x, y) = A(x, y) / c$$

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 3	Operasi Dasar Pengolahan Citra	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 39 dari 183



Gambar 6. Contoh Pembagian Skalar Citra  
Sumber: Google Images

### Operasi Boolean

Operasi logika yang biasa digunakan adalah operasi NOT, OR, dan AND. Pada citra biner operasi ini dapat dilakukan langsung pada piksel-piksel citra tersebut. Tetapi pada citra dengan resolusi keabuan lebih dari 2 misalkan  $L=8$ , maka nilai grey-level dari suatu piksel dikonversikan terlebih dahulu menjadi bilangan biner. Contohnya untuk nilai grey-level 3 dikonversi menjadi 001,5 adalah 101 dan seterusnya. Sehingga jika terdapat dua citra yang berukuran masih-masing  $4 \times 5$  dan mempunyai resolusi keabuan  $L = 8$  sebagai berikut:

2	3	1	4	4
5	1	1	1	3
7	0	7	7	0
2	3	1	5	6

dan

1	0	2	3	1
4	7	6	1	1
7	5	5	5	5
1	4	3	7	0

Hasil operasi AND dari dua citra tersebut adalah:

010&001=000	011&000=000	001&010=000	100&011=000	100&001-000
101&100=100	001&111=001	001&110=000	001&001=001	011&001=001
111&111=111	000&101=000	111&101=101	111&101=101	000&101=000
010&001=000	011&100=000	001&011=001	101&111=101	110&000=000

atau sama dengan

## PENGENALAN CITRA DAN POLA

Pertemuan : 3

**Operasi Dasar Pengolahan Citra**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 40 dari 183

0	0	0	0	0
4	1	0	1	1
7	0	5	5	0
0	0	1	5	0

Hasil Operasi OR dari dua citra tersebut adalah:

010&001=011	011&000=011	001&010=011	100&011=111	100&001=101
101&100=101	001&111=111	001&110=111	001&001=001	011&001=001
111&111=111	000&101=101	111&101=111	111&101=111	000&101=101
010&001=011	011&100=111	001&011=001	101&111=111	110&000=110

atau sama dengan

3	3	3	7	5
5	7	7	1	3
7	5	7	7	5
3	7	3	7	6

Peningkatan mutu citra dengan menggunakan operasi logika AND dan OR biasanya digunakan untuk tujuan mengambil bagian tertentu dari citra (masking). Dalam citra mask AND dan OR, warna terang atau putih mewakili nilai biner 1 dan sebaliknya warna gelap (hitam) menyatakan nilai biner 0 (Hermawati 2013).

Operasi logika sering kali digunakan pada citra biner yang nilainya hanya memiliki alternatif 0 atau 1. Operasi ini bekerja sesuai dengan prinsip operator logika untuk nilai boolean (true atau false). Nilai 1 itu bernilai true sedangkan nilai 0 berarti false. Prinsip kerjanya juga bisa dipandang seperti operasi pada himpunan.

### NOT

Operator NOT akan menegasikan nilai 1 menjadi 0 dan 0 menjadi 1. Pada teori himpunan operasi ini adalah operasi komplemen.

### AND

Operator AND akan menghasilkan true (1) jika kedua pixel yang dioperasikan bernilai true (1). Pada teori himpunan, operator ini adalah operator irisan.

### OR

## PENGENALAN CITRA DAN POLA

Pertemuan : 3

Operasi Dasar Pengolahan Citra

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

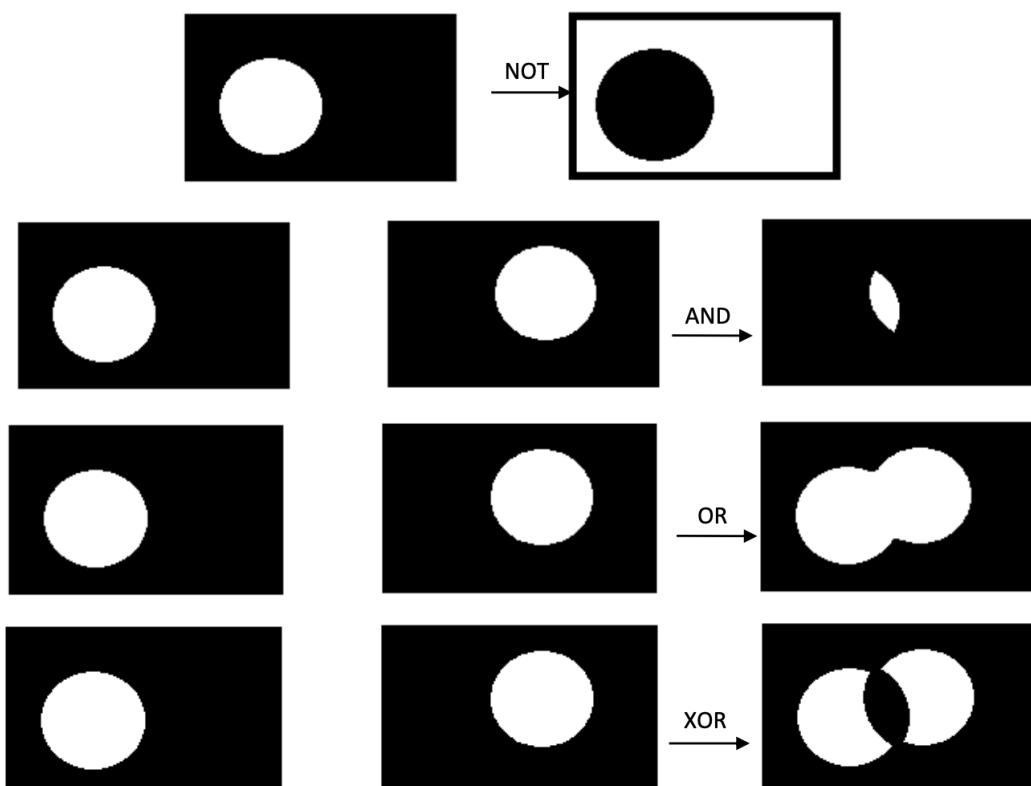
Hal 41 dari 183

Operator OR akan menghasilkan true (1) jika minimal salah satu piksel yang dioperasikan bernilai true (1). Pada teori himpunan, operator ini adalah operator gabungan (U).

### XOR

(Exclusive OR) agar hasil operasi XOR menghasilkan nilai true (1), maka harus salah satu piksel yang bernilai true (1):

Ilustrasi secara lengkap dari operasi logika pada citra dapat dilihat pada gambar berikut:



Gambar 7. Ilustrasi Operasi Logika pada Citra

Sumber: Google Images

Perlu diingat dalam melakukan operasi logika, sama halnya seperti operasi MAX dan MIN yaitu matriks citra pertama dan citra kedua harus memiliki resolusi yang sama (Hidayatullah 2017).

### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 3	<b>Operasi Dasar Pengolahan Citra</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 42 dari 183

2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

#### **F. Keselamatan Kerja**

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporan ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik
9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

#### **G. Lembar Kerja**

Untuk mengerjakan praktikum pertemuan 3 pastikan sudah sukses dalam instalasi python dan library yang dibutuhkan seperti Open CV dan Numpy. Selanjutnya buat file baru dengan format .py pada folder yang sama dengan latihan pada pertemuan 1.

##### **Kegiatan Praktikum 1: Konversi RGB ke Grayscale**

Konversi citra RGB ke citra Grayscale/Keabuan dapat dilakukan dengan beberapa cara. Pertama dengan menggunakan function yang sudah disediakan oleh OpenCV. Kedua dengan menggunakan persamaan tertentu. Pada praktikum kali ini akan menggunakan persamaan berikut:

$$\text{greyscale} = 0.299R + 0.587G + 0.114B$$

Langkah pertama yang dilakukan adalah menyiapkan gambar. Load gambar RGB lalu konversikan ke greyscale menggunakan persamaan diatas. Berikut contoh coding untuk mengubah ke grayscale.

```
import cv2
from matplotlib import pyplot as plt
import numpy as np

img = cv2.imread("gambar/lena.jpg")
H, W = img.shape[2:]
```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

## PENGENALAN CITRA DAN POLA

Pertemuan : 3	Operasi Dasar Pengolahan Citra	100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022

```
gray = np.zeros((H, W), np.uint8)
for i in range(H):
    for j in range(W):
        #Perhatikan format gambar B,G,R
        gray[i,j]= np.clip(0.299 * img[i, j, 2] + 0.587 * img[i, j, 1] + 0.114 * img[i, j, 0], 0, 255)

fig = plt.figure()
fig.add_subplot(211)
plt.imshow(img)
fig.add_subplot(212)
plt.imshow(gray)
plt.show()
```

### Kegiatan Praktikum 2: Operasi Penjumlahan/Pencerahan Citra

Salah satu operasi aritmatika adalah melakukan penjumlahan. Pada praktikum ini masing-masing nilai pixel akan ditambahkan nilai 50. Pertama import library yang dibutuhkan, lalu read image dan pastikan path lokasi gambar sesuai.

```
import cv2
from cv2 import waitKey
from matplotlib import pyplot as plt
import numpy as np
img = cv2.imread("gambar/lena.jpg")

cv2.imshow('Gambar Asli', img)
waitKey(0)
```

Setelah gambar sukses di load selanjutkan adalah mengubah ke citra greyscale. Disini digunakan function yang sudah disediakan oleh library OpenCV. Variabel h dan w digunakan untuk mengetahui height dan width dari gambar yang akan diolah.

```
#mengubah ke greyscale menggunakan function pada OpenCV
img2 = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
brightness = 50
h, w = img2.shape[:2]
```

Lakukan perulangan sejumlah height dan width dari gambar. Setiap perulangan di masing-masing pixel akan ditambahkan nilai 50. Hasil penjumlahan akan dicek apakah melewati 255 atau tidak karena nilai maksimal adalah 255.

```
#perulangan untuk melakukan proses pertambahan
for i in np.arange(h):
    for j in np.arange(w):
        a = img2.item(i, j)
        b = a + brightness

        if b > 255:
            b = 255
```

## PENGENALAN CITRA DAN POLA

Pertemuan : 3

Operasi Dasar Pengolahan Citra

100 menit

Kode MK: TIW-043

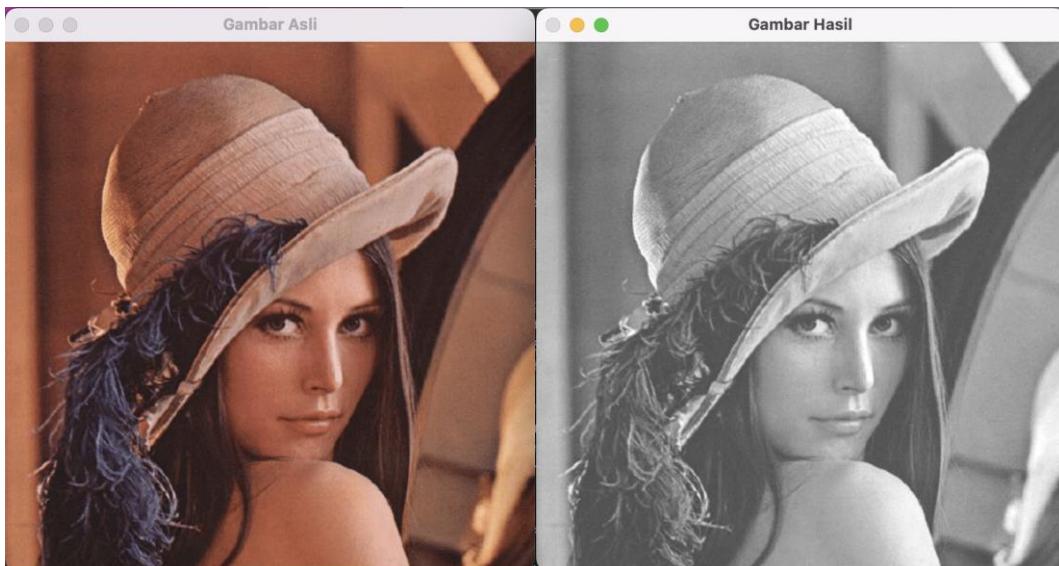
Rev : 00

Tgl : 01/03/2022

Hal 44 dari 183

```
elif b < 0:  
    b =0  
else:  
    b =b  
img2.itemset((i, j), b)
```

Berikut hasil dari proses penjumlahah pada gambar. Tampak pada hasil terlihat adanya peningkatan kecerahan. Untuk melihat perbedaan silahkan bisa mengubah nilai dari variabel brightness.



### Kegiatan Praktikum 3: Operasi Boolean

Untuk praktikum pada operasi boolean siapkan 2 buah gambar. Letakkan pada folder yang sama seperti gambar lena. Load kedua gambar dan parsing ke citra grayscale.

```
img1 = cv2.imread("gambar/gambar1.jpg")  
img2 = cv2.imread("gambar/gambar2.jpg")  
  
img1=cv2.cvtColor(img1,cv2.COLOR_BGR2RGB)  
img2 = cv2.cvtColor(img2, cv2.COLOR_BGR2RGB)
```

Selanjutnya gunakan operator and, or, xor dan not terhadap dua gambar tersebut. Lihat hasil dan analisis perbedaannya.

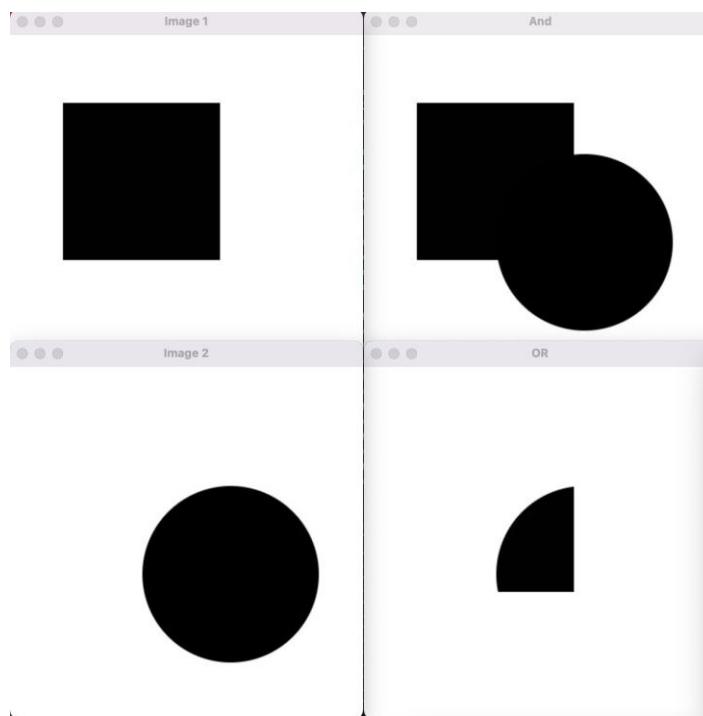
```
op_and=cv2.bitwise_and(img1,img2)  
op_or=cv2.bitwise_or(img1,img2)  
op_xor=cv2.bitwise_xor(img1,img2)
```

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 3	<b>Operasi Dasar Pengolahan Citra</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 45 dari 183

```

cv2.imshow('Image 1', img1)
waitKey(0)
cv2.imshow('Image 2', img2)
waitKey(0)
cv2.imshow('And', op_and)
waitKey(0)
cv2.imshow('OR', op_or)
waitKey(0)
cv2.imshow('XOR', op_xor)
waitKey(0)
    
```

Berikut hasil dari proses operasi boolean.



## H. Tugas Praktikum

Berikut tugas untuk praktikum:

- Siapkan 2 buah gambar lalu lakukan proses aritmatika untuk pengurangan, perkalian dan pembagian pada dua buah gambar!
- Lakukan operasi boolean NOT pada kedua gambar sebelumnya!
- Analisis hasil dari 2 proses aritmatika yang telah di proses pada soal sebelumnya!

## I. Daftar Referensi

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 3	<b>Operasi Dasar Pengolahan Citra</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 46 dari 183

- Hidayatullah, Priyanto. 2017. Pengolahan Citra Digital - Teori dan Aplikasi Nyata. Bandung: Informatika.
- Hermawati, Fajar Astuti. 2013. Pengolahan Citra Digital. Yogyakarta: Andi Publisher.
- Modul Praktikum Pengolahan Citra Digital, Institute Teknologi Bandung, Bandung, 2019
- <https://github.com/danylaksono/OpenCV-PCD>

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

**PETUNJUK PRAKTIKUM #4**

**OPERASI DASAR PENGOLAHAN CITRA  
(OPERASI GEOMETRI)**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 3	<b>Operasi Dasar Pengolahan Citra</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 48 dari 183

### A. Capaian Pembelajaran Mata Kuliah

1. Mampu mendefinisikan kebutuhan pengguna atau pasar terhadap kinerja (menganalisis, mengevaluasi dan mengembangkan) algoritma/metode berbasis komputer
2. Mampu mengambil keputusan secara tepat dalam konteks penyelesaian masalah di bidang keahliannya, berdasarkan hasil analisis informasi dan data

### B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu menganalisis dan menguraikan operasi-operasi dasar pada Pengolahan Citra Digital

### C. Tujuan

1. Mahasiswa mampu menjelaskan operasi geometri pada pengolahan citra digital
2. Mahasiswa mampu menerapkan operasi geometri pada citra digital seperti translasi, rotasi, refleksi, dan penskalaan

### D. Dasar Teori

Secara umum, geometri adalah ilmu ukur atau ilmu bangun adalah cabang matematika yang bersangkutan dengan pertanyaan bentuk, ukuran, posisi relatif gambar, dan sifat ruang. Seorang ahli matematika yang bekerja di bidang geometri disebut ahli geometri. Geometri muncul secara independen di sejumlah budaya awal sebagai ilmu pengetahuan praktis tentang panjang, luas, dan volume, dengan unsur-unsur dari ilmu matematika.

Operasi Geometri adalah proses perubahan hubungan spasial antara setiap pixel pada sebuah citra. Operasi geometri menentukan kembali pixel citra input dari posisi awal ( $x_1, y_1$ ) ke posisi baru ( $x_2, y_2$ ) pada citra output. Proses yang tergolong ke dalam operasi geometri di antaranya adalah translasi, rotasi, refleksi dan penskalaan (Putra 2010).

#### Translasi

Operasi translasi adalah memindahkan setiap elemen pixel citra input ke posisi baru pada citra output dimana dimensi dari kedua citra (citra input dan citra output) pada umumnya adalah sama. Posisi baru dari suatu pixel ditentukan dari nilai variabel translasi ( $p, q$ ). Secara umum operasi translasi melakukan perubahan dengan cara menambahkan koordinat awal dengan nilai variabel translasi.

$$x_2 = x_1 + p$$

$$y_2 = y_1 + q$$

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 49 dari 183

Jika ukuran citra output di-set sama dengan citra input maka bila terdapat posisi hasil yang berada di luar batas citra output, pixel tersebut tidak dipetakan. Untuk posisi citra output yang tidak memiliki nilai piksel diset dengan nilai 0 atau warna hitam (Putra 2010).



Gambar 4.1 Hasil Operasi Translasi

Sumber: dokumen penulis

### Rotasi

Rotasi merupakan suatu transformasi geometri memindahkan nilai-nilai pixel dari posisi awal menuju posisi akhir yang ditentukan melalui nilai variabel rotasi sebesar  $\theta^\circ$  terhadap sudut  $0^\circ$  atau garis horizontal citra.

Proses rotasi dapat dilakukan dengan rumus sebagai berikut.

$$x_2 = \sin(\theta) (x_1 - x_0) - \cos(\theta) (y_1 - y_0) + x_0$$

$$y_2 = \cos(\theta) (x_1 - x_0) + \sin(\theta) (y_1 - y_0) + y_0$$

Dimana  $(x_0, y_0)$  adalah koordinat titik pusat dari citra input dan  $\theta$  adalah sumbu putar. Sumbu putar pada umumnya memiliki arah putar searah jarum jam dengan garis horizontal. Seperti halnya operasi translasi, hasil perhitungan posisi hasil rotasi dapat memberikan nilai di luar batas citra output (apabila ukuran citra output sama dengan citra input).

Untuk kasus seperti itu, ada beberapa implementasi yang memberikan nilai pixel tersebut tanpa dipetakan ulang dan ada yang memetakan ke citra output sehingga menyebabkan ukuran citra membesar (Putra 2010).

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 50 dari 183



Gambar 4.2 Contoh Operasi Rotasi  
Sumber: dokumen penulis

### Refleksi

Refleksi atau pencerminan adalah proses pengolahan citra secara geometri dengan memindahkan nilai-nilai pixel pada posisi awal  $(x_1, y_1)$  ke posisi baru di  $(x_2, y_2)$  pada citra output sesuai dengan posisi pencerminan. Posisi pencerminan ada tiga jenis yaitu pencerminan terhadap sumbu  $x$ , pencerminan terhadap sumbu  $y$ , dan pencerminan terhadap sumbu  $x$  dan  $y$ .

Pencerminan terhadap sumbu  $x$  di posisi  $x_0$  dapat digambarkan dengan rumus sebagai berikut.

$$x_2 = -x_1 + (2 * x_0)$$

$$y_2 = y_1$$

Pencerminan terhadap sumbu  $y$  di posisi  $y_0$  sebagai berikut.

$$x_2 = x_1$$

$$y_2 = -y_1 + (2 * y_0)$$

Pencerminan yang dilakukan terhadap kedua sumbu baik posisi  $x$  dan  $y$  di posisi  $(x_0, y_0)$  sebagai berikut (Putra 2010).

$$x_2 = -x_1 + (2 * x_0)$$

$$y_2 = -y_1 + (2 * y_0)$$

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 51 dari 183



Gambar 4.3 Citra koala Awal (kiri) dan Hasil refleksi (kanan)

Sumber: Wallpaper Windows 7 / Dokumen Penulis

### Penskalaan

Penskalaan adalah sebuah operasi geometri yang memberikan efek memperbesar atau memperkecil ukuran citra input sesuai dengan variabel penskalaan citranya. Ukuran baru hasil penskalaan didapat melalui perkalian antara ukuran citra input dengan variabel penskalaan. Proses penskalaan dapat dilakukan dengan rumus:

$$P_0 = S_p * P_i$$

$$L_0 = S_l * L_i$$

Di mana  $(P_i, L_i)$  adalah ukuran citra input,  $(P_0, L_0)$  adalah ukuran citra output, dan  $(S_p, S_l)$  adalah variabel penskalaan yang diinginkan. Jika variabel penskalaannya lebih besar dari 1 maka hasil penskalaannya akan memperbesar ukuran citra, sebaliknya pabila variabel penskalaannya lebih kecil dari 1 maka hasilnya akan memperkecil ukuran citra (Putra 2010).

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 4	Operaso Dasar Pengolahan Citra (Operasi Geomatri)		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 52 dari 183



Gambar 4.5 Contoh Operasi Penskalaan dengan Citra koala  
Sumber: Dokumen Penulis

#### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

#### F. Keselamatan Kerja

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporan ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik
9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

#### G. Lembar Kerja

##### Kegiatan Praktikum 1: Rotasi Citra

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 53 dari 183

Proses rotasi pada citra dengan library OpenCV dapat menggunakan fungsi `getRotationMatrix2D`. Fungsi ini menyediakan rotasi skala dengan pusat rotasi yang dapat disesuaikan sehingga Anda dapat memutar di lokasi mana pun yang Anda inginkan.

Pertama pastikan melakukan import library

```
import cv2
from cv2 import waitKey
```

Selanjutnya load gambar yang ingin di rotasikan. Pada tahap ini gambar terletak pada folder gambar. Pastikan path untuk mengambil gambarnya sudah sesuai.

```
img = cv2.imread("gambar/lena.jpg")
cv2.imshow('Gambar Asli', img)
H, W = img.shape[:2]
```

Tentukan besaran nilai rotasi yang akan digunakan, misalkan 90/45/-90/-45 derajat. Jika nilai derajat minus (-) maka citra akan berotasi searah jaruh jam.

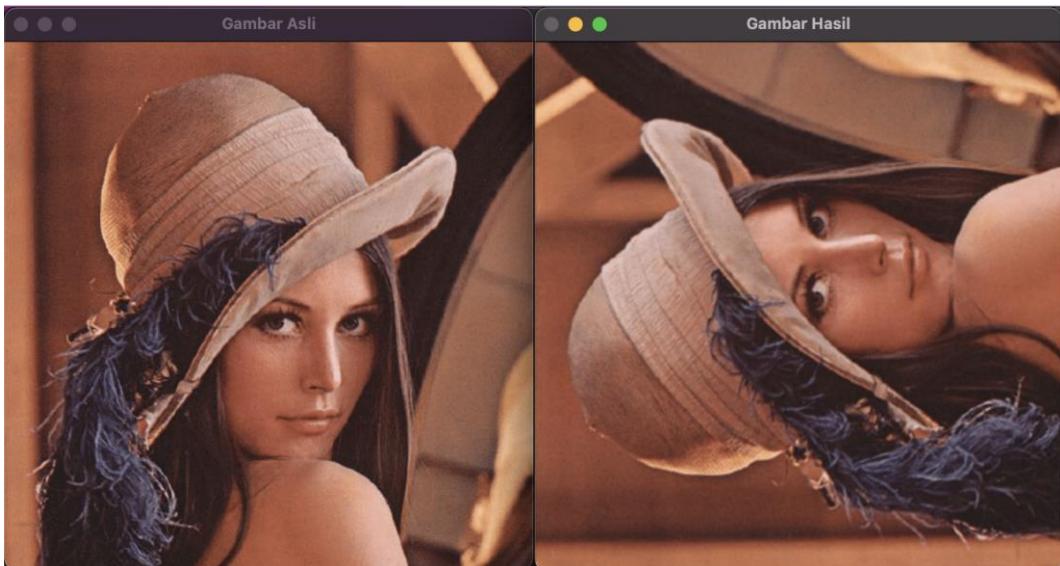
```
degree = 90
rotationMatrix = cv2.getRotationMatrix2D((W / 2, H / 2), degree, 1)

#untuk memposisikan gambar pada titik 0,0
# cos = np.abs(rotationMatrix[0, 0])
# sin = np.abs(rotationMatrix[0, 1])
# print(cos, sin, rotationMatrix[0, 0])
# nW = int((H * sin) + (W * cos))
# nH = int((H * cos) + (W * sin))
# rotationMatrix[0, 2] += (nW / 2) - W / 2
# rotationMatrix[1, 2] += (nH / 2) - H / 2

rot_image = cv2.warpAffine(img, rotationMatrix, (H,W))
cv2.imshow('Gambar Hasil', rot_image)
waitKey(0)
```

Hasil ditunjukkan pada gambar berikut.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 54 dari 183



### Kegiatan Praktikum 2: Translasi Citra

Translasi citra adalah proses menggeser citra dengan menambahkan atau mengurakan koordinat X dan Y. Jika diilustrasikan menggunakan matriks:

$$T = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

$t_x$  dan  $t_y$  adalah nilai translasi, citra akan bergeser dimana x bergeser ke arah horizontal dan y bergeser ke arah vertikal.

Pertama import library, pastikan library numpy sudah terinstall karena pada praktikum ini akan menggunakan matrix. Fungsi **cv2.warpAffine()** adalah fungsi yang digunakan untuk melakukan proses translasi.

```
import cv2
from cv2 import waitKey
import numpy as np

img = cv2.imread("gambar/lena.jpg")
H, W = img.shape[:2]
```

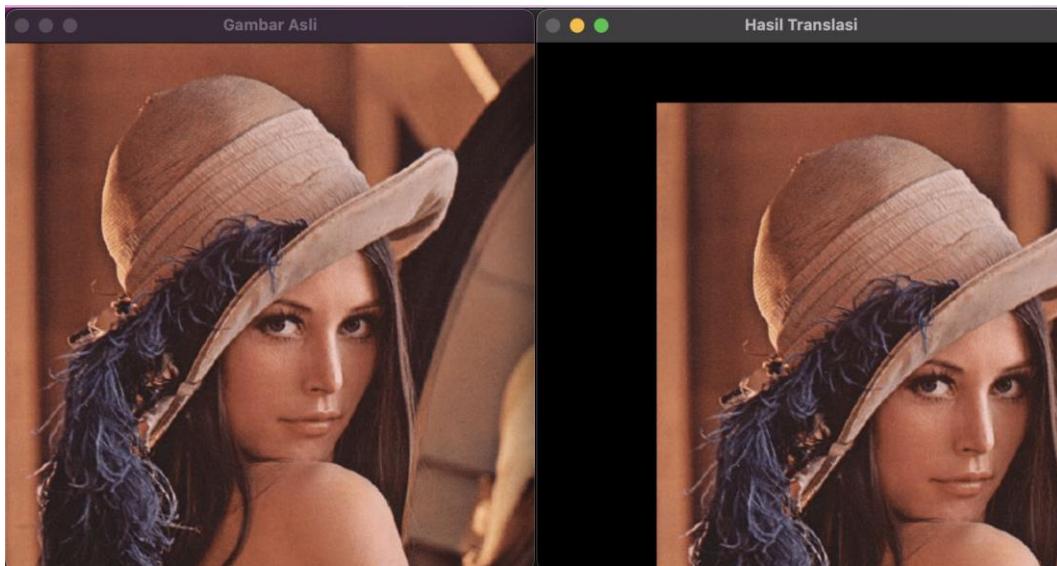
Selanjutnya tentukan nilai pergeseran yaitu 100 untuk  $t_x$  dan 50 untuk  $t_y$ .

```
M = np.float32([[1,0,100],[0,1,50]])
hasil = cv2.warpAffine(img,M,(W,H))

cv2.imshow('Gambar Asli', img)
cv2.imshow('Hasil Translasi', hasil)
cv2.waitKey(0)
```

Hasil ditunjukkan pada gambar berikut:

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 55 dari 183



## H. Tugas Praktikum

Berikut tugas untuk praktikum:

1. Siapkan sebuah gambar lalu lakukan proses geometri untuk melakukan proses refleksi (tidak ditentukan jenis refleksinya)!
2. Buatlah sebuah program untuk dapat memperbesar dan memperkecil ukuran citra sesuai dengan variabel penskalaan yang ditentukan!

## I. Daftar Referensi

- Putra, Darma. 2010. Pengolahan Citra Digital. 1 ed. ed. Westriningsih. Yogyakarta: CV. ANDI OFFSET.

**PETUNJUK PRAKTIKUM #5**

**OPERASI PENGOLAHAN CITRA  
OPERASI TITIK BAGIAN 1**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 57 dari 183

#### A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer
2. Mampu menerapkan konsep teoritis dalam bidang rekayasa perangkat lunak

#### B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu mengimplementasikan metode-metode image enhancement

#### C. Tujuan

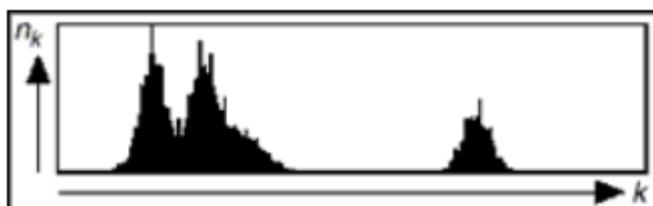
Mahasiswa mampu menjelaskan mengenai konsep image enhancement dan operasi titik pada citra digital serta menguraikan proses pengolahan citra berdasarkan operasi titik seperti brightness, negation, contrast

#### D. Dasar Teori

##### Histogram Citra

Perbaikan kualitas citra (image enhancement) merupakan salah satu proses awal dalam pengolahan citra (image preprocessing). Perbaikan kualitas diperlukan karena seringkali citra yang dijadikan objek pembahasan mempunyai kualitas yang buruk. Misalkan citra mengalami derau (noise) pada saat pengiriman melalui saluran transmisi, citra terlalu terang/gelap, citra kurang tajam, kabur, dan sebagainya. Untuk memperbaiki kualitas citra ini salah satunya adalah dengan perataan histogram (histogram equalization). Misalkan citra digital memiliki  $L$  derajat keabuan, yaitu dari nilai 0 sampai  $L - 1$  (misalkan pada citra dengan kuantisasi derajat keabuan 8-bit, nilai derajat keabuan dari 0 sampai 255). Gambar 1 memperlihatkan contoh sebuah histogram citra, yang dalam hal ini menyatakan derajat keabuan dan  $n_k$  menyatakan jumlah pixel yang memiliki nilai keabuan  $k$  (Yelly N. Nabuasa 2019).

Histogram juga bisa memperlihatkan diagram yang menunjukkan jumlah kemunculan nilai grey-level dan sumbu  $-y$  yang mewakili jumlah kemunculan grey-level tertentu (Hermawati 2013).



 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 58 dari 183

Gambar 5.1 Histogram Citra

Sumber: Google Images

Seringkali pada beberapa operasi pengolahan citra jumlah pixel yang memiliki derajat keabuan k dinormalkan terhadap jumlah seluruh pixel di dalam citra (Yelly N. Nabuasa 2019). Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas piksel dari suatu citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan (relative) citra tersebut. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (brightness) dan kontras (contrast) dari sebuah citra. Secara matematis histogram citra dihitung dengan persamaan:

$$p_r(r_k) = \frac{n_k}{n}; 0 \leq r_k \leq 1; k = 0, 1, 2, \dots, L - 1$$

Keterangan:

$L$ : jumlah level

$p_r(r_k)$ : probabilitas kemunculan level ke- $k$

$n_k$ : jumlah kemunculan level  $k$  pada citra

$n$ : total jumlah pixel dalam citra

Menurut (Hermawati 2013). Ada Empat tipe dasar citra yang dapat digambarkan dengan sebuah histogram, yaitu:

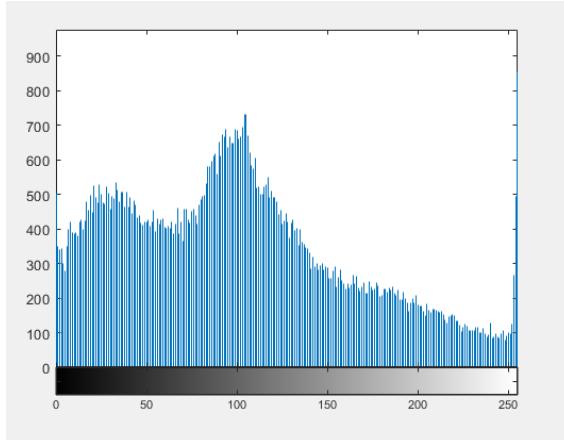
1. Citra gelap: histogram cenderung ke sebelah kiri
2. Citra terang: histogram cenderung ke sebelah kanan
3. Citra low contrast: histogram mengumpul disuatu tempat
4. Citra high contrast: histogram merata disemua tempat

Pengetahuan praktis untuk memahami histogram citra dibutuhkan untuk melihat perubahan-perubahan pada citra setelah dilakukan operasi tertentu. Beberapa pengetahuan praktis yang biasa digunakan dalam melihat histogram citra adalah sebagai berikut:

1. Histogram citra yang terdistribusi merata pada seluruh tingkat keabuan memiliki kontras yang baik.
2. Histogram citra yang mengumpul pada daerah gelap memiliki citra yang redup.
3. Histogram citra yang mengumpul pada daerah terang atau terkonsentrasi pada intensitas citra yang tinggi menampilkan citra yang terang.

Dalam upaya menampakkan informasi sebanyak mungkin pada citra maka histogram dibuat semerata mungkin yang disebut dengan perataan histogram (histogram equalization).

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 59 dari 183



Gambar 5.2 Contoh Histogram dari Citra koala  
Sumber: dokumen penulis

### Histogram Equalization

Histogram equalization adalah metode dalam pengolahan citra yang menggunakan histogram dari suatu citra untuk mengatur tingkat kecerahan citra tersebut. Metode ini menaikkan global contrast dari citra khusunya ketika data citra yang digunakan dapat digantikan dengan nilai kecerahan tetangga terdekatnya. Melalui pengaturan ini, nilai intensitas suatu citra dapat merata dengan baik sesuai dengan histogramnya. Metode ini juga bertujuan untuk menjadikan nilai yang kontras atau kecerahannya kurang menjadi kontars yang memiliki nilai kebih tinggi. Metode histogram equalization menggunakan nilai yang paling sering digunakan untuk menaikkan nilai kontras (Yelly N. Nabuasa 2019).

Singkatnya dengan mengubah pemenetaan grey level agar sebarannya (kontrasnya) lebih luas yaitu 0-255. Sifat dari histogram equalization adalah:

1. Grey-Level yang sering muncul lebih jarang jaraknya.
2. Grey-Level yang jarang muncul bisa lebih dirapatkan jaraknya dengan grey-level sebelumnya
3. Histogram baru pasti mencapai nilai maksimal keabuan (Hermawati 2013).

Tujuan ekualisasi histogram adalah untuk memperoleh penyebaran histogram yang merata, sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama atau seragam. Dasar konsepnya adalah transformasi probability density function menjadi uniform density (bentuk kontinyu) dan agar dapat dimanfaatkan dalam pengolahan citra digital.

Keadaan citra yang tepat untuk dijadikan sebagai objek dari histogram equalization adalah citra dengan kedekatan kontras yang sangat tinggi. Artinya dynamic range kecil atau lebih mudahnya variasi level abu-abu sangat kecil atau sempit. Contoh citra seperti ini adalah citra dengan latar belakang (background)

## PENGENALAN CITRA DAN POLA

Pertemuan : 4

**Operaso Dasar Pengolahan Citra**  
**(Operasi Geomatri)**

100 menit

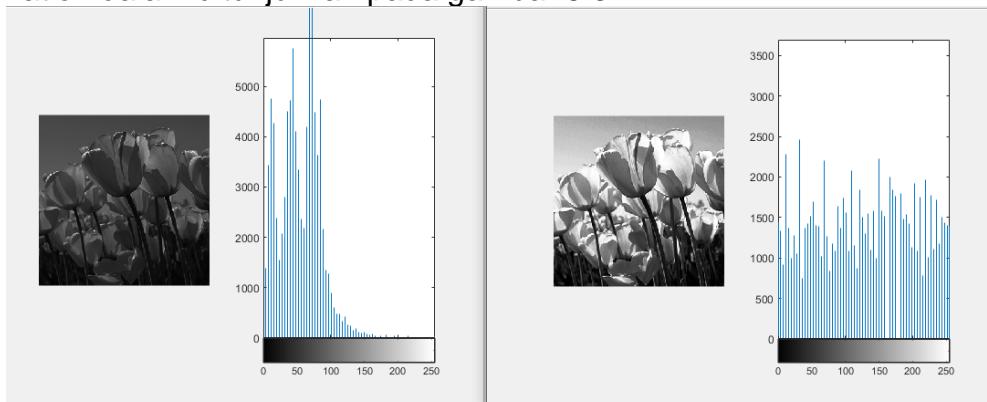
Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 60 dari 183

dan objek (foreground) sama-sama berwarna cerah atau keduanya sama-sama berwarna gelap. Bukan berarti metode ini tidak bisa digunakan untuk citra jenis lain, hanya saja tidak memberikan pengaruh besar. Definisi dynamic range secara sederhana diartikan sebagai range antara intensitas paling redup (gelap) hingga kepaling cerah (terang) (Yelly N. Nabuasa 2019). Contoh implementasi histogram equalization dalam ditunjukkan pada gambar 5.3.



Gambar 5.3 Histogram Ekualisasi: Sebelum (Kiri) dan Sesudah (Kanan)

Sumber: dokumen penulis

Terdapat pula ekualisasi histogram lain, seperti:

1. Histogram Processing Specific Grey-Level (Histogram Specification) Ekualisasi yang dilakukan tidak pada seluruh bagian dari histogram tapi hanya pada bagian dengan derajat keabuan tertentu saja. Metode ini diterapkan untuk mengatasi kekurangan dari histogram equalization dimana kadangkala, hasil histogram ekualisasi tidak sesuai dengan yang diharapkan.
2. Histogram Equalization Spesific Area (Local Enhancement) Ekualisasi hanya dilakukan pada bagian tertentu dari citra (Hermawati 2013).

### Brightness Adjustment

Penyesuaian kecerahan (brightness) merupakan operasi pixel yang paling sederhana. Tingkat kecerahan suatu citra dapat dilihat dari histogramnya. Semua pixel biasanya terkonsentrasi pada salah satu sisi histogram dengan rentangan gray level tertentu. Semakin dinaikkan tingkat brightness suatu citra maka konsentrasi nilai pixel pada histogram akan bergeser ke sisi kanan, demikian juga sebaliknya, semakin diturunkan maka konsentrasi nilai pixel pada histogram akan bergeser ke sisi kiri.

## PENGENALAN CITRA DAN POLA

Pertemuan : 4

Operaso Dasar Pengolahan Citra  
(Operasi Geomatri)

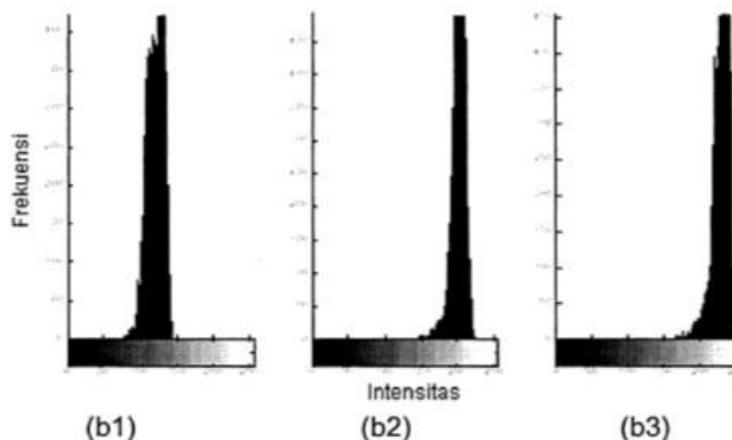
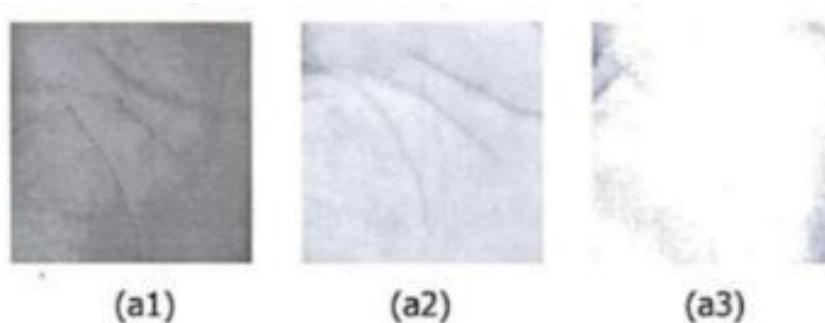
100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 61 dari 183



Gambar 5.4 Perbandingan Citra (a) dengan Histogramnya (b)  
Sumber: Google Images

Penyesuaian tingkat brightness dapat dinyatakan sebagai :

$$U' = U + c$$

Dengan  $U'$  dan  $U$  berturut-turut menyatakan citra setelah dan sebelum penyesuaian kecerahan sedangkan  $c$  adalah suatu konstanta yang merupakan faktor penyesuaian. Proses penyesuaian atas dilakukan dengan menambahkan (bisa juga mengurangkan) nilai setiap pixel dengan suatu konstanta. Apabila nilai pixel setelah penyesuaian melebihi nilai maksimum intensitas yang mungkin untuk citra grayscale. Nilai maksimum intensitas adalah 255 maka nilai pixel tersebut akan dijadikan 255. Demikian pula sebaliknya, bila nilai pixel hasil penyesuaian lebih kecil dari 0 (nol) maka nilai pixel tersebut dijadikan 0.

### Negation

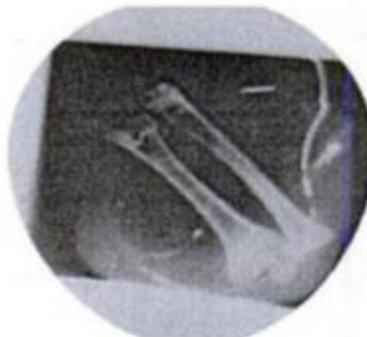
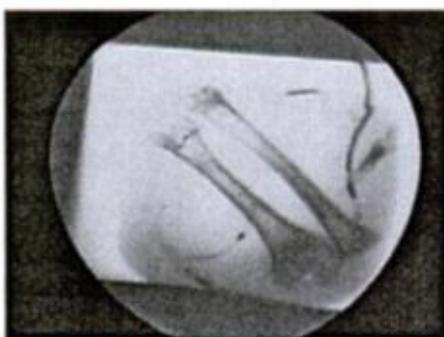
Negasi adalah proses pemetaan nilai pixel suatu citra, yaitu pada citra biner, pixel hitam dijadikan putih dan putih dijadikan hitam. Pada citra grayscale atau

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 62 dari 183

berwarna, nilai maksimum pixel dikurangi dengan nilai pixel yang sedang diproses. Secara matematis, negasi dapat dinyatakan sebagai berikut.

$$U' = 2^l - U$$

Dengan  $U'$  dan  $U$  berturut-turut menyatakan citra setelah dan sebelum proses negasi dan  $l$  menyatakan nilai bit dari gray level.



Gambar 5.5 Contoh Negasi: Citra Awal (Kiri) dan Citra Negasi (Kanan)  
Sumber: Google Images

### Koreksi Gamma (Gamma Correction)

Brightness suatu citra juga dapat diperbaiki dengan menggunakan koreksi gamma. Bentuk umum dari transformasi gamma sebagai berikut.

$$U' = U^{\frac{1}{y}}$$

Dengan  $U'$  dan  $U$  berturut-turut adalah citra setelah dan sebelum mengalami koreksi gamma. Sedangkan  $y$  adalah faktor koreksi gamma. Bila  $y = 1$  maka hasil transformasi akan sama dengan citra masukan. Semakin kecil faktor koreksi maka citra output akan semakin gelap dan semakin tinggi faktor koreksi maka citra output akan mendekati citra asli (lebih terang).

## PENGENALAN CITRA DAN POLA

Pertemuan : 4

**Operaso Dasar Pengolahan Citra**  
**(Operasi Geomatri)**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 63 dari 183



(a)



(b)



(c)



(d)

Gambar 5.6 Contoh Koreksi Gamma: Citra Asli (a), Faktor Koreksi 0.25 (b),  
Faktor Koreksi 0.5 (c), dan Faktor Koreksi 0.65 (d)

Sumber: Google Images

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 64 dari 183



Gambar 5.7 Hasil Koreksi Gamma dengan Faktor 0.5  
pada Citra Koala: Citra Awal (Kiri) dan Citra Koreksi (Kanan)  
Sumber: Google Images

### Peregangan Kontras (Contrast Stretching)

Kontras suatu citra adalah distribusi pixel terang dan gelap. Citra grayscale dengan kontras rendah maka akan terlihat terlalu gelap, terlalu terang, atau terlalu abu-abu. Histogram citra dengan kontras rendah, semua pixels akan terkonsentrasi pada sisi kiri, sisi kanan, atau tengah. Semua pixel akan terkelompok secara rapat pada suatu sisi tertentu dan menggunakan sebagian kecil dari semua kemungkinan nilai pixel citra dengan kontras tinggi memiliki daerah gelap dan terang yang luas. Histogram citra dengan kontras memiliki dua puncak besar. Satu puncak terkonsentrasi pada sisi kiri dan yang satunya terkonsentrasi pada sisi kanan histogram. Citra dengan kontras yang bagus menampilkan rentangan nilai pixel yang lebar. Histogramnya relatif menunjukkan distribusi nilai pixel yang seragam, tidak memiliki puncak utama, atau tidak memiliki lembah.

Peregangan kontras adalah teknik yang sangat berguna untuk memperbaiki kontras citra terutama citra yang memiliki kontras rendah. Teknik ini bekerja dengan baik pada citra yang memiliki distribusi gaussian atau mendekati distribusi gaussian.

## PENGENALAN CITRA DAN POLA

Pertemuan : 4

**Operaso Dasar Pengolahan Citra**  
**(Operasi Geomatri)**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

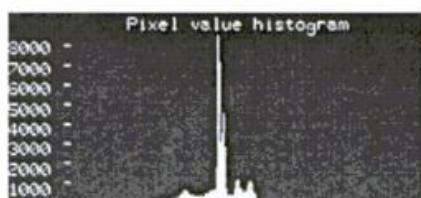
Hal 65 dari 183



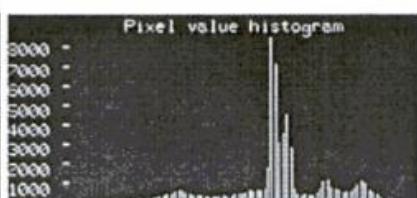
(a)



(b)



(c)



(d)

(a) Citra asli

(b) Citra hasil perenggangan kontras dengan nilai  $c = 79$ ,  $d = 136$

(c) Histogram gambar (a)

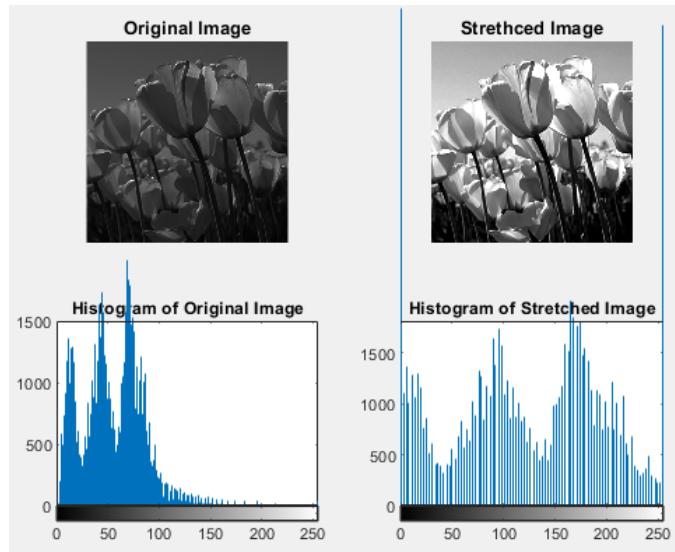
(d) Histogram gambar (b)

Gambar 5.8: Citra Asli (a) dengan Peregangan Kontras (b) Beserta Masing-masing Histogramnya (c dan d)

Sumber: Google Images

Gambar 8 menunjukkan contoh peregangan kontras. Citra masukan memiliki kontras rendah, terlalu abu-abu (histogram terpusat di tengah), kemudian diregangkan dengan nilai  $c = 79$  dan  $d = 136$  sehingga citra hasil lebih jelas (histogram lebih menyebar).

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 66 dari 183



Gambar 5.9 Contoh Peregangan Kontas Beserta Histogramnya:  
Sebelum (Kiri) dan Sesudah (Kanan)

Sumber: dokumen penulis

#### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

#### F. Keselamatan Kerja

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporan ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 67 dari 183

9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

## G. Lembar Kerja

### Kegiatan Praktikum 1: Menampilkan histogram

Siapkan sebuah gambar dan simpan didalam folder yang sudah di tentukan. Selanjutnya import modul dan baca gambar yg telah disimpan. Pastikan path alamat gambar sudah sesuai.

```
# memanggil library yang dibutuhkan
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread("gambar/bunga.jpg")
```

Histogram dapat ditampilkan dengan menggunakan banyak cara. Fungsi yang dapat digunakan antara lain cv2.calcHist(), np.histogram() dan plt.hist(). Pada praktikum ini akan menggunakan cv2.calcHist(). Parameter untuk fungsi cv2.calcHist() adalah sebagai berikut:

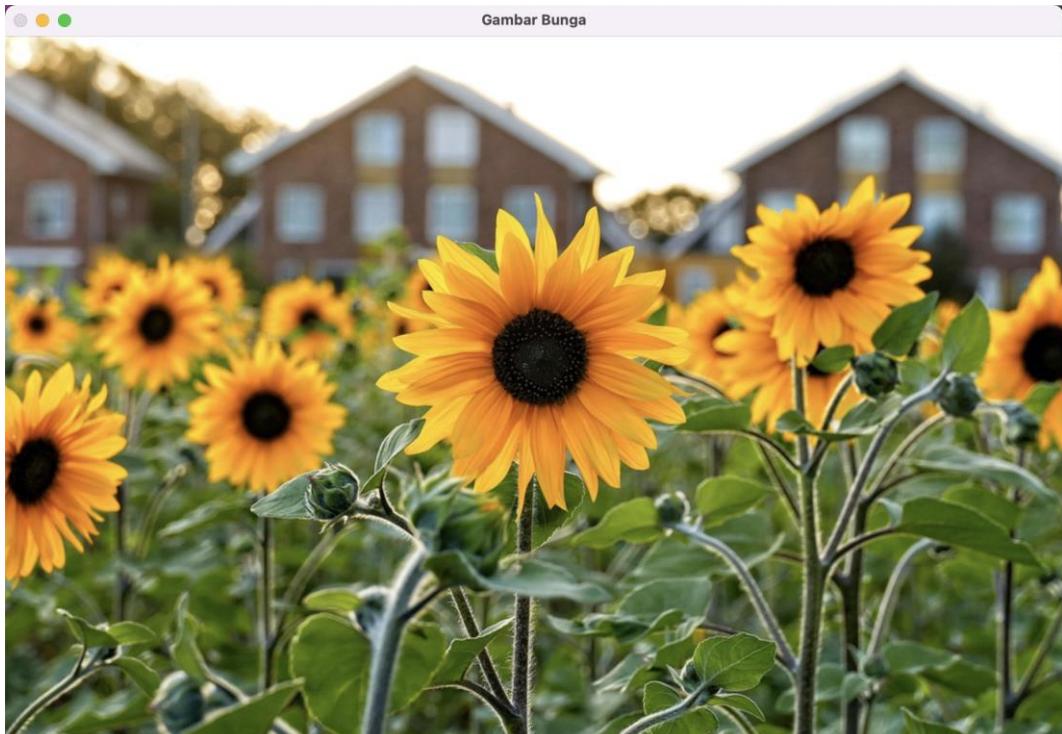
```
cv.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]])
```

- images : gambar yang dibaca menggunakan fungsi cv.read dengan type uint8 or float32. Gambar ini harus dinyatakan sebagai sebuah array dengan kurung siku "[img]".
- channels : band yang akan dihitung histogramnya. Gunakan [0] untuk citra grayscale. Untuk citra berwarna, kita dapat berikan nilai [0], [1] atau [2] untuk menghitung histogram dari masing-masing band.
- mask : digunakan apabila kita ingin memperoleh histogram dari sebagian citra saja.
- histSize : Ukuran histogram, atau banyaknya kolom yang digambar. Kita dapat gunakan nilai sesuai dengan jumlah pixel [256].
- ranges : rentang nilai. Pada citra 8 bit, nilainya adalah [0,256].

Karena gambar yang disimpan dan dibaca dalam bentuk RGB maka akan digunakan 3 channel yaitu B,G,R. Perhatikan format baca pada opencv adalah BGR bukan RGB. Berikut contoh kodennya:

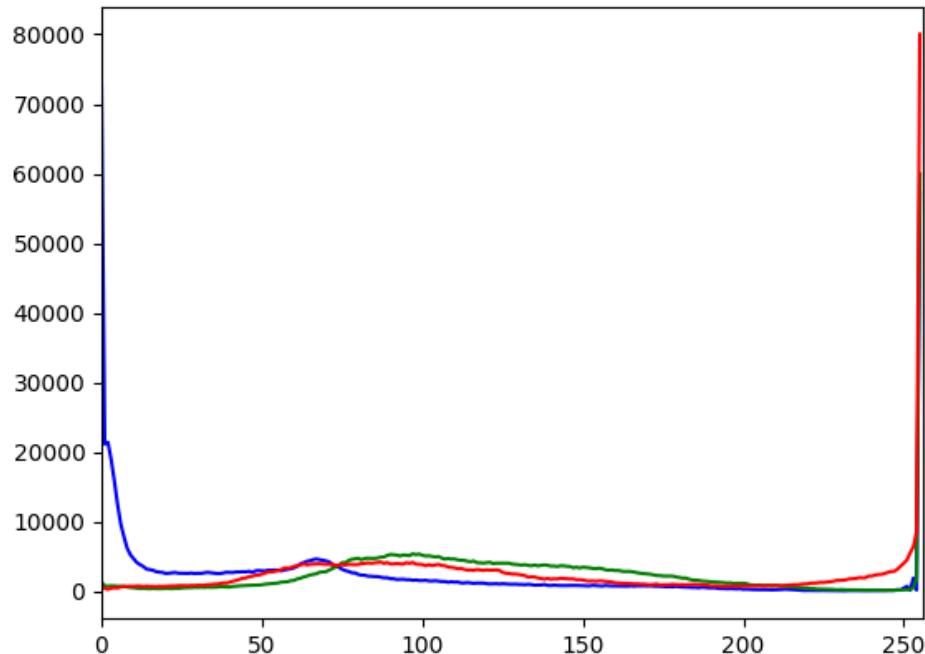
```
color = ('b','g','r')
for i,col in enumerate(color):
    histr = cv2.calcHist([img],[i],None,[256],[0,256])
    plt.plot(histr,color = col)
    plt.xlim([0,256])
plt.show()
```

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1>PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 68 dari 183



Tampilan bunga akan menghasilkan histogram seperti berikut:

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 69 dari 183



### Kegiatan Praktikum 2: Negation

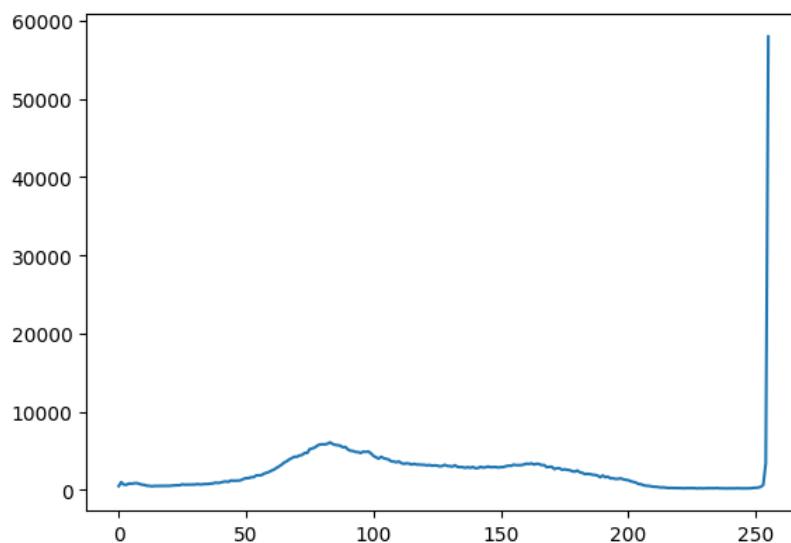
Untuk membuat citra negasi pertama ubah citra warna menjadi citra grayscale. Lalu cek nilai histogram dari citra grayscale.

```
# memanggil library yang dibutuhkan
import cv2
import numpy as np
from matplotlib import pyplot as plt
from cv2 import waitKey

img = cv2.imread("gambar/bunga.jpg")
img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
cv2.imshow('Gambar Bunga Grayscale', img)
histr = cv2.calcHist([img],[0],None,[256],[0,256])
plt.plot(histr)
plt.show()
```

Berikut hasil dari perubahan ke citra grayscale dan nilai histogramnya. Untuk citra grayscale hanya ada 1 channel untuk nilai histrgoram. Pada gambar di bawah tampak intensitas tinggi pada nilai level atas.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 70 dari 183



Untuk mengubah menjadi citra negation dapat dengan menentukan nilai maksimal intensitas dan mengurangi dengan nilai pixel pada tiap posisi.

```

h, w = img.shape[:2]
max_intensity = 255
for i in range(h):
    for j in range(w):
        a = img.item(i, j)
        b = max_intensity - a
        img.itemset((i, j), b)
    
```

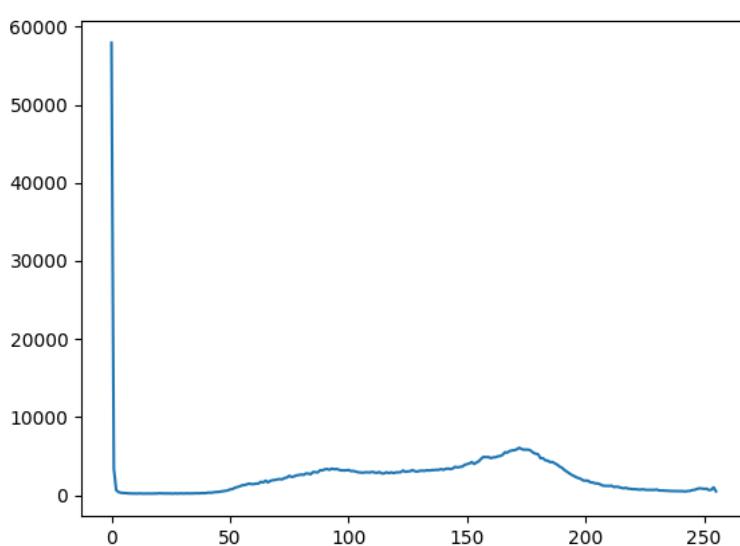
 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 71 dari 183

```

cv2.imshow('Gambar Hasil', img)
histr = cv2.calcHist([img],[0],None,[256],[0,256])
plt.plot(histr)
plt.show()
waitKey(0)

```

Berikut hasil dari perubahan ke citra negation dan nilai histogramnya. Pada gambar di bawah, nilai histogram mengalami perubahan ke dominan pada nilai level rendah/bawah.



## Kegiatan Praktikum 2: Gamma Correction

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 72 dari 183

Gamma correction juga dikenal sebagai Power Law Transform. Pertama, intensitas piksel gambar harus diskalakan dari kisaran [0, 255] menjadi [0, 1.0]. Selanjutnya dikonversikan kembali ke range [0, 255].

```
def gammaCorrection(image, gamma=1.0):
    # build a lookup table mapping the pixel values [0, 255] to their adjusted gamma values
    invGamma = 1.0 / gamma
    table = np.array([(i / 255.0) ** invGamma * 255
                     for i in np.arange(0, 256)]).astype("uint8")
    # apply gamma correction using the lookup table
    return cv2.LUT(image, table)
```

Gamma correction dapat diimplementasikan dengan menggunakan tabel pencarian (LUT). Tahap ini memetakan nilai piksel input ke nilai output. Untuk setiap nilai piksel dalam rentang [0, 255] dihitung nilai koreksi gamma yang sesuai. OpenCV menyediakan fungsi LUT yang melakukan transformasi tabel pencarian.

```
img = cv2.imread("gambar/bunga.jpg")
gammalmg = gammaCorrection(img, 2.2)

cv2.imshow('Original image', img)
cv2.imshow('Gamma corrected image', gammalmg)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Nilai gamma  $< 1$  akan menggeser gambar ke arah ujung spektrum yang lebih gelap sedangkan nilai gamma  $> 1$  akan membuat gambar tampak lebih terang. Nilai gamma  $G=1$  tidak akan memengaruhi gambar. Berikut hasil pengolahan gamma correction.



Gamma: 1



Gamma: 0.5

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 4	<b>Operaso Dasar Pengolahan Citra (Operasi Geomatri)</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 73 dari 183



Gamma: 2.5

## H. Tugas Praktikum

1. Carilah sembarang gambar dan simpan di dalam project. Lalu lakukan operasi brightness adjustment dengan nilai C adalah 75. Tampilkan hasil sebelum dan setelahnya
2. Dari gambar nomer 1 ubahlah gambar tersebut ke dalam citra HSV lalu tampilkan histogram untuk band/channel Hue, Saturation dan Value.

## I. Daftar Referensi

- Hermawati, Fajar Astuti. 2013. Pengolahan Citra Digital. Yogyakarta: AndiPublisher.
- Yelly N. Nabuasa. 2019. "Pengolahan Citra Digital Perbandingan Metode Histogram Equalization dan Spesification pada Citra Abu-abu." J-ICON - Jurnal Komputer dan Informatika 7(1): 87–95.
- <https://github.com/danylaksono/OpenCV-PCD>

**PETUNJUK PRAKTIKUM #6**

**OPERASI PENGOLAHAN CITRA  
OPERASI TITIK BAGIAN 2**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 6	Operasi Pengolahan Citra: Operasi Titik Bagian 2		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 75 dari 183

#### A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer
2. Mampu menerapkan konsep teoritis dalam bidang rekayasa perangkat lunak

#### B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu mengimplementasikan metode-metode image enhancement

#### C. Tujuan

Tujuan Mahasiswa mampu menjelaskan mengenai konsep image enhancement dan operasi titik pada citra digital serta menguraikan proses pengolahan citra berdasarkan operasi titik seperti intensity slicing, bit extraction, range compression, ekualisasi histogram, ekualisasi histogram adaptif

#### D. Dasar Teori

Perbaikan citra atau image enhancement bertujuan meningkatkan kualitas tampilan citra untuk mengkonversi suatu citra agar memiliki format yang lebih baik sehingga citra tersebut menjadi lebih mudah diolah dengan mesin (komputer). Perbaikan terhadap suatu citra dapat dilakukan dengan operasi titik (point operation), operasi spasial (spatial operation), operasi geometri (geometric operation), dan operasi aritmatik (arithmetic operation) (Putra 2010).

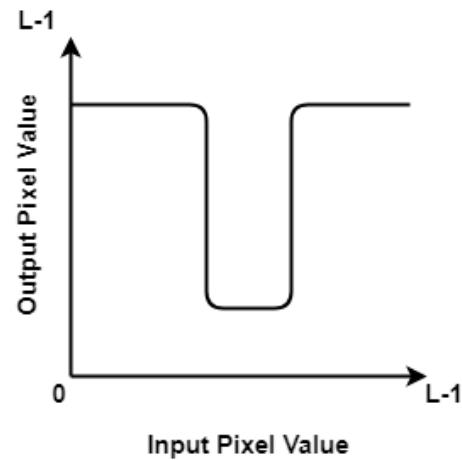
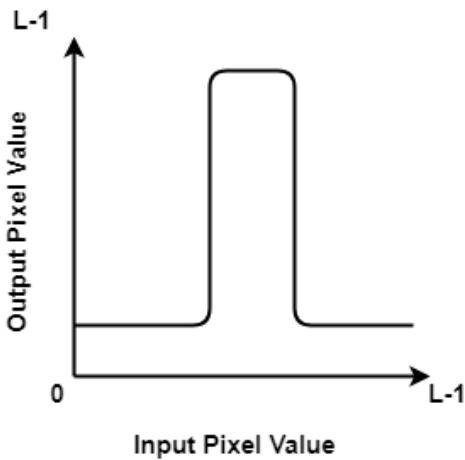
#### Pengirisan Intensitas (Intensity Slicing)

Pengirisan intensitas berguna ketika ingin menonjolkan intensitas (gray level) tertentu saja pada citra. Ada dua pendekatan yang dapat digunakan untuk melakukan pengirisan intensitas sebagai berikut. Pertama, memberi nilai tinggi pada rentangan nilai intensitas yang ingin ditonjolkan dan nilai intensitas lainnya tetap dipertahankan. Pendekatan pertama dipilih bila latar belakang diabaikan, sedangkan pendekatan kedua untuk mempertahankan latar belakang citra. Secara matematis, kedua pendekatan diatas dapat dinyatakan sebagai berikut. Pendekatan pertama:

$$o = \begin{cases} L, & \text{jika } a \leq u \leq b \\ 0, & \text{untuk lainnya} \end{cases}$$

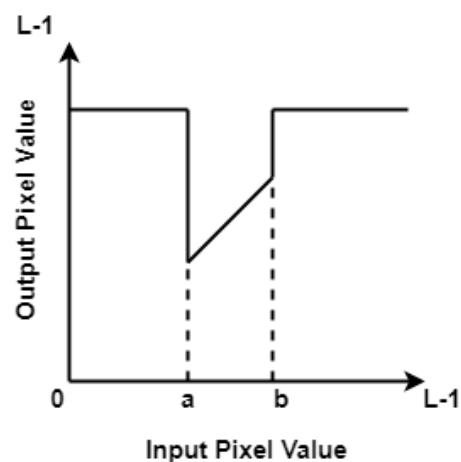
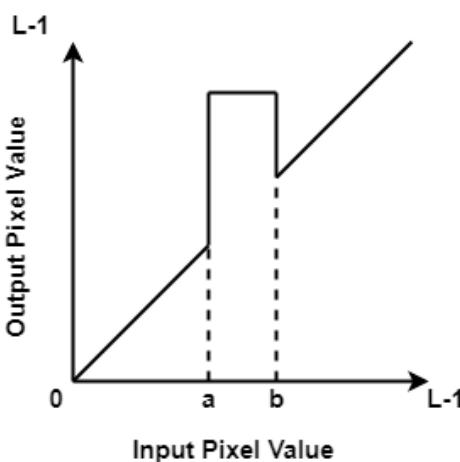
Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 6	Operasi Pengolahan Citra: Operasi Titik Bagian 2	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 76 dari 183



Pendekatan kedua:

$$o = \begin{cases} L, & \text{jika } a \leq u \leq b \\ u, & \text{untuk lainnya} \end{cases}$$



Dengan a dan b berturut-turut menyatakan nilai batas bawah dan atas intensitas yang mau ditonjolkan (Putra 2010).

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 6	<b>Operasi Pengolahan Citra: Operasi Titik Bagian 2</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 77 dari 183



Gambar 6.1 Implementasi Intensity Slicing pada Citra koala  
Sumber: dokumen penulis

### Pemisahan Bit (Bit Extraction)

Suatu pixel merupakan kombinasi dari beberapa significant bit, misal suatu pixel dengan nilai maximum 255 memiliki 8 significant bit, yaitu dari bit ke-0 sampai ke-7. Sesuai dengan namanya, metode pemisahan bit memisahkan beberapa bit dari pixel untuk menghasilkan nilai pixel yang baru. Metode ini dijelaskan sebagai berikut.

Setiap pixel pada citra dapat dinyatakan dengan komposisi bit sebagai berikut (Putra 2010).

$$u = k_1 2^{b-1} + k_1 2^{b-2} + \cdots + k_n 2^{b-n} + \cdots + k_{b-1} 2^1 + k_b$$

Nilai pixel baru dapat diperoleh dengan,

$$o = \begin{cases} L, & \text{jika } k_n = 1 \\ 0, & \text{untuk lainnya} \end{cases}$$

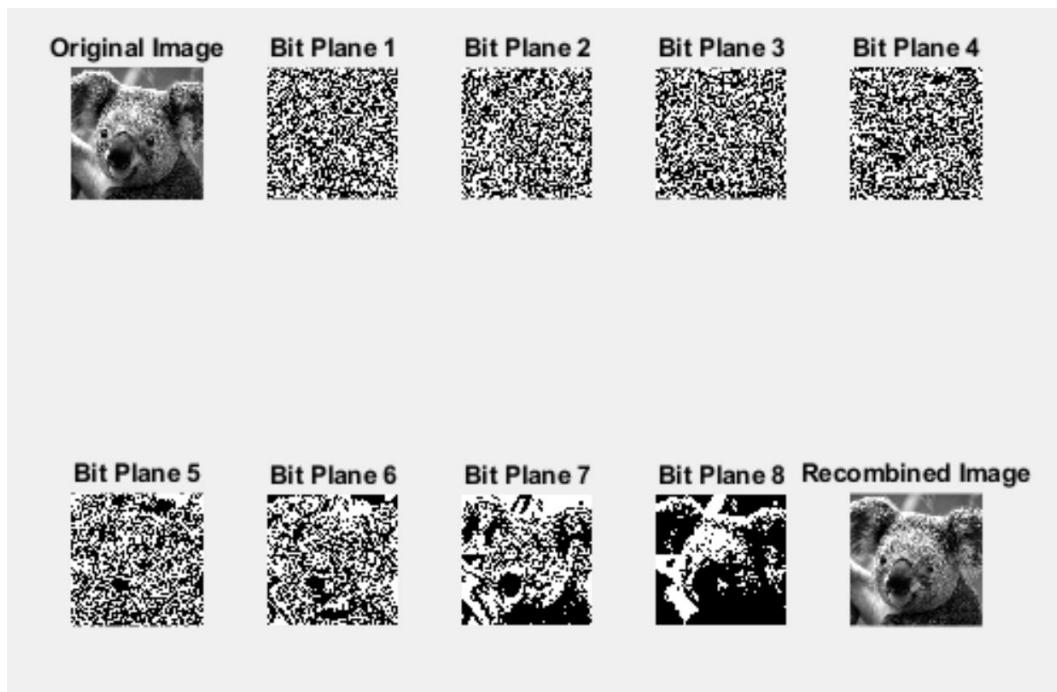
Dengan nilai k,

$$k_n = i_n - 2i_{n-1}$$

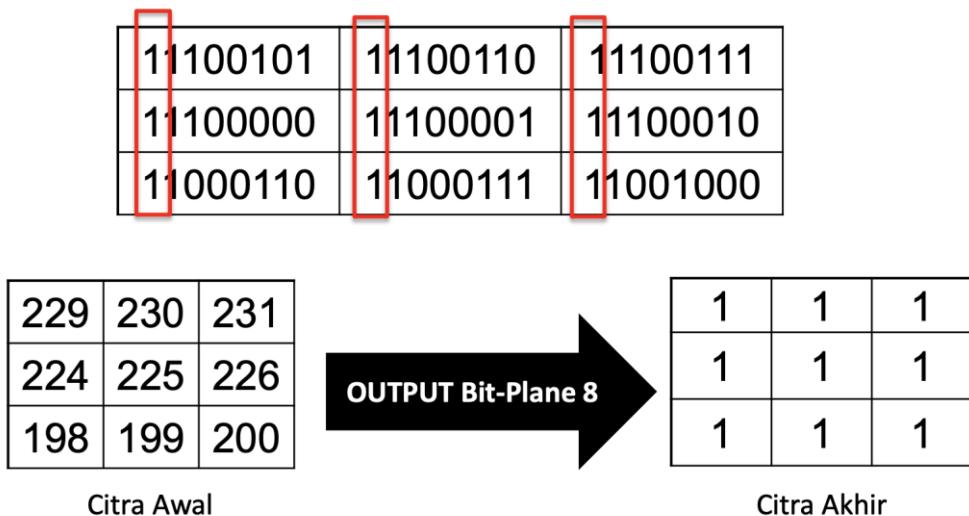
Dan

$$i_n = \int \left[ \frac{u}{2^{b-n}} \right] \int [x] = \text{bagian integer dari } x$$

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 6	Operasi Pengolahan Citra: Operasi Titik Bagian 2		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 78 dari 183



Gambar 6.2 Contoh Penerapan Pemisahan Bit pada Citra koala  
Sumber: dokumen penulis



### Pemampatan Rentangan (Range Compression)

Rentangan dinamis pixel suatu citra terkadang sangat besar, di mana hanya sedikit pixel yang terlihat. Untuk mengatasi permasalahan ini, dapat dilakukan

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 6	Operasi Pengolahan Citra: Operasi Titik Bagian 2	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 79 dari 183

dengan menempatkan rentangan yang besar tadi dengan metode pemampatan rentangan atau sering juga disebut metode transformasi logaritmik, yang dapat dinyatakan sebagai berikut.

$$o = c \log_{10}(1 + |u|)$$

Dengan  $o$  dan  $u$  berturut-turut menyatakan nilai pixel setelah dan sebelum diolah, dan  $c$  merupakan faktor penskalaan sebagai berikut.

$$c = \frac{255}{\log(1 + |R|)}$$

Dengan  $R$  merupakan nilai maksimum magnitude pada citra input. Penghitungan dengan nilai  $c$  dengan cara di atas akan dapat menjaga nilai maksimum pixel dari citra output adalah 225 (Putra 2010).

### Ekuallisasi Histogram

Citra yang memiliki kontras rendah bisa diperbaiki dengan berbagai teknik, salah satunya histogram equalization. Ide dasarnya adalah semua pixel harus didistribusikan secara merata ke seluruh rentang nilai yang ada. Artinya tujuan dari histogram adalah mengubah citra sehingga citra output memiliki histogram yang lebih datar. Secara matematis ekuallisasi histogram dapat dihitung menggunakan persamaan sebagai berikut (Hidayatullah 2017).

$$S_k = \frac{(n_g - 1)}{n} \sum_{j=0}^k n_{rj}$$

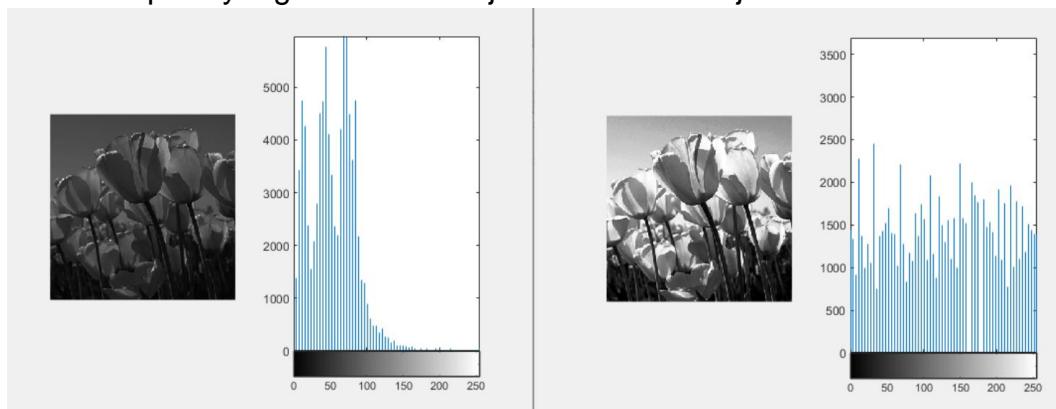
Keterangan:

$n_g$  = Jumlah tingkat keabuan (gray-level) di dalam citra = L.

$n$  = Jumlah seluruh piksel di dalam citra.

$k$  =  $(0, 1, \dots, n_g, -1)$ .

$n_{rj}$  = Jumlah pixel yang memiliki derajat keabuan ke -  $rj$  = ni

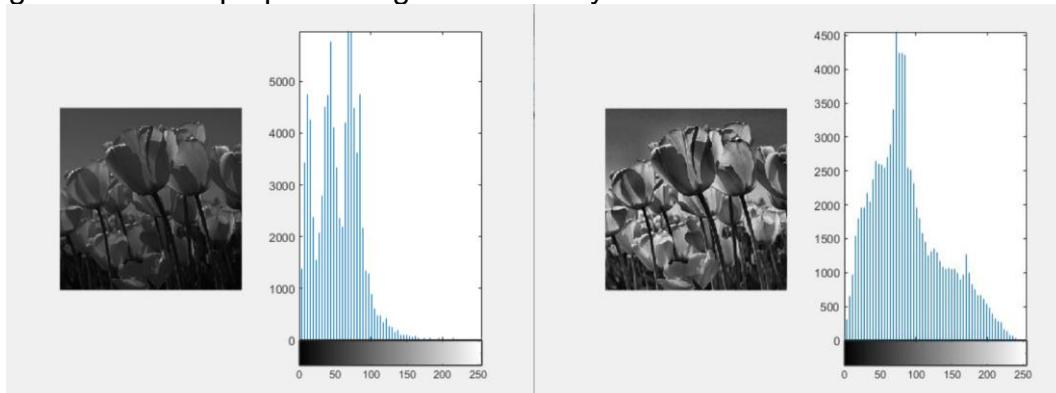


 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 6	<b>Operasi Pengolahan Citra: Operasi Titik Bagian 2</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 80 dari 183

**Gambar 6.3 Ekualisasi Histogram**  
Citra Awal (Kiri) dan Citra Setelah Ekualisasi (Kanan)  
Sumber: dokumen penulis

#### **Ekualisasi Histogram Adaptif**

Pada dasarnya ekualisasi histogram adaptif sama dengan ekualisasi histogram sebelumnya. Hanya saja pada sajekualisasi histogram adaptif citra dibagi menjadi blok-blok (sub-image) dengan ukuran  $n \times n$ , kemudian di setiap blok dilakukan proses ekualisasi histogram. Ukuran blok ( $n$ ) dapat bervariasi dan setiap ukuran blok akan memberikan hasil yang berbeda. Setiap blok dapat saling tumpeng tindih beberapa pixel dengan blok lainnya.



**Gambar 6.4 Ekualisasi Histogram Adaptif:**  
Citra Awal (Kiri) dan Citra Setelah Ekualisasi (Kanan)  
Sumber: dokumen penulis

#### **E. Alat/Instrument/Aparatus/Bahan**

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

#### **F. Keselamatan Kerja**

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 6	Operasi Pengolahan Citra: Operasi Titik Bagian 2		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 81 dari 183

2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporan ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik
9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

## G. Lembar Kerja

### Kegiatan Praktikum 1: Intensity Slicing

Siapkan sebuah gambar dan simpan pada folder projek. Perhatikan lokasi tempat penyimpanan, apakah ada di dalam folder atau tidak. Pada praktikum ini gambar diload dengan format grayscale. Hal ini dimungkinkan pada opencv dengan melakukan set parameter bernilai 0 pada function imread. Pada praktikum sebelumnya tidak dilakukan set nilai sehingga diload gambar berwarna.

```
import cv2
import numpy as np
# Load the image, nilai 0 berarti image dibaca dalam format greyscale
img = cv2.imread('gambar/forest.jpg', 0)
cv2.imshow('Original image', img)
# Cari nilai height dan width
row, column = img.shape
# Buat zeros array untuk menyimpan slice image
img1 = np.zeros((row,column),dtype = 'uint8')
```

Selanjutnya tentukan batas nilai untuk intensitas atas dan bawah. Perulangan dilakukan sesuai input gambar dan jika terdapat nilai pixel yang memenuhi maka set bernilai 0. Jika tidak memenuhi set sesuai nilai pixel aslinya. Berikut kode programnya.

```
# tentukan nilai batas min and max
min_range = 75
max_range = 150

# Perulangan sejumlah input dan jika nilai pixel memenuhi kondisi set ke 0 dan jika tidak set nilai pixel
yang asli
for i in range(row):
```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

## PENGENALAN CITRA DAN POLA

Pertemuan : 6

**Operasi Pengolahan Citra:**  
**Operasi Titik Bagian 2**

100 menit

Kode MK: TIW-043

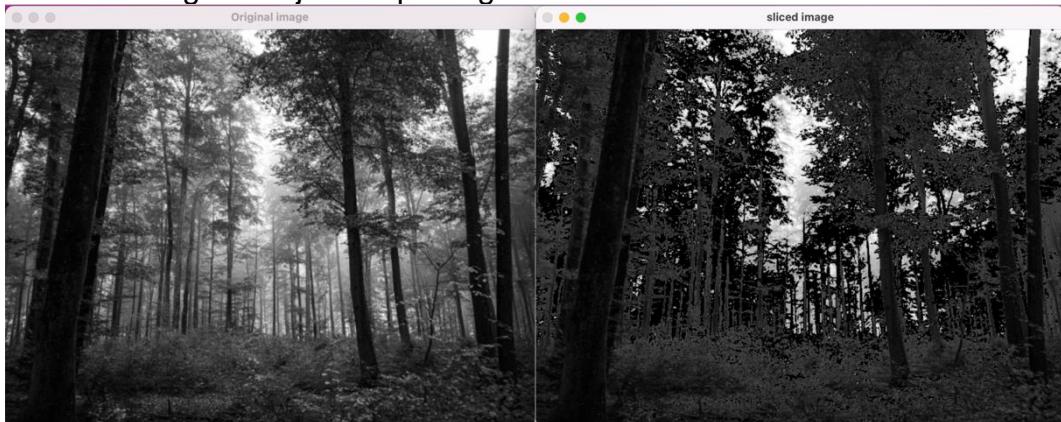
Rev : 00

Tgl : 01/03/2022

Hal 82 dari 183

```
for j in range(column):
    if img[i,j]>min_range and img[i,j]<max_range:
        img1[i,j] = 0
    else:
        img1[i,j] = img[i,j]
# Tampilkan image
cv2.imshow('sliced image', img1)
cv2.waitKey(0)
```

Hasil slice image ditunjukkan pada gambar berikut.



### Kegiatan Praktikum 2: Bit-Plane Slicing

Pada praktikum kedua akan diproses nilai dari bit slicing. Selain menonjolkan range tingkat keabuan tertentu, terkadang diperhatikan juga bagaimana menonjolkan kontribusi dari bit tertentu pada kemunculan citra. Misalkan intensitas tiap piksel dalam citra dinyatakan dengan 8 bit. Sehingga citra tersusun atas 8 bidang 1-bit, mulai dari bidang bit 0 untuk “least significant bit” (LSB) sampai bidang bit 7 untuk “the most significant bit” (MSB). Implementasi dari bit slicing dari citra input dalam format grayscale. Kemudian masing-masing nilai piksel dirubah ke dalam bilangan biner dengan format 8bit. Untuk mengubah bilangan decimal ke biner digunakan fungsi `np.binary_repr()`.

```
import numpy as np
import cv2
# Baca gambar dalam grayscale
img = cv2.imread('gambar/bunga.jpg',0)

#Ulangi setiap piksel dan ubah nilai piksel menjadi biner menggunakan np.binary_repr() dan simpan
dalam array
lst = []
for i in range(img.shape[0]):
    for j in range(img.shape[1]):
```

## PENGENALAN CITRA DAN POLA

Pertemuan : 6

**Operasi Pengolahan Citra:**  
**Operasi Titik Bagian 2**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 83 dari 183

```
Ist.append(np.binary_repr(img[i][j] ,width=8)) # width = no. of bits
```

Setelah didapatkan nilai biner selanjutnya lakukan perulangan untuk mengekstrak nilai bit dan simpan ke dalam sebuah varibel. Hasil rekonstruksi nilai masing-masing bit dilakukan perkalian dengan nilai  $2^{(n-1)}$ . Tampilkan gambar menjadi satu frame dengan menggunakan fungsi `cv.hconcat()`.

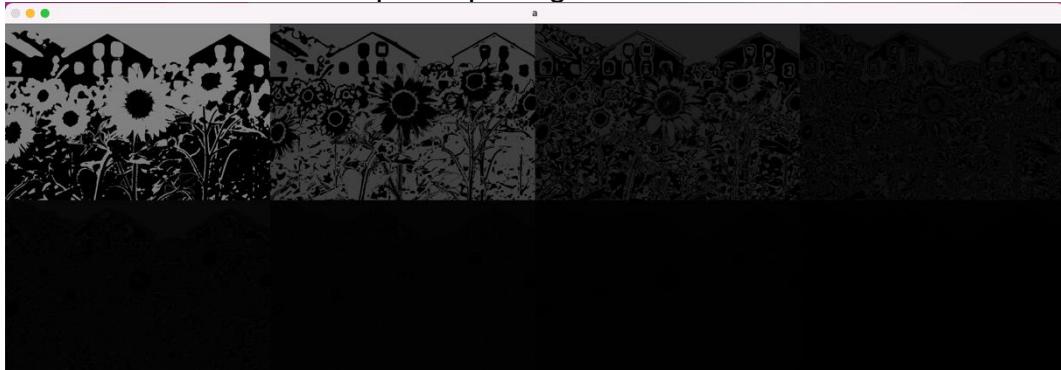
```
# Array menyimpan string di mana setiap string mewakili nilai piksel biner
# Untuk mengekstrak bidang bit, kita perlu mengulangi string dan menyimpan karakter yang sesuai
dengan bidang bit ke dalam daftar
# Kalikan dengan  $2^{(n-1)}$  dan bentuk ulang untuk merekonstruksi gambar bit
eight_bit_img = (np.array([int(i[0]) for i in Ist],dtype = np.uint8) * 128).reshape(img.shape[0],img.shape[1])
seven_bit_img = (np.array([int(i[1]) for i in Ist],dtype = np.uint8) * 64).reshape(img.shape[0],img.shape[1])
six_bit_img = (np.array([int(i[2]) for i in Ist],dtype = np.uint8) * 32).reshape(img.shape[0],img.shape[1])
five_bit_img = (np.array([int(i[3]) for i in Ist],dtype = np.uint8) * 16).reshape(img.shape[0],img.shape[1])
four_bit_img = (np.array([int(i[4]) for i in Ist],dtype = np.uint8) * 8).reshape(img.shape[0],img.shape[1])
three_bit_img = (np.array([int(i[5]) for i in Ist],dtype = np.uint8) * 4).reshape(img.shape[0],img.shape[1])
two_bit_img = (np.array([int(i[6]) for i in Ist],dtype = np.uint8) * 2).reshape(img.shape[0],img.shape[1])
one_bit_img = (np.array([int(i[7]) for i in Ist],dtype = np.uint8) * 1).reshape(img.shape[0],img.shape[1])

#Gabungkan gambar untuk kemudahan tampilan menggunakan cv2.hconcat()
finalr = cv2.hconcat([eight_bit_img,seven_bit_img,six_bit_img,five_bit_img])
finalv = cv2.hconcat([four_bit_img,three_bit_img,two_bit_img,one_bit_img])

# Vertically concatenate
final = cv2.vconcat([finalr,finalv])

# Tampilkan images
cv2.imshow('a',final)
cv2.waitKey(0)
```

Hasil dari ekstraksi bit ditampilkan pada gambar berikut.



### Kegiatan Praktikum 3: Ekualisasi Histogram

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 6	<b>Operasi Pengolahan Citra: Operasi Titik Bagian 2</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 84 dari 183

Ekualisasi Histogram merupakan suatu cara yang bertujuan untuk memperoleh histogram dengan intensitas terdistribusi secara beragam pada citra. Pada library opencv sudah disediakan fungsi dengan nama `cv2.equalizeHist()`. Berikut kode program untuk memproses histogram.

```

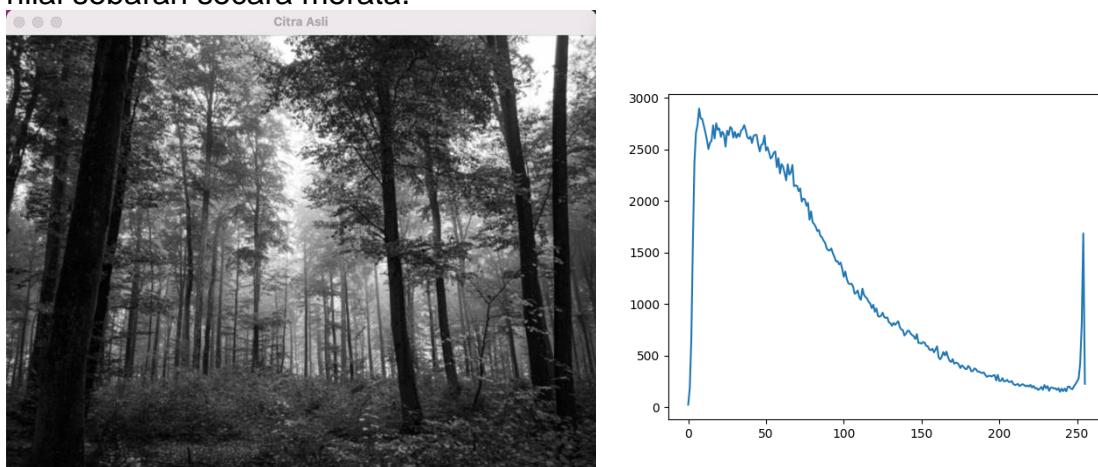
import cv2
import numpy as np
import matplotlib
from matplotlib import pyplot as plt

img = cv2.imread('gambar/forest.jpg',0)
cv2.imshow('Citra Asli', img)
histo_asli = cv2.calcHist([img], [0], None, [256], [0,256])
plt.plot(histo_asli)
plt.show()

img_ekual = cv2.equalizeHist(img)
cv2.imshow('Hasil Ekualisasi Citra', img_ekual)
histo_hasil = cv2.calcHist([img_ekual], [0], None, [256], [0,256])
plt.plot(histo_hasil)
plt.show()
cv2.waitKey(0)

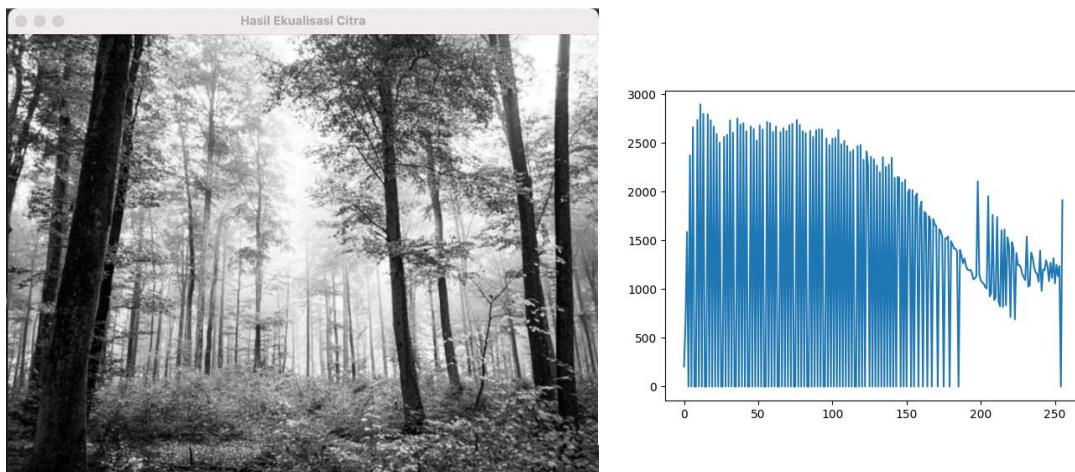
```

Hasil dari proses ekualisasi histogram ditunjukkan pada gambar berikut. Perhatikan pada grafik sebelum dilakukan distribusi dan sesudahnya. Terlihat nilai sebaran secara merata.



Tampilan citra asli dan plot histogramnya

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 6	<b>Operasi Pengolahan Citra: Operasi Titik Bagian 2</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 85 dari 183



Tampilan citra hasil ekualisasi dari histogram dan plot histogramnya

#### Kegiatan Praktikum 4: Adaptif Histogram

Pada kegiatan praktikum 4 akan dibuat Adaptive Histogram Equalization. Teknik ekualisasi histogram adaptif (Adaptive Histogram Equalization, AHE) pada prinsipnya sama dengan ekualisasi histogram. Nama lain dari AHE adalah local histogram processing (Gonzales, 2008), yaitu mengerjakan proses ekualisasi histogram sebanyak beberapa kali masing-masing untuk setiap blok citra (subimage). Ukuran blok citra telah ditentukan sesuai kondisi citra atau kebutuhan penelitian yaitu antara lain 2x2, 4x4, 8x8, 16x16 piksel, atau ukuran yang lain. Pada praktikum ini akan digunakan blok 8x8.

```

import cv2
import numpy as np

# baca gambar input dengan format warna
image = cv2.imread("gambar/forest.jpg")

# ubah menjadi citra grayscale
image_bw = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
equalized = cv2.equalizeHist(image_bw)

# Deklarasi CLAHE
# clipLimit -> Threshold for contrast limiting
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
final_img = clahe.apply(image_bw) + 30

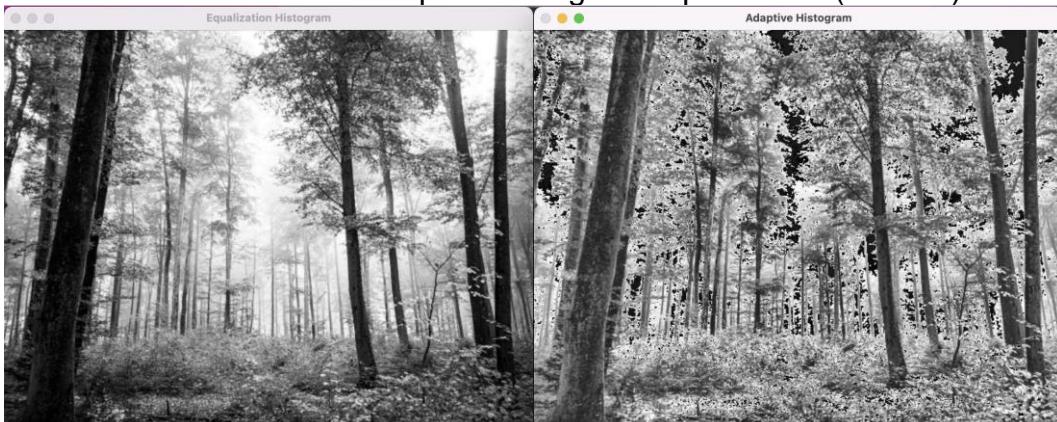
# tampilkan citra hasil
cv2.imshow("Equalization Histogram", equalized)

```

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 6	<b>Operasi Pengolahan Citra: Operasi Titik Bagian 2</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 86 dari 183

```
cv2.imshow("Adaptive Histogram", final_img)
cv2.waitKey(0)
```

Hasil dari adaptif histogram ditunjukkan pada gambar berikut. Gambar sebelah kiri merupakan hasil dari equalization histogram dan sebelah kanan hasil dari metode Contrast Limited Adaptive Histogram Equalization (CLAHE).



## H. Tugas Praktikum

- Carilah sebuah gambar dan simpan di dalam folder project. Lakukan intensity slicing dengan menampilkan rentang intensitas diantara nilai 60 dan 145 dengan memberikan warna putih dan menekan semua intensitas lain menjadi warna hitam atau sebaliknya. Hasil berupa citra biner.

## I. Daftar Referensi

- Hidayatullah, Priyanto. 2017. Pengolahan Citra Digital - Teori dan Aplikasi Nyata. Bandung: Informatika.
- Putra, Darma. 2010. Pengolahan Citra Digital. 1 ed. ed. Westriningsih. Yogyakarta: CV. ANDI OFFSET.

**PETUNJUK PRAKTIKUM #7**

**OPERASI PENGOLAHAN CITRA  
OPERASI SPASIAL**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	<b>Operasi Pengilahan Citra: Operasi Spasial</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 89 dari 183

#### A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer
2. Mampu menerapkan konsep teoritis dalam bidang rekayasa perangkat lunak

#### B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu mengimplementasikan metode-metode image enhancement

#### C. Tujuan

Mahasiswa mampu menjelaskan mengenai konsep image enhancement dan operasi spasial pada citra digital serta menguraikan proses image enhancement berdasarkan operasi spasial seperti kernel atau konvolusi

#### D. Dasar Teori

##### Operasi Spasial (Filtering)

Penapisan pada pengolahan citra biasa disebut dengan pentapisan spasial (spasial filtering). Pada proses pentapisan, nilai pixel baru umumnya dihitung berdasarkan pixel tetangga. Cara perhitungan nilai pixel baru tersebut dapat dikelompokkan menjadi 2, yaitu pertama, pixel baru diperoleh melalui kombinasi linier pixel tetangga dan kedua, pixel baru diperoleh langsung dari salah satu nilai pixel tetangga. Berdasarkan kedua cara tersebut maka tapis juga dapat dikelompokkan menjadi dua yaitu tapis linier (tapis untuk cara pertama) dan tapis nonlinier (tapis untuk cara kedua). Oleh karena itu, penjelasan tentang tapis berikut ini akan dikelompokkan berdasarkan kedua jenis tapis tersebut. Proses penapisan spasial tidak dapat dilepaskan dari teori kernel (mask) dan konvolusi. Untuk itu sebelum membahas proses penapisan akan diawali dengan perbaasan konsep kernel dan konvolusi

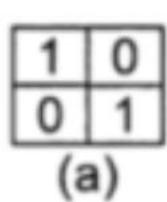
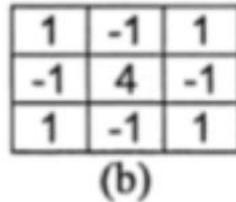
##### Kernel

Kernel adalah matrik yang pada umumnya berukuran kecil dengan elemennya adalah berupa bilangan. Kernel digunakan pada proses konvolusi. Oleh karena itu kernel juga disebut dengan convolution window (jendela konvolusi). Ukuran kernel dapat berbeda-beda, seperti (2 x 2); (3 x 3); (5 x 5); dan sebagainya. Elemen-elemen kernel yang juga disebut bobot (weight) merupakan bilangan-bilangan yang membentuk pola-pola tertentu. Kernel biasa juga disebut dengan tapis, template, mask, serta sliding window. Dalam konsep morphologi kernel

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	Operasi Pengolahan Citra: Operasi Spasial		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 90 dari 183

disebut juga dengan structuring element. Gambar 1 menyajikan contoh kernel (2 x 2) dan (3 x 3).

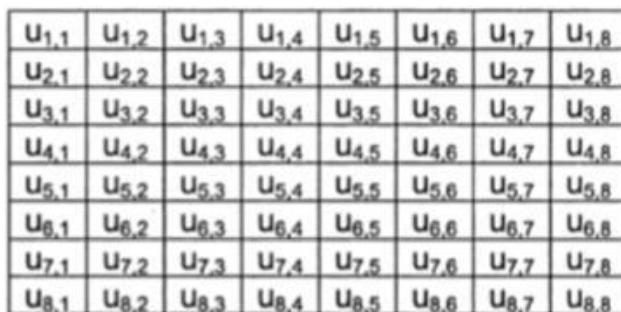
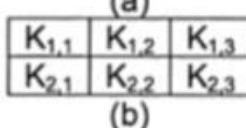
 <b>(a)</b>	 <b>(b)</b>
---	--

Gambar 7.1 Ilustrasi Kernel: 2x2 (a) dan 3x3 (b)

Sumber: Google Images

### Konvolusi (Convolution)

Konvolusi merupakan operator sentral pengolah citra dan telah digunakan secara luas pada berbagai piranti lunak pengolah citra. Proses konvolusi dapat dijalaskan sebagai kernel (sliding window) diletakkan pada setiap pixel dari citra input dan menghasilkan pixel baru. Nilai pixel baru dihitung dengan mengalikan setiap nilai pixel tetangga dengan bobot yang berhubungan pada kernel dan kemudian menjumlah hasil perkalian tersebut. Contoh citra input dan kernel ditunjukkan pada Gambar 7.2.

 <b>(a)</b>	 <b>(b)</b>
--	---

Gambar 7.2 Ilustrasi Pixel Citra Input dan Kernel

Sumber: Google Images

Berdasarkan Gambar 2 maka nilai pixel baru sebagai berikut.

$$O_{4,3} = (u_{4,3} \times K_{1,1}) + (u_{4,4} \times K_{1,2}) + (u_{4,5} \times K_{1,3}) + (u_{5,3} \times K_{2,1}) + (u_{5,4} \times K_{2,2}) + (u_{5,5} \times K_{2,3})$$

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	<b>Operasi Pengolahan Citra: Operasi Spasial</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 91 dari 183

Operator konvolusi biasanya menggunakan tanda ( ). Secara matematika proses konvolusi U•K dapat dinyatakan sebagai berikut.

$$O(i, j) = \sum_{k=1}^m \sum_{l=1}^n U(i+k-1, j+l-1) K(k, l)$$

Dengan  $i = 1 \dots M - m + 1$  dan  $j = 1 \dots N - n + 1$ . M dan N menyatakan ukuran baris dan kolom dari citra Input, sedangkan m dan n menyatakan ukuran baris dan kolom dari kernel. Ukuran citra hasil proses konvolusi di atas adalah  $(M - m + 1)$  baris dan  $(N - n + 1)$  kolom, yang berarti ukuran citra hasil proses konvolusi lebih kecil dari ukuran citra awal.

### Tapis Linier

Tapis linier adalah suatu tapis di mana pixel keluaran adalah kombinasi linier dari pixel masukan.

#### a) Tapis Mean

Tapis mean sangat sederhana dan mudah diimplementasikan serta berguna untuk mengurangi variasi nilai suatu pixel (intensitas) dengan pixel berikutnya. Tapis mean menghitung nilai pixel baru dengan nilai rata-rata dari pixel tetangga dan pixel bersangkutan. Gambar 3 menunjukkan beberapa contoh tapis mean.

$\begin{bmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}$	$\begin{bmatrix} 0 & \frac{1}{8} & 0 \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ 0 & \frac{1}{8} & 0 \end{bmatrix}$
(a)	(b)	(c)

Gambar 7.3 Tapis Mean: 2x2 (a), 3x3 (b), dan 5x5 (c)

Sumber: Google Images

Secara matematis, tapis mean dengan semua pixel tetangga memiliki bobot yang sama, dapat dinyatakan sebagai berikut.

$$O(i, j) = \frac{1}{mn} \sum_{k=1}^m \sum_{l=1}^n U(i+k-1, j+l-1)$$

Dengan  $O(i,j)$  dan  $U(i,j)$  berturut-turut menyatakan citra output dan citra input, m dan n berturut-turut menyatakan ukuran baris dan kolom dari tapis. Untuk tapis mean dengan pixel tetangga tidak memiliki bobot yang sama maka pentapisan dapat dilakukan dengan proses konvolusi. Contoh berikut menunjukkan penerapan tapis mean 3 x 3 pada citra 5 x 5 (Putra 2010).

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	Operasi Pengolahan Citra: Operasi Spasial		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 92 dari 183

$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} * \begin{bmatrix} 1 & 1 & 3 & 3 & 4 \\ 1 & 1 & 4 & 4 & 3 \\ 2 & 1 & 3 & 3 & 3 \\ 1 & 1 & 4 & 4 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 3 & 0 \\ 0 & 2 & 2 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



Gambar 7.4 Konvolusi Citra koala: Citra Awal (Kiri) dan Citra Hasil Konvolusi (Kanan)

Sumber: dokumen penulis

### b) Tapis Gaussian

Gaussian filter atau filter gaussian blur merupakan salah satu teknik pengolahan citra yang mendasar dan banyak digunakan. Gaussian Filter didapat dari operasi konvolusi. Operasi konvolusi yang dilakukan adalah perkalian antara matriks kernel dengan matriks gambar asli. Pada pelaksanaan konvolusi, kernel digeser sepanjang baris dan kolom dalam citra sehingga diperoleh nilai citra yang baru. Tidak seperti metode filtering lainnya yang membutuhkan pengetahuan signal yang lengkap untuk menentukan filter yang optimal. Filter Gaussian adalah salah satu filter linier dengan nilai pembobotan untuk setiap anggotanya dipilih berdasarkan bentuk fungsi Gaussian. Filter Gaussian dipilih sebagai filter penghalusan berdasarkan pertimbangan bahwa filter ini mempunyai pusat kernel. Filter Gaussian sangat baik untuk menghilangkan noise yang bersifat sebaran normal, yang banyak di jumpai pada sebaran citra hasil proses digitasi menggunakan kamera karena merupakan fenomena alamiah akibat sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	<b>Operasi Pengolahan Citra: Operasi Spasial</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 93 dari 183



Gambar 7.5 Filter Gaussian pada Citra koala: Citra Awal (Kiri) dan Citra Hasil Filter (Kanan)

Sumber: dokumen penulis

### c) Tapis Lолос Rendah (Low Pass Filter/LPF)

Tapis ini mempunyai sifat dapat meloloskan yang frekuensi rendah dan menghilangkan yang frekuensi tinggi. Efek yang diberikan pada citra adalah citra tampak lebih lembut (smoothing). Tapis ini digunakan untuk menghaluskan derau atau untuk interpolasi tepi objek dalam citra (Putra 2010).

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \quad \frac{1}{8} \begin{pmatrix} 1 & 3 & 1 \\ 3 & 16 & 3 \\ 1 & 3 & 1 \end{pmatrix}$$

Gambar 7.6 Dua Tipe Tapis Lолос Rendah

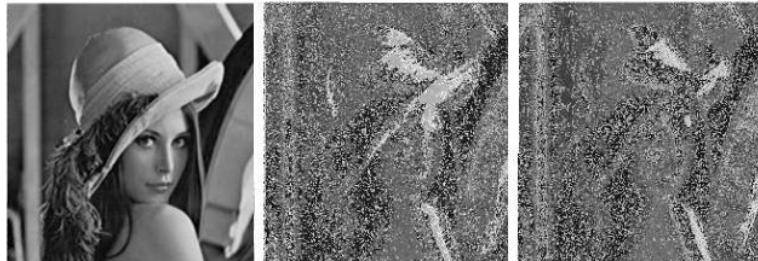
Sumber: Google Images

Tapis ini dalam penggunaannya dapat menggunakan kernel atau tanpa kernel. Tapis lolos rendah yang tidak menggunakan kernel adalah tapis median sedangkan yang menggunakan kernel salah satu contohnya adalah tapis rerata. Kernel (mask) adalah matrik berukuran kecil yang digunakan dengan cara konvolusi pada citra yang ditapis. Kernel tapis lolos rendah memiliki ciri-ciri, yaitu:

- 1) Jumlah semua elemen kernel bernilai satu.
- 2) Elemen kernel tidak ada yang bernilai negatif.
- 3) Tinggi dan lebar kernel ganjil.
- 4) Bobot dalam kernel bersifat simetris terhadap piksel pusat.

Semakin besar ukuran kernel yang digunakan maka citra yang dihasilkan semakin lembut. Gambar 7 memperlihatkan perbandingan hasil penapisan untuk ukuran kernel 3 x 3 dengan 5 x 5 (Sulistiyanti, Setyawan, dan Komarudin 2016).

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	<b>Operasi Pengolahan Citra: Operasi Spasial</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 94 dari 183



Gambar 7.7 Contoh Implementasi Tapis Lolos Rendah:  
Citra Awal (a), dengan Kernel 3x3 (b), dengan Kernel 5x5 (c)  
Sumber: Google Images

#### d) High-pass Filter

Berkebalikan dengan low-pass filtering, high-pass filtering adalah proses filter yang melewatkkan komponen citra dengan nilai intensitas yang tinggi dan meredam komponen citra dengan nilai intensitas yang rendah. High-pass filter akan menyebabkan tepi objek tampak lebih tajam dibandingkan sekitarnya. Aturan kernel untuk high-pass filter adalah koefisien kernel dapat positif, negatif, atau nol dan jumlah semua koefisien kernel adalah nol atau satu. Contoh kernel yang dapat digunakan pada high-pass filter beserta implementasinya sebagai berikut.

## PENGENALAN CITRA DAN POLA

Pertemuan : 7

**Operasi Pengolahan Citra:  
Operasi Spasial**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 95 dari 183

$$(i) \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (ii) \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (iii) \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

$\Sigma = 0$



(a) Citra asli

$\Sigma = 1$



(b) Hasil filtering dengan kernel (i)



(c) Hasil filtering dengan kernel (ii)



(d) Hasil filtering dengan kernel (iii)

**Gambar 7.8 Contoh Implementasi High-Pass Filter**  
Sumber: Google Images

### Tapis Non-Linear

#### a) Tapis Median

Median filter adalah salah satu filtering non-linear yang mengurutkan nilai intensitas sekelompok pixel, kemudian mengganti nilai pixel yang diproses dengan nilai mediannya. Median filter telah digunakan secara luas untuk memperhalus dan mengembalikan bagian dari citra yang mengandung noise yang berbentuk bintik putih.

Metode median filter merupakan filter non-linear yang dikembangkan Tukey, yang berfungsi untuk menghaluskan dan mengurangi noise atau gangguan pada citra. Dikatakan nonlinear karena cara kerja penapis ini tidak termasuk kedalam kategori operasi konvolusi. Operasi nonlinear dihitung dengan mengurutkan nilai intensitas sekelompok pixel, kemudian menggantikan nilai pixel yang diproses dengan nilai tertentu.

## PENGENALAN CITRA DAN POLA

Pertemuan : 7

**Operasi Pengolahan Citra:**  
**Operasi Spasial**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 96 dari 183

Pada median filter suatu window atau penapis yang memuat sejumlah pixel ganjil digeser titik per titik pada seluruh daerah citra. Nilai-nilai yang berada pada window diurutkan secara ascending untuk kemudian dihitung nilai mediannya. Nilai tersebut akan menggantikan nilai yang berada pada pusat bidang window.

Jika suatu window ditempatkan pada suatu bidang citra, maka nilai pixel pada pusat bidang window dapat dihitung dengan mencari nilai median dari nilai intensitas sekelompok pixel yang telah diurutkan. Secara matematis dapat dirumuskan sebagai berikut:

$$g(x, y) = \text{median}\{f(x - i, y - j), (i, j) \in w\}$$

Di mana  $g(x, y)$  merupakan citra yang dihasilkan dari citra  $f(x, y)$  dengan  $w$  sebagai window yang ditempatkan pada bidang citra dan  $(i, j)$  elemen dari window tersebut.



Gambar 7.9 Implementasi Median Filter:  
Citra koala dengan Noise (Kiri) dan Citra Hasil Median Filter (Kanan)  
Sumber: dokumen penulis

### b) Tapis Konservatif

Bila tapis mean menggunakan nilai rata-rata, tapis median menggunakan nilai tengah pixel tetangga dan termasuk pixel bersangkutan, maka tapis konservatif menggunakan nilai minimum dan maksimum pixel tetangga (tidak termasuk pixel yang bersangkutan). Ketentuan untuk menghitung nilai pixel baru adalah bila pixel pusat berada di antara jangkauan nilai minimum dan maksimum maka nilai pixel baru tetap sama dengan nilai sebelumnya. Bila pixel pusat lebih besar dari nilai maksimum maka nilai pixel baru sama dengan nilai maksimum. Bila nilai pixel pusat lebih kecil dari nilai minimum, maka nilai pixel baru adalah sama dengan nilai minimum.

### c) Tapis Kuwahara

Tepi memegang peranan penting dalam analisis citra. Proses penghalusan pada suatu citra dapat mengganggu ketajaman suatu tepi. Tapis Kuwahara dapat

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	Operasi Pengolahan Citra: Operasi Spasial		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 97 dari 183

menghaluskan citra dengan tetap menjaga ketajaman suatu tepi. Tapis Kuwahara bekerja dengan membagi sliding window menjadi 4 area, dan nilai pixel keluaran, yaitu nilai pixel pada pusat koordinat sliding window diganti dengan nilai rata-rata dari area memiliki nilai varian terkecil (Putra 2010).

#### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

#### F. Keselamatan Kerja

1. Sebelum mulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporan ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik
9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

#### G. Lembar Kerja

##### Kegiatan Praktikum 1: Mean dan Gaussian Blur

Siapkan sebuah file baru, copy kode berikut dan coba jalankan.

```
import cv2
import numpy as np

# read the image
image = cv2.imread('gambar/forest.jpg')

# apply the 3x3 mean filter on the image
kernel = np.ones((3,3),np.float32)/9
processed_image = cv2.filter2D(image,-1,kernel)
```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	<b>Operasi Pengolahan Citra: Operasi Spasial</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 98 dari 183

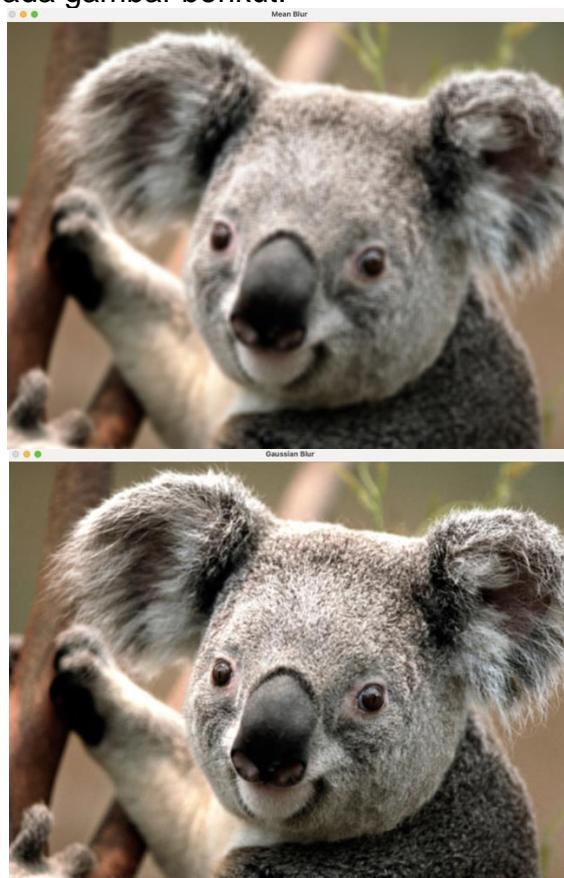
```

mean.blur = cv2.blur(image,(5,5))
gaussian.blur = cv2.GaussianBlur(image,(5,5),0)

# display image
cv2.imshow('Image Original', image)
cv2.imshow('Mean Filter Processing', processed_image)
cv2.imshow('Mean Blur', mean.blur)
cv2.imshow('Gaussian Blur', gaussian.blur)
# pause the execution of the script until a key on the keyboard is pressed
cv2.waitKey(0)

```

Hasil ditunjukkan pada gambar berikut:



### Kegiatan Praktikum 2: Low Pass dan High Pass Filtering Low Pass Filtering

Siapkan sebuah file baru, copy kode berikut dan coba jalankan.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	<b>Operasi Pengolahan Citra: Operasi Spasial</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 99 dari 183

```

import cv2
import numpy as np
from matplotlib import pyplot as plt
from cv2 import waitKey

def lowPassFiltering(img,size):
    h, w = img.shape[0:2]
    h1,w1 = int(h/2), int(w/2)
    img2 = np.zeros((h, w), np.uint8)
    img2[h1-int(size/2):h1+int(size/2), w1-int(size/2):w1+int(size/2)] = 1
    img3=img2*img
    return img3

gray = cv2.imread("gambar/koala.jpg", 1)
gray = cv2.cvtColor(gray, cv2.COLOR_BGR2GRAY)
gray = cv2.resize(gray, (1280, 720))

h,w =gray.shape

for i in range(3000): #Add 3000 Noise Points
    x = np.random.randint(0, h)
    y = np.random.randint(0, w)
    gray[x,y] = 255

# Fourier transform
img_dft = np.fft.fft2(gray)
dft_shift = np.fft.fftshift(img_dft)

# Low-pass filter
dft_shift = lowPassFiltering(dft_shift, 200)
res = np.log(np.abs(dft_shift))

# Inverse Fourier Transform
idft_shift = np.fft.ifftshift(dft_shift)
ifimg = np.fft.ifft2(idft_shift)
ifimg = np.abs(ifimg)
cv2.imshow("ifimg", np.int8(ifimg))
cv2.imshow("gray", gray)

# Draw pictures

```

# PENGENALAN CITRA DAN POLA

<b>INSTIKI</b> INSTITUT BISNIS DAN TEKNOLOGI INDONESIA	Pertemuan : 7	<b>Operasi Pengolahan Citra: Operasi Spasial</b>		100 menit
	Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 100 dari 183

```
plt.subplot(131), plt.imshow(gray, 'gray'), plt.title('Original Image')
plt.axis('off')
plt.subplot(132), plt.imshow(res, 'gray'), plt.title('Low-pass filter')
plt.axis('off')
plt.subplot(133), plt.imshow(np.int8(ifimg), 'gray'), plt.title('Effect after filtering')
plt.axis('off')
plt.show()
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Hasil untuk pemrosesan low pass filtering ditunjukan pada gambar berikut.



Figure 1



Original Image      Low-pass filter      Effect after filtering

## High Pass Filtering

Siapkan sebuah file baru, copy kode berikut dan coba jalankan.

```
import cv2  
import numpy as np  
from matplotlib import pyplot as plt  
from cv2 import waitKey
```

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	<b>Operasi Pengolahan Citra: Operasi Spasial</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 101 dari 183

```

def highPassFiltering(img,size):
    h, w = img.shape[0:2]
    h1,w1 = int(h/2), int(w/2)
    img[h1-int(size/2):h1+int(size/2), w1-int(size/2):w1+int(size/2)] = 0
    return img

gray = cv2.imread("gambar/koala.jpg", 1)
gray = cv2.cvtColor(gray, cv2.COLOR_BGR2GRAY)
gray = cv2.resize(gray, (640, 420))

# Fourier transform
img_dft = np.fft.fft2(gray)
dft_shift = np.fft.fftshift(img_dft) # Move frequency domain from upper left to middle

#High pass filter
dft_shift=highPassFiltering(dft_shift,200)
res = np.log(np.abs(dft_shift))

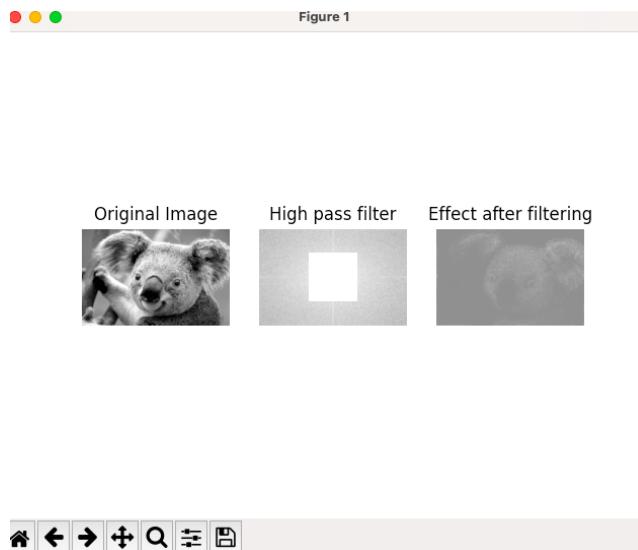
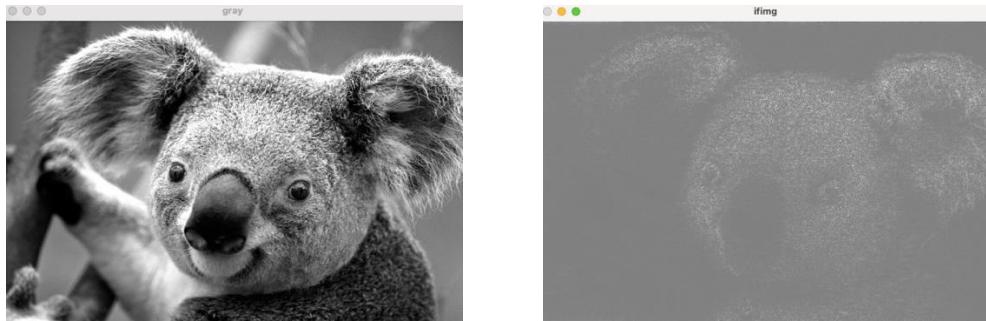
# Inverse Fourier Transform
idft_shift = np.fft.ifftshift(dft_shift) #Move the frequency domain from the middle to the upper left corner
ifimg = np.fft.ifft2(idft_shift) # Fourier library function call
ifimg = np.abs(ifimg)
cv2.imshow("ifimg",np.int8(ifimg))
cv2.imshow("gray",gray)

# Draw pictures
plt.subplot(131), plt.imshow(gray, 'gray'), plt.title('Original Image')
plt.axis('off')
plt.subplot(132), plt.imshow(res, 'gray'), plt.title('High pass filter')
plt.axis('off')
plt.subplot(133), plt.imshow(np.int8(ifimg), 'gray'), plt.title('Effect after filtering')
plt.axis('off')
plt.show()
cv2.waitKey(0)
cv2.destroyAllWindows()

```

Hasil dari proses high pass filtering ditunjukkan pada gambar berikut.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	<b>Operasi Pengolahan Citra: Operasi Spasial</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 102 dari 183



### Kegiatan Praktikum 3: Median Filter

Siapkan sebuah file baru, copy kode berikut dan coba jalankan.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('gambar/koala.jpg')

blur = cv2.medianBlur(img,9)

plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(blur),plt.title('Blurred')
plt.xticks([]), plt.yticks([])
plt.show()
```

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 7	<b>Operasi Pengolahan Citra: Operasi Spasial</b>		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 103 dari 183



## H. Tugas Praktikum

1. Buatlah program perbaikan citra dengan menggunakan mean dan gaussian filter dengan menggunakan kernel ukuran 3x3, 5x5, 7x7. Lakukan analisa dan lihat perbedaan yang terjadi!
2. Buatlah program untuk memperbaiki citra dengan menggunakan median filter menggunakan ukuran 5x5 dan 7x7 pada citra berwarna.

## I. Daftar Referensi

- Putra, Darma. 2010. Pengolahan Citra Digital. 1 ed. ed. Westriningsih. Yogyakarta: CV. ANDI OFFSET.
- Sulistiyantri, Sri Ratna, FX Arinto Setyawan, dan Muhamad Komarudin. 2016. Pengolahan Citra: Dasar dan Contoh Penerapannya. 1 ed. Yogyakarta: TEKNOSAIN. <http://repository.lppm.unila.ac.id/2976/2/05-Buku Ajar Pengolahan Citra.pdf>.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p>		
	<b>PENGENALAN CITRA DAN POLA</b>		
	Pertemuan : 7	<b>Operasi Pengolahan Citra: Operasi Spasial</b>	100 menit
	Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022

## PENGOLAHAN CITRA DAN POLA 104

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

## **PETUNJUK PRAKTIKUM #8**

### **OPERASI MORPHOLOGY**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 8	<b>Image Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 106 dari 183

### A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer
2. Mampu menerapkan konsep teoritis dalam bidang rekayasa perangkat lunak

### B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu memahami dan mengaplikasikan metode-metode image morphology

### C. Tujuan

Tujuan mahasiswa mampu menjelaskan mengenai konsep image morphology, menjabarkan jenis operasi dasar image morphology seperti dilasi, erosi, opening, closing, serta mampu menerapkan algoritma image morphology.

### D. Dasar Teori

Morfologi merupakan teknik pengolahan citra berdasarkan bentuk segmen citra yang bertujuan untuk memperbaiki hasil segmentasi. Teknik morfologi biasanya digunakan pada citra biner atau untuk beberapa kasus juga bisa diterapkan pada citra keabuan (grayscale).

#### Operasi Dasar Morfologi

##### Elemen Penstruktur (Structuring Element)

Elemen penstruktur berpengaruh terhadap hasil dari sebuah operasi morfologi. Pemilihan elemen penstruktur yang tidak tepat akan mengakibatkan kegagalan tercapainya hasil citra yang diinginkan. Beberapa elemen penstruktur diperlihatkan pada Gambar 1.

1	1	1
1	1	1
1	1	1

0	1	0
1	1	1
0	1	0

1	0	1
0	1	0
1	0	1

0	1	0
0	1	0
0	1	0

0	0	0
1	1	1
0	0	0

0	0	1
0	1	0
1	0	0

1	0	0
0	1	0
0	0	1

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h2>PENGENALAN CITRA DAN POLA</h2>		
Pertemuan : 8	<b>Image Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 107 dari 183

Gambar 8.1 Berbagai Elemen Penstruktur

Sumber: Google Images

### Dilasi

Operasi dilasi adalah operasi penebalan objek yang terdapat pada citra biner. Operasi dilasi dilakukan dengan menambah pixel pada kontur dari objek sesuai dengan elemen penstruktur. Berikut adalah persamaan yang digunakan untuk operasi dilasi.

$$A \oplus B = t \in z^2; t = a + b, a \in A, b \in B$$



Gambar 8.2 Operasi Dilasi pada Citra koala: Citra Awal (Kiri)

Citra Setelah Dilasi (Kanan)

Sumber: dokumen penulis

### Erosi

Operasi erosi adalah operasi penipisan objek yang terdapat pada citra biner. Operasi erosi dilakukan dengan mengurangi pixel pada kontur dari objek, sesuai dengan elemen penstrukturnya. Persamaan dari operasi erosi adalah sebagai berikut.

$$A \ominus B = A^c \oplus B^c$$

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 8	<b>Image Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 108 dari 183



Gambar 8.3 Operasi Erosi pada Citra koala: Citra Awal (Kiri)

Citra Setelah Dilasi (Kanan)

Sumber: dokumen penulis

### Opening

Operasi opening adalah operasi erosi yang dilanjutkan dengan operasi dilasi. Operasi opening pada citra mempunyai efek memperhalus batas-batas objek, memisahkan objek-objek yang sebelumnya bergandengan, dan menghilangkan objek-objek yang lebih kecil daripada ukuran structuring. Berikut adalah persamaan pada operasi morfologi opening.

$$A \circ B = A \ominus B \oplus B$$

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 8	<b>Image Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 109 dari 183



Gambar 8.4 Operasi Opening pada Citra koala:  
Citra Awal (Kiri) dan Citra Setelah Dilasi (Kanan)

Sumber: dokumen penulis

### Closing

Morfologi closing merupakan kebalikan dari morfologi opening dimana pada perasi ini, citra diolah dari dilasi yang kemudian dilanjutkan dengan operasi erosi. Operasi closing juga cenderung akan memperhalus objek pada citra, namun dengan cara menyambung pecahan-pecahan dan menghilangkan lubang-lubang kecil pada objek. Berikut adalah persamaan dari operasi closing.

$$A \cdot B = A \oplus B \ominus B$$

## PENGENALAN CITRA DAN POLA

Pertemuan : 8

Image Morphology

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 110 dari 183



Gambar 8.5 Operasi Closing pada Citra koala:  
Citra Awal (Kiri) dan Citra Setelah Dilasi (Kanan)

Sumber: dokumen penulis

### Hit or Miss

Transformasi hit or miss adalah operasi morfologi biner umum yang dapat digunakan untuk mencari pola pixel latar depan dan latar belakang tertentu dalam gambar. Sebenarnya ini adalah operasi dasar dari morfologi biner karena hampir semua operator morfologi biner lainnya dapat diturunkan darinya. Seperti operator morfologi biner lainnya, hit or miss mengambil citra biner dan elemen penataan sebagai masukan, dan menghasilkan citra biner lain sebagai keluaran.

Hit or Miss Transform ( $A^*S$ ) adalah kumpulan titik-titik dimana  $S_1$  menemukan match di  $A$  dan pada saat yang bersamaan  $S_2$  juga menemukan match di luar  $A$ . Persamaan dari hit or miss transform sebagai berikut.

$$A^* S = (A \Theta S_1) \cap (A^c \Theta S_2)$$

### Algoritma Dasar Morphology

#### Ekstraksi Batas (Boundary Extraction)

Dalam citra biner selalu melihat ada keterhubungan nilai pixel yang satu dandengan tetangganya di mana sebuah objek akan memiliki nilai pixel 1. Pixel batas merupakan sebuah objek dimana dalam ketetanggaan  $3 \times 3$  yang berarti 8 pixel terdapat setidaknya satu pixel bernilai 0. Operasi pada ekstraksi batas diawali dengan operasi erosi. Persamaan dalam operasi boundary extraction adalah sebagai berikut (Hidayatullah 2017).

## PENGENALAN CITRA DAN POLA

Pertemuan : 8

Image Morphology

100 menit

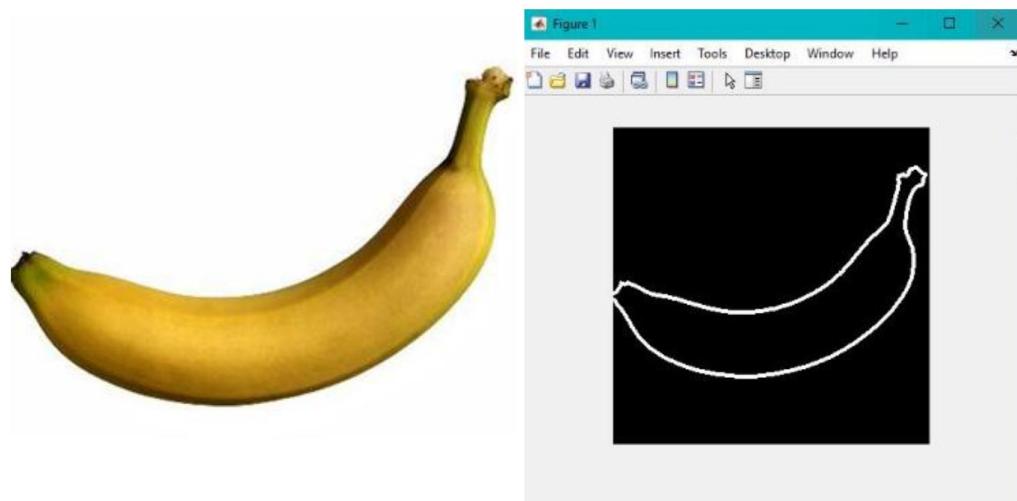
Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 111 dari 183

$$\partial A = A - A \ominus B$$



Gambar 8.6 Implementasi Ekstraksi Batas pada Citra banana:

Citra Awal (Kiri) dan Citra Hasil Ekstraksi (Kanan)

Sumber: dokumen penulis

### Region Filling

Dimulai dari satu titik  $p$  di dalam boundary, tujuannya adalah mengisi keseluruhan region dengan 1. Jika digunakan konvensi bahwa semua titik non-boundary dilabeli 0, maka kita memberi nilai 1 pada  $p$  untuk memulai.



Gambar 8.7 Contoh Implementasi Region Filling

Sumber: dokumen penulis

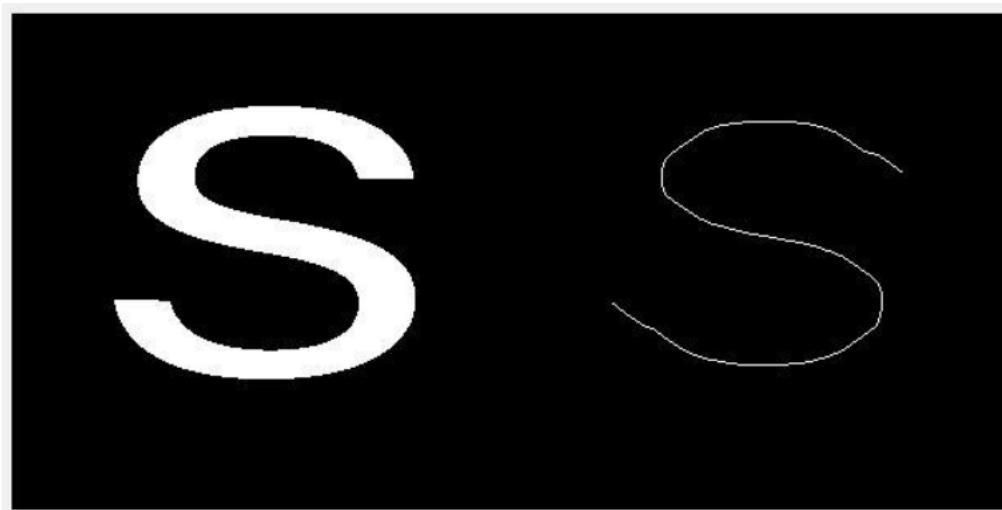
### Penipisan (Thinning)

Operasi thinning (penipisan) sama seperti operasi erosi tetapi tidak membuat objek menjadi menghilang. Operasi thinning akan mereduksi menjadi hanya rangka yang mendekati garis sumbu objek dan bertujuan untuk mengurangi bagian yang tidak perlu sehingga tebal objek tersebut hanya menjadi satu pixel. Salah satu

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 8	<b>Image Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 112 dari 183

implementasi dari operasi thinning adalah untuk melakukan identifikasi dari huruf / karakter (Hidayatullah 2017). Operasi thinning dapat direpresentasikan dalam persamaan berikut.

$$A \otimes B = A - (A * B) = A \cap (A * B)^c$$



Gambar 8.8 Contoh Implementasi Thinning  
Sumber: dokumen penulis

### Penebalan (Thickening)

Operasi thickening sama seperti operasi dilasi tetapi pada operasi ini tidak akan menyebabkan tergabungnya objek yang terpisah. Operasi thickening bisa disebut juga operasi dual dari thinning (Hidayatullah 2017). Operasi thickening dapat direpresentasikan dalam persamaan berikut.

$$A \bullet B = A \cup (A * B)$$

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 8	<b>Image Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 113 dari 183



Gambar 8.9 Contoh Implementasi Thickening  
 Sumber: dokumen penulis

#### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

#### F. Keselamatan Kerja

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporan ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik
9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 8	<b>Image Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 114 dari 183

## G. Lembar Kerja

### Kegiatan Praktikum 1: Operasi Dasar Morfologi

#### Dilasi

Siapkan sebuah file baru dan gambar, copy kode berikut dan coba jalankan.

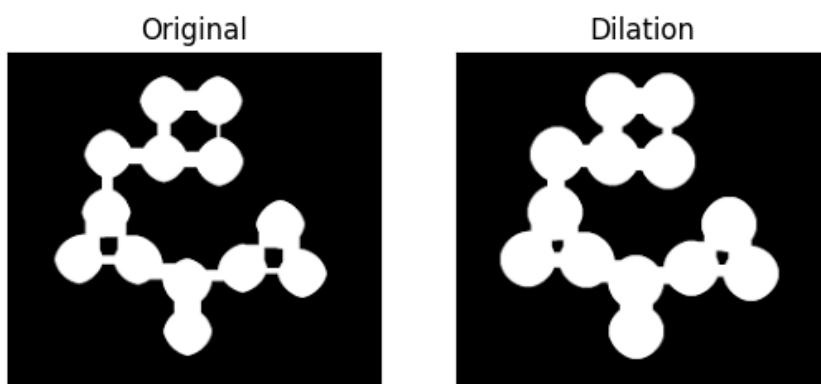
```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('gambar/test_image.jpg')

kernel = np.ones((5,5),np.uint8)
dilation = cv2.dilate(img,kernel,iterations = 1)

plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(dilation),plt.title('Dilation')
plt.xticks([]), plt.yticks([])
plt.show()
```

Hasil ditunjukkan pada gambar berikut. Terlihat adanya peningkatan wilayah putih dalam gambar.



#### Erosi

Siapkan sebuah file baru dan gambar, copy kode berikut dan coba jalankan.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('gambar/forest.jpg')

kernel = np.ones((5,5),np.uint8)
erosion = cv2.erode(img,kernel,iterations = 1)
```

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

## PENGENALAN CITRA DAN POLA

Pertemuan : 8

Image Morphology

100 menit

Kode MK: TIW-043

Rev : 00

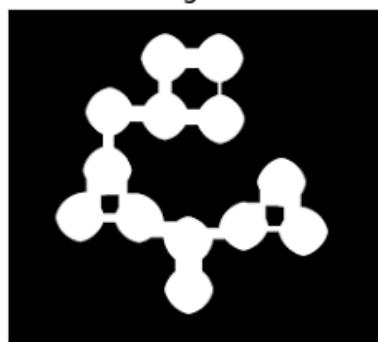
Tgl : 01/03/2022

Hal 115 dari 183

```
plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(erosion),plt.title('erosion')
plt.xticks([]), plt.yticks([])
plt.show()
```

Hasil ditunjukkan pada gambar berikut. Terlihat adanya pengikisan batas-batas objek latar depan (Selalu mencoba untuk menjaga latar depan putih).

Original



erosion



### Opening

Siapkan sebuah file baru dan gambar, copy kode berikut dan coba jalankan.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('gambar/forest.jpg')

kernel = np.ones((5,5),np.uint8)
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)

plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(opening),plt.title('Opening')
plt.xticks([]), plt.yticks([])
plt.show()
```

Hasil ditunjukkan pada gambar berikut. Proses opening hanyalah nama lain dari erosi diikuti oleh pelebaran. Ini berguna untuk menghilangkan noise. Fungsi yang digunakan adalah **cv2.morphologyEx()**.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 8	<b>Image Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 116 dari 183



### Closing

Siapkan sebuah file baru dan gambar, copy kode berikut dan coba jalankan.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('gambar/forest.jpg')

kernel = np.ones((5,5),np.uint8)
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)

plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(closing),plt.title('Closing')
plt.xticks([]), plt.yticks([])
plt.show()
```

Hasil ditunjukkan pada gambar berikut. Proses Closing adalah kebalikan dari Pembukaan (Opening), Pelebaran diikuti oleh Erosi. Hal ini berguna dalam menutup lubang kecil di dalam objek latar depan, atau titik hitam kecil pada objek.



### Kegiatan Praktikum 2: Region Filling

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

## PENGENALAN CITRA DAN POLA

Pertemuan : 8

**Image Morphology**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 117 dari 183

Siapkan sebuah file baru dan gambar, copy kode berikut dan coba jalankan.

```
import cv2;
import numpy as np;

# Read image
im_in = cv2.imread("gambar/water_coins.jpeg", cv2.IMREAD_GRAYSCALE);

# th, im_th = cv2.threshold(im_in, 220, 255, cv2.THRESH_BINARY_INV);
th, im_th = cv2.threshold(im_in, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)

# Copy the thresholded image.
im_floodfill = im_th.copy()

# Mask used to flood filling.
# Notice the size needs to be 2 pixels than the image.
h, w = im_th.shape[:2]
mask = np.zeros((h+2, w+2), np.uint8)

# Floodfill from point (0, 0)
cv2.floodFill(im_floodfill, mask, (0,0), 255);

# Invert floodfilled image
im_floodfill_inv = cv2.bitwise_not(im_floodfill)

# Combine the two images to get the foreground.
im_out = im_th | im_floodfill_inv

# Display images.
cv2.imshow("Thresholded Image", im_th)
cv2.waitKey(0)
cv2.imshow("Floodfilled Image", im_floodfill)
cv2.waitKey(0)
cv2.imshow("Inverted Floodfilled Image", im_floodfill_inv)
cv2.waitKey(0)
cv2.imshow("Foreground", im_out)
cv2.waitKey(0)
```

Hasil dari proses ini ditunjukkan pada gambar berikut. Proses dimulai dari membaca gambar. Lalu tentukan ambang batas gambar input untuk mendapatkan gambar biner. Copy gambar dan set nilai piksel bernilai 0 semua. Gabungkan gambar yang diberi ambang batas dengan gambar invert menggunakan operasi bitwise OR untuk mendapatkan mask latar depan akhir dengan lubang yang terisi.

## PENGENALAN CITRA DAN POLA

Pertemuan : 8

Image Morphology

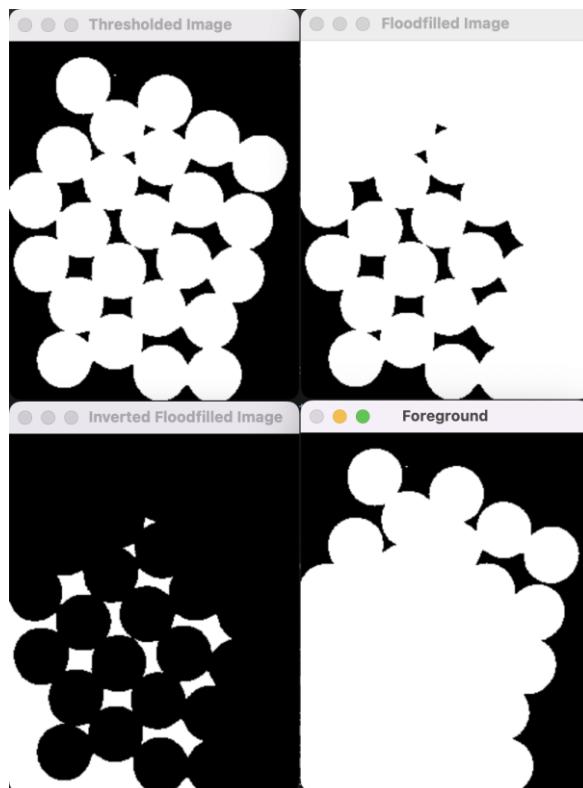
100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 118 dari 183



### Kegiatan Praktikum 3: Penipisan/Thinning

Siapkan sebuah file baru dan gambar, copy kode berikut dan coba jalankan.

```
Import cv2
import numpy as np

# Read the image as a grayscale image
img = cv2.imread('gambar/bunga.jpg',0)

# Threshold the image
ret,img = cv2.threshold(img, 127, 255, 0)

# Step 1: Create an empty skeleton
size = np.size(img)
skel = np.zeros(img.shape[:2], np.uint8)

# Get a Cross Shaped Kernel
element = cv2.getStructuringElement(cv2.MORPH_CROSS, (3,3))

# Repeat steps 2-4
while True:
```

## PENGENALAN CITRA DAN POLA

Pertemuan : 8

**Image Morphology**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 119 dari 183

```
#Step 2: Open the image
open = cv2.morphologyEx(img, cv2.MORPH_OPEN, element)
#Step 3: Subtract open from the original image
temp = cv2.subtract(img, open)
#Step 4: Erode the original image and refine the skeleton
eroded = cv2.erode(img, element)
skel = cv2.bitwise_or(skel,temp)
img = eroded.copy()
# Step 5: If there are no white pixels left ie.. the image has been completely eroded, quit the loop
if cv2.countNonZero(img)==0:
    break

# Displaying the final skeleton
cv2.imshow("Skeleton",skel)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Hasil dari proses ini ditunjukkan pada gambar berikut. Proses dimulai dari membaca gambar.



### H. Tugas Praktikum

Carilah sebuah gambar lalu lakukan proses penebalan atau thickening pada gambar tersebut. Tampilkan gambar sebelum dan sesudah di proses untuk melihat perbedaannya!

### I. Daftar Referensi

- Hidayatullah, Priyanto. 2017. Pengolahan Citra Digital - Teori dan Aplikasi Nyata. Bandung: Informatika.

## **PENGENALAN CITRA DAN POLA**

Pertemuan : 9

**Operasi Morphology**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 0 dari 183

### **PETUNJUK PRAKTIKUM #9**

### **OPERASI MORPHOLOGY** **Bagian 2**

**PENGOLAHAN CITRA DAN POLA** 0

Dibuat oleh :

Dilarang memperbanyak sebagian atau seluruh isi dokumen  
tanpa ijin tertulis dari Program Studi Teknik Informatika  
Institut Bisnis dan Teknologi Indonesia (INSTIKI)

Diperiksa oleh :

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 9	<b>Operasi Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 121 dari 183

### A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer
2. Mampu menerapkan konsep teoritis dalam bidang rekayasa perangkat lunak

### B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu memahami dan mengaplikasikan metode-metode image morphology

### C. Tujuan

Tujuan mahasiswa mampu menjelaskan mengenai konsep image morphology, menjelaskan jenis operasi dasar image morphology seperti dilasi, erosi, opening, closing, serta mampu menerapkan algoritma image morphology.

### D. Dasar Teori

Morfologi merupakan teknik pengolahan citra berdasarkan bentuk segmen citra yang bertujuan untuk memperbaiki hasil segmentasi. Teknik morfologi biasanya digunakan pada citra biner atau untuk beberapa kasus juga bisa diterapkan pada citra keabuan (grayscale).

#### Operasi Dasar Morfologi

##### Elemen Penstruktur (Structuring Element)

Elemen penstruktur berpengaruh terhadap hasil dari sebuah operasi morfologi. Pemilihan elemen penstruktur yang tidak tepat akan mengakibatkan kegagalan tercapainya hasil citra yang diinginkan. Beberapa elemen penstruktur diperlihatkan pada Gambar 1.

1	1	1
1	1	1
1	1	1

0	1	0
1	1	1
0	1	0

1	0	1
0	1	0
1	0	1

0	1	0
0	1	0
0	1	0

0	0	0
1	1	1
0	0	0

0	0	1
0	1	0
1	0	0

1	0	0
0	1	0
0	0	1

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 9	<b>Operasi Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 122 dari 183

Gambar 8.1 Berbagai Elemen Penstruktur

Sumber: Google Images

### Dilasi

Operasi dilasi adalah operasi penebalan objek yang terdapat pada citra biner. Operasi dilasi dilakukan dengan menambah pixel pada kontur dari objek sesuai dengan elemen penstruktur. Berikut adalah persamaan yang digunakan untuk operasi dilasi.

$$A \oplus B = t \in z^2; t = a + b, a \in A, b \in B$$



Gambar 8.2 Operasi Dilasi pada Citra koala: Citra Awal (Kiri)

Citra Setelah Dilasi (Kanan)

Sumber: dokumen penulis

### Erosi

Operasi erosi adalah operasi penipisan objek yang terdapat pada citra biner. Operasi erosi dilakukan dengan mengurangi pixel pada kontur dari objek, sesuai dengan elemen penstrukturnya. Persamaan dari operasi erosi adalah sebagai berikut.

$$A \ominus B = A^c \oplus B^c$$

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 9	<b>Operasi Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 123 dari 183



Gambar 8.3 Operasi Erosi pada Citra koala: Citra Awal (Kiri) dan  
Citra Setelah Dilasi (Kanan)  
Sumber: dokumen penulis

### Opening

Operasi opening adalah operasi erosi yang dilanjutkan dengan operasi dilasi. Operasi opening pada citra mempunyai efek memperhalus batas-batas objek, memisahkan objek-objek yang sebelumnya bergandengan, dan menghilangkan objek-objek yang lebih kecil daripada ukuran structuring. Berikut adalah persamaan pada operasi morfologi opening.

$$A \circ B = A \ominus B \oplus B$$

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 9	<b>Operasi Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 124 dari 183



Gambar 8.4 Operasi Opening pada Citra koala:  
Citra Awal (Kiri) dan Citra Setelah Dilasi (Kanan)

Sumber: dokumen penulis

### Closing

Morfologi closing merupakan kebalikan dari morfologi opening dimana pada perasi ini, citra diolah dari dilasi yang kemudian dilanjutkan dengan operasi erosi. Operasi closing juga cenderung akan memperhalus objek pada citra, namun dengan cara menyambung pecahan-pecahan dan menghilangkan lubang-lubang kecil pada objek. Berikut adalah persamaan dari operasi closing.

$$A \cdot B = A \oplus B \ominus B$$

## PENGENALAN CITRA DAN POLA

Pertemuan : 9

Operasi Morphology

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 125 dari 183



Gambar 8.5 Operasi Closing pada Citra koala:  
Citra Awal (Kiri) dan Citra Setelah Dilasi (Kanan)

Sumber: dokumen penulis

### Hit or Miss

Transformasi hit or miss adalah operasi morfologi biner umum yang dapat digunakan untuk mencari pola pixel latar depan dan latar belakang tertentu dalam gambar. Sebenarnya ini adalah operasi dasar dari morfologi biner karena hampir semua operator morfologi biner lainnya dapat diturunkan darinya. Seperti operator morfologi biner lainnya, hit or miss mengambil citra biner dan elemen penataan sebagai masukan, dan menghasilkan citra biner lain sebagai keluaran.

Hit or Miss Transform ( $A^*S$ ) adalah kumpulan titik-titik dimana  $S_1$  menemukan match di  $A$  dan pada saat yang bersamaan  $S_2$  juga menemukan match di luar  $A$ . Persamaan dari hit or miss transform sebagai berikut.

$$A^* S = (A \Theta S_1) \cap (A^c \Theta S_2)$$

### Algoritma Dasar Morphology

#### Ekstraksi Batas (Boundary Extraction)

Dalam citra biner selalu melihat ada keterhubungan nilai pixel yang satu dandengan tetangganya di mana sebuah objek akan memiliki nilai pixel 1. Pixel batas merupakan sebuah objek dimana dalam ketetanggaan  $3 \times 3$  yang berarti 8 pixel terdapat setidaknya satu pixel bernilai 0. Operasi pada ekstraksi batas diawali dengan operasi erosi. Persamaan dalam operasi boundary extraction adalah sebagai berikut (Hidayatullah 2017).

## PENGENALAN CITRA DAN POLA

Pertemuan : 9

Operasi Morphology

100 menit

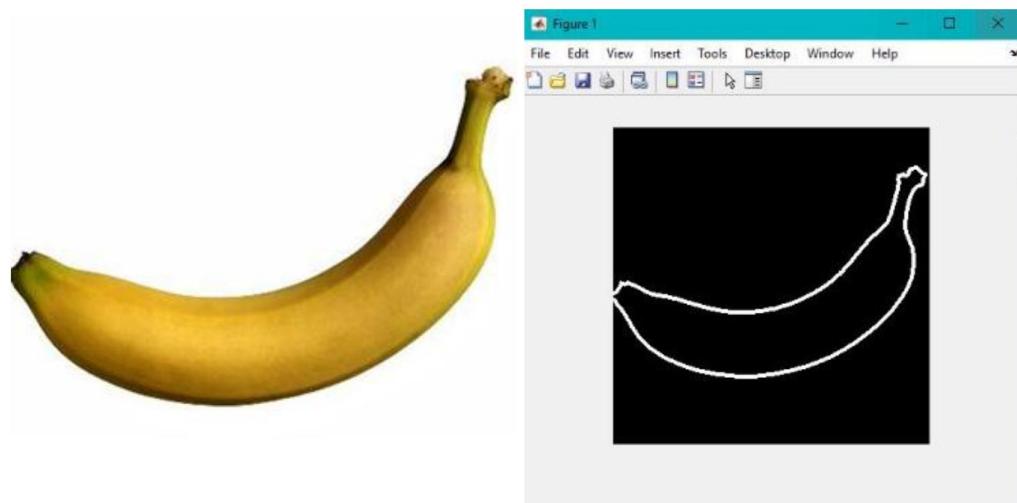
Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 126 dari 183

$$\partial A = A - A \ominus B$$



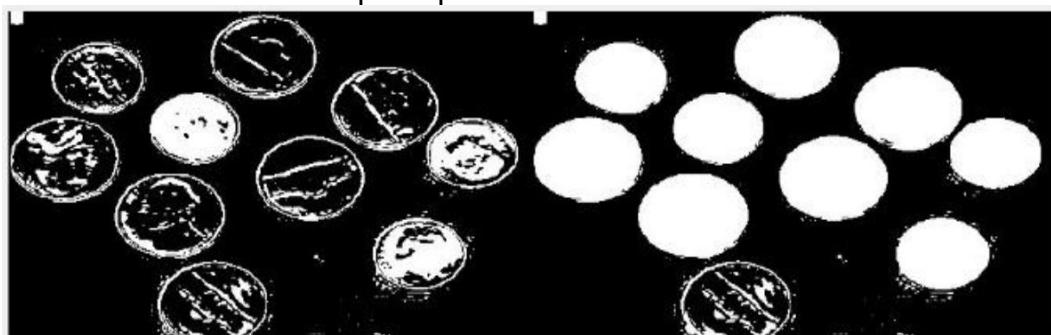
Gambar 8.6 Implementasi Ekstraksi Batas pada Citra banana:

Citra Awal (Kiri) dan Citra Hasil Ekstraksi (Kanan)

Sumber: dokumen penulis

### Region Filling

Dimulai dari satu titik  $p$  di dalam boundary, tujuannya adalah mengisi keseluruhan region dengan 1. Jika digunakan konvensi bahwa semua titik non-boundary dilabeli 0, maka kita memberi nilai 1 pada  $p$  untuk memulai.



Gambar 8.7 Contoh Implementasi Region Filling

Sumber: dokumen penulis

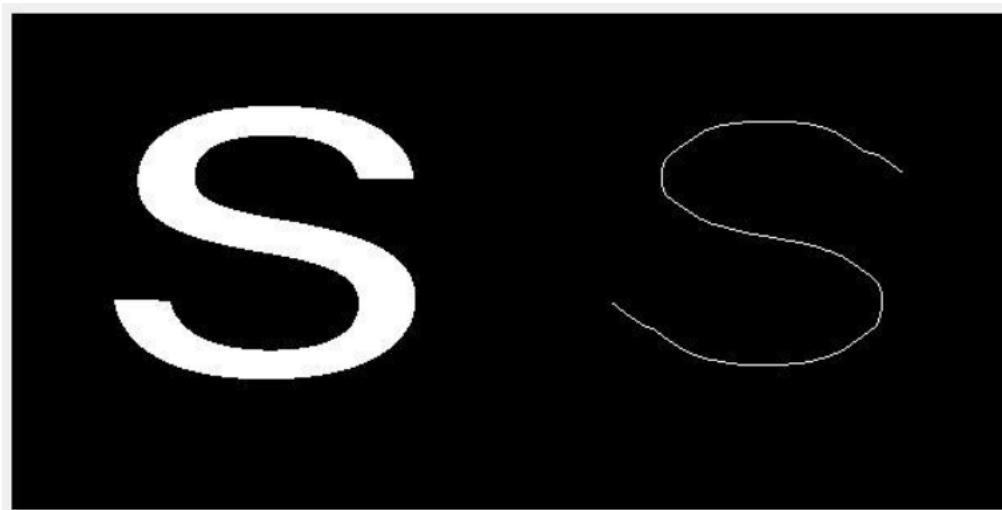
### Penipisan (Thinning)

Operasi thinning (penipisan) sama seperti operasi erosi tetapi tidak membuat objek menjadi menghilang. Operasi thinning akan mereduksi menjadi hanya rangka yang mendekati garis sumbu objek dan bertujuan untuk mengurangi bagian yang tidak perlu sehingga tebal objek tersebut hanya menjadi satu pixel. Salah satu

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 9	<b>Operasi Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 127 dari 183

implementasi dari operasi thinning adalah untuk melakukan identifikasi dari huruf / karakter (Hidayatullah 2017). Operasi thinning dapat direpresentasikan dalam persamaan berikut.

$$A \otimes B = A - (A * B) = A \cap (A * B)^c$$



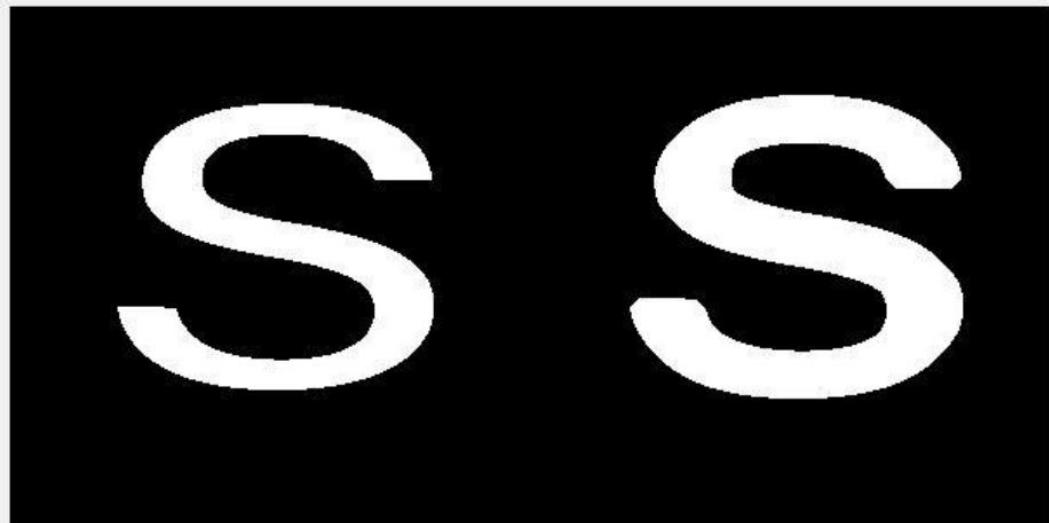
Gambar 8.8 Contoh Implementasi Thinning  
Sumber: dokumen penulis

### Penebalan (Thickening)

Operasi thickening sama seperti operasi dilasi tetapi pada operasi ini tidak akan menyebabkan tergabungnya objek yang terpisah. Operasi thickening bisa disebut juga operasi dual dari thinning (Hidayatullah 2017). Operasi thickening dapat direpresentasikan dalam persamaan berikut.

$$A \bullet B = A \cup (A * B)$$

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 9	<b>Operasi Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 128 dari 183



Gambar 8.9 Contoh Implementasi Thickening  
 Sumber: dokumen penulis

#### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

#### F. Keselamatan Kerja

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera lapor ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik
9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

#### G. Lembar Kerja

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 9	<b>Operasi Morphology</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 129 dari 183

### Kegiatan Praktikum 1: Hit-and-Miss Transform

Siapkan sebuah file untuk digunakan dalam praktikum 1 dan file python baru. Copy kode program berikut ini. Pastikan untuk menyesuaikan file gambar.

```

import cv2
import numpy as np
from cv2 import waitKey

#input image
img = cv2.imread('gambar/coin.jpeg')

gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY_INV +cv2.THRESH_OTSU)

cv2.imshow('image', thresh)

# Construct the structuring element
kernel = np.array([
    [1, 1, 1],
    [0, 1, -1],
    [0, 1, -1]], dtype="int")

# Apply hit-or-miss transformation
output_image = cv2.morphologyEx(thresh, cv2.MORPH_HITMISS, kernel)

cv2.imshow('image asal', img)
cv2.imshow('image hasil', output_image)
waitKey(0)

```

Jalankan program dan periksa apakah jika terjadi error pada terminal. Hasil ditunjukkan pada gambar berikut.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

## PENGENALAN CITRA DAN POLA

Pertemuan : 9

Operasi Morphology

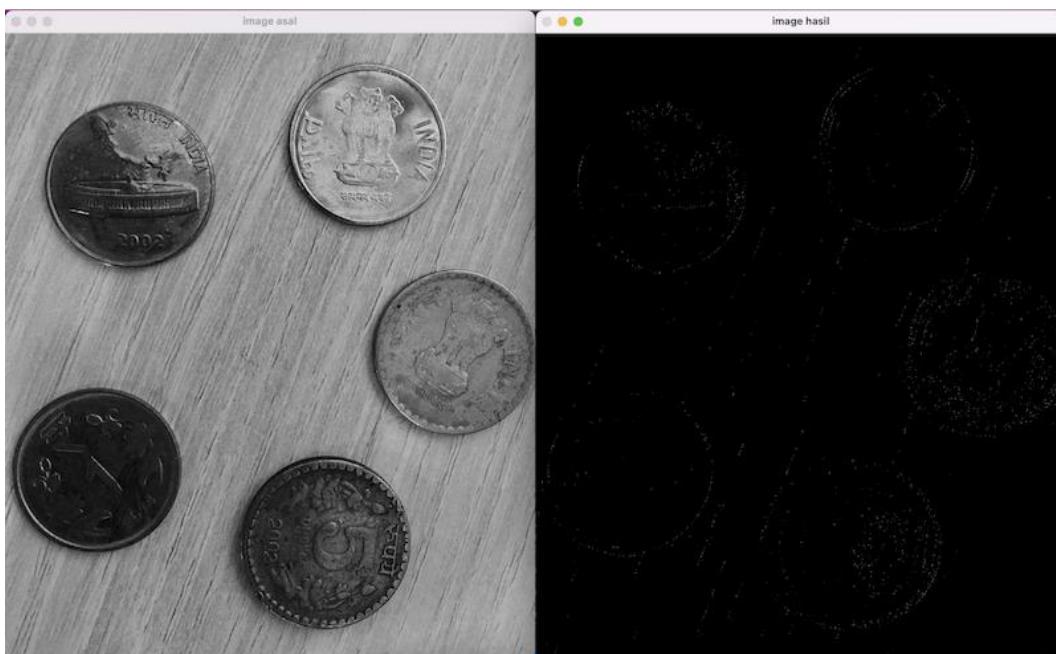
100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 130 dari 183



Hasil terlihat pada gambar diatas dimana terlihat pola tertentu dari piksel latar depan dan latar belakang dalam sebuah gambar. Ini sebenarnya adalah operasi dasar morfologi biner karena hampir semua operator morfologi biner lainnya dapat diturunkan darinya. Seperti operator morfologi biner lainnya, dibutuhkan sebagai input gambar biner dan elemen penataan, dan menghasilkan gambar biner lain sebagai output.

### Kegiatan Praktikum 2: Thickening/Penebalan

Siapkan sebuah file untuk digunakan dalam praktikum 1 dan file python baru. Copy kode program berikut ini. Pastikan untuk menyesuaikan file gambar.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
from cv2 import waitKey

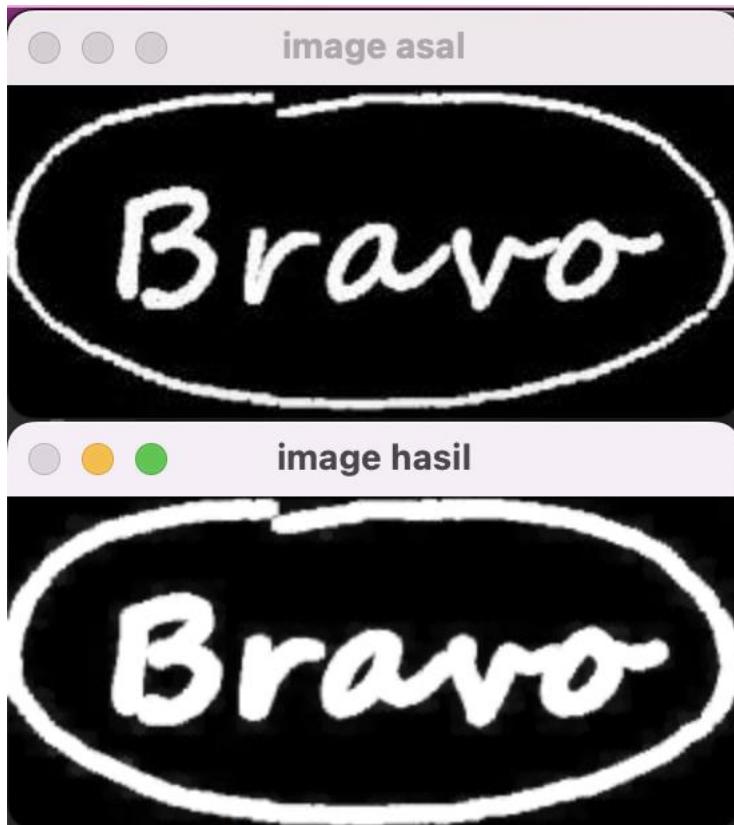
img = cv2.imread('gambar;bravo.jpg')

kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (5, 5))
result = cv2.dilate(img, kernel)

cv2.imshow('image asal', img)
cv2.imshow('image hasil', result)
waitKey(0)
```

Hasil ditunjukkan pada gambar di bawah ini. Jika terjadi error bisa dilihat pada terminal console.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 9	Operasi Morphology	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 131 dari 183



Hasil terlihat penebalan wilayah piksel latar depan yang dipilih dalam gambar biner, seperti dilasi atau closing. Ini memiliki beberapa aplikasi, termasuk menentukan perkiraan *convex hull* suatu bentuk, dan menentukan *skeleton by zone of influence*. Penebalan biasanya hanya diterapkan pada citra biner, dan menghasilkan citra biner lain sebagai output.

#### H. Tugas Praktikum

Carilah sebuah gambar lalu lakukan proses penebalan atau thickening dengan menggunakan kernel ukuran 3x3, 5x5, 7x7, dan 9x9. Analisis hasil output, apakah terjadi perbedaan atau tidak.

#### I. Daftar Referensi

- Hidayatullah, Priyanto. 2017. Pengolahan Citra Digital - Teori dan Aplikasi Nyata. Bandung: Informatika.

## **PETUNJUK PRAKTIKUM #10**

### **PENDETEKSIAN TEPI**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 10	<b>Pendeteksian Tepi</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 133 dari 183

### A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer
2. Mampu menerapkan konsep teoritis dalam bidang rekayasa perangkat lunak

### B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu memahami dan mengaplikasikan metode-metode image segmentation

### C. Tujuan

Tujuan mahasiswa mampu menjelaskan mengenai konsep image segmentation berupa deteksi tepi pada citra dan menerapkan operator deteksi tepi turunan pertama dan kedua.

### D. Dasar Teori

#### Deteksi Tepi

Tepi adalah sebuah himpunan dari piksel-piksel yang terhubung yang berada pada batas (boundary) diantara dua region. Definisi tepi membutuhkan kemampuan untuk mengukur transisi gray-level dengan cara yang tepat. Suatu model tepi yang ideal merupakan suatu himpunan connected pixel yang masing-masing berada pada suatu perubahan langkah tegak lurus (orthogonal) dalam gray-level. Tapi kenyataannya, hasil sampling atau akuisisi citra belum tentu menghasilkan citra yang sesempurna itu, misalkan citra lebih kabur dan sebagainya. Akibatnya tepi lebih mendekati model 'ramplike'. Kemiringan (slope) dari ramp sesuai dengan tingkat kekaburan dari tepi. Ketebalan dari tepi ditentukan oleh panjang dari ramp, yang merupakan transisi dari gray-level awal sampai gray-level akhir. Panjang ramp ditentukan oleh slope, yang ditentukan juga oleh tingkat kekaburan. Berarti bahwa tepi yang kabur cenderung tebal dan tepi yang tajam cenderung tipis. Magnitude dari turunan pertama dapat digunakan untuk mendeteksi adanya suatu tepi pada sebuah titik dalam citra (yang berarti menentukan apakah titik tersebut ada di ramp). Turunan pertama dalam pengolahan citra diimplementasikan menggunakan besaran dari gradient.

Kita mengetahui bahwa pengaburan (blurring) citra pada domain spasial dilakukan dengan rata-rata piksel (averaging) dalam suatu window ketetanggaan. Averaging analog dengan integral, maka dapat dikatakan penajaman citra analog dengan diferensiasi spasial. Metode yang paling umum dari diferensiasi dalam aplikasi pengolahan citra adalah gradient. Untuk suatu fungsi  $f(x, y)$ , gradient dari  $f$  pada koordinat  $(x, y)$  didefinisikan sebagai vektor (Hermawati 2013).

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

## PENGENALAN CITRA DAN POLA

Pertemuan : 10

Pendeteksian Tepi

100 menit

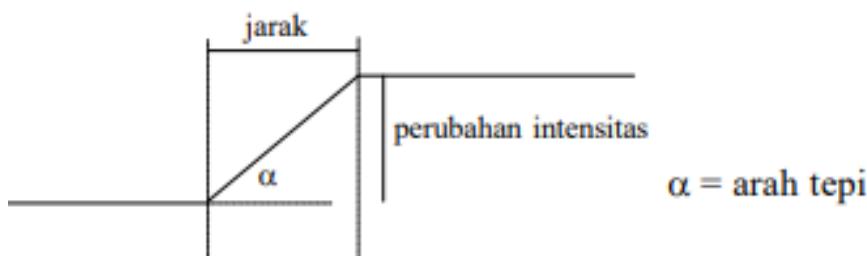
Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 134 dari 183

Tepi (edge) juga adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Perbedaan intensitas inilah yang menampakkan rincian pada gambar. Tepi biasanya terdapat pada batas antara dua daerah berbeda pada suatu citra. Tepi dapat diorientasikan dengan suatu arah (perhatikan Gambar 10.1) dan arah ini berbeda-beda pada bergantung pada perubahan intensitas.



Gambar 10.1 Ilustrasi Perubahan Intensitas pada Tepi Citra  
Sumber: Google Images

Ada tiga macam tepi yang terdapat di dalam citra digital sebagai berikut:

1. Tepi curam, yaitu tepi dengan perubahan intensitas yang tajam. Arah tepi berkisar  $90^\circ$ .
2. Tepi landai, disebut juga tepi lebar, yaitu tepi dengan sudut arah yang kecil. Tepi landai dapat dianggap terdiri dari sejumlah tepi-tepi lokal yang lokasinya berdekatan.
3. Tepi yang mengandung derau (noise). Umumnya tepi yang terdapat pada aplikasi computer vision mengandung derau. Operasi peningkatan kualitas citra (image enhancement) dapat dilakukan terlebih dahulu sebelum pendeksiian tepi.

## PENGENALAN CITRA DAN POLA

Pertemuan : 10

**Pendeteksian Tepi**

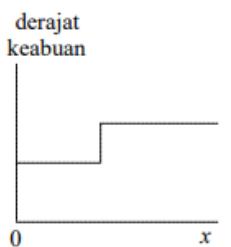
100 menit

Kode MK: TIW-043

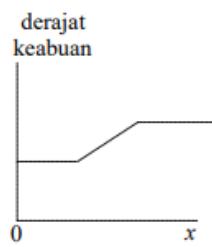
Rev : 00

Tgl : 01/03/2022

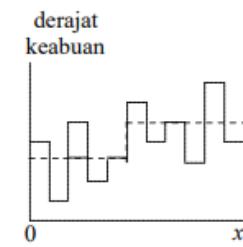
Hal 135 dari 183



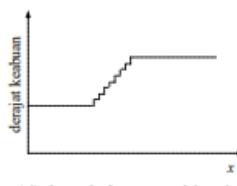
(a) Tepi curam



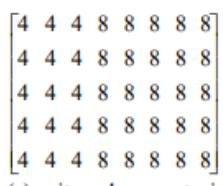
(b) tepi landai



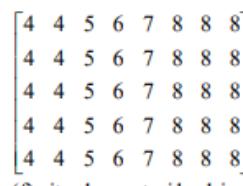
(c) tepi curam dengan derau



(d) break down tepi landai



(e) citra dengan tepi curam



(f) citra dengan tepi landai curam

*Jenis-jenis tepi*

Gambar 10.2 Ilustrasi Jenis Tepi dalam Citra Digital

Sumber: Google Images

### Tujuan Pendekatan

Pendeteksian tepi merupakan langkah pertama untuk melingkupi informasi di dalam citra. Tepi mencirikan batas-batas objek dan karena itu tepi berguna untuk proses segmentasi dan identifikasi objek di dalam citra. Tujuan operasi pendekatan tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra. Tepi termasuk ke dalam komponen berfrekuensi tinggi maka pendekatan tepi dapat dilakukan dengan penapis lolos-tinggi. Terdapat beberapa teknik yang digunakan untuk mendekati tepi, antara lain operator gradien pertama (differential gradient) dan operator turunan kedua (laplacian).

### Deteksi Tepi Turunan Pertama

Perubahan intensitas yang besar dalam jarak yang singkat dipandang sebagai fungsi yang memiliki kemiringan yang besar. Kemiringan fungsi biasanya dilakukan dengan menghitung turunan pertama (gradient).

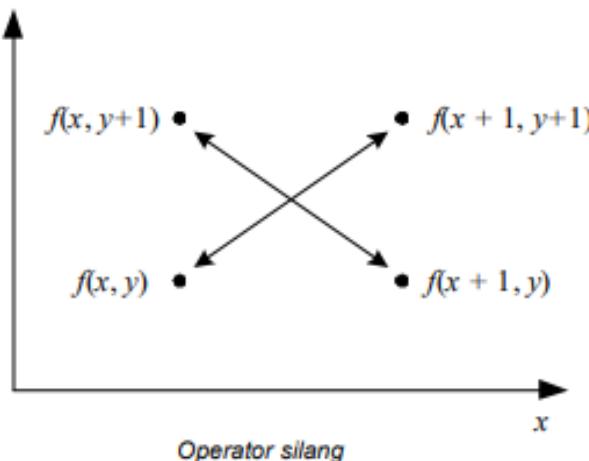
### Operator Roberts

Operator Roberts sering disebut juga operator silang. Gradien Roberts dalam arah-x dan arah-y dihitung dengan rumus:

$$R + (x, y) = f(x + 1, y + 1) - (x, y)$$

$$R - (x, y) = f(x, y + 1) - f(x + 1, y)$$

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 10	Pendeteksian Tepi	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 136 dari 183



Dalam bentuk mask konvolusi, operator Roberts adalah:

$$R+ = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ dan } R- = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Gambar 10.3 memperlihatkan pendekripsi tepi dengan operator Roberts.

$$\begin{bmatrix} 3 & 4 & 2 & 5 & 1 \\ 2 & 1 & 6 & 4 & 2 \\ 3 & 5 & 7 & 1 & 3 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 3 & 2 \end{bmatrix} \quad \begin{bmatrix} 4 & 3 & 3 & 6 & * \\ 5 & 7 & 8 & 2 & * \\ 2 & 5 & 4 & 4 & * \\ 1 & 1 & 8 & 7 & * \\ * & * & * & * & * \end{bmatrix}$$

Gambar 10.3 Implementasi Mask Konvolusi Operator Roberts:  
Citra Awal (Kiri) dan Hasil Konvolusi (Kanan)  
Sumber: Google Images

Nilai 4 pada pojok kiri atas pada citra hasil konvolusi (kanan) diperoleh dengan perhitungan sebagai berikut:

$$f'[0,0] = |3 - 1| + |4 - 2| = 4$$

Citra s diubah ke dalam citra black and white. Untuk mendekripsi tepi citra s menggunakan operator Roberts, digunakan fungsi edge dengan parameter Roberts. Citra awal dan citra hasil deteksi selanjutnya ditampilkan. Gambar 10.4 memperlihatkan hasilnya.

## PENGENALAN CITRA DAN POLA

Pertemuan : 10

Pendeteksian Tepi

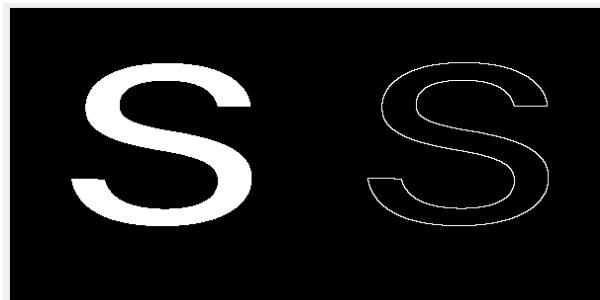
100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 137 dari 183



Gambar 10.4 Implementasi Operator Roberts pada Citra s  
Sumber: dokumen penulis

### Operator Sobel

Tinjau pengaturan pixel di sekitar pixel  $(x,y)$ :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Dalam bentuk mask,  $S_x$  dan  $S_y$  dapat dinyatakan sebagai:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Contoh berikut memperlihatkan pendekatan tepi dengan operator Sobel. Konvolusi pertama dilakukan terhadap pixel yang bernilai 1 (di titik pusat mask):

$$\begin{bmatrix} 3 & 4 & 2 & 5 & 1 \\ 2 & 1 & 6 & 4 & 2 \\ 3 & 5 & 7 & 1 & 3 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 3 & 2 \end{bmatrix}$$

(i) citra semula

$$\begin{bmatrix} * & * & * & * & * \\ * & 18 \end{bmatrix}$$

(ii) hasil konvolusi

Nilai 18 pada citra hasil konvolusi diperoleh dengan perhitungan berikut:

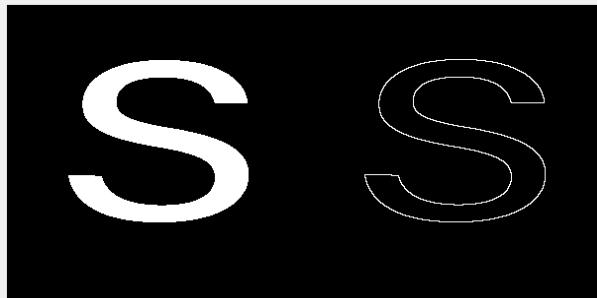
 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 10	Pendeteksian Tepi	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 138 dari 183

$$S_x = (3)(-1) + (2)(-2) + (3)(-1) + (2)(1) + (6)(2) + (7)(1) = 11$$

$$S_y = (3)(1) + (4)(2) + (2)(1) + (3)(-1) + (5)(-2) + (7)(-1) = -7$$

$$M = \sqrt{S_x^2 + S_y^2} = \sqrt{11^2 + (-7)^2} \approx |S_x| + |S_y| = |11| + |-7| = 18$$

Citra s diubah ke dalam citra black and white dengan menggunakan fungsi pada opencv. Untuk mendeteksi tepi citra s menggunakan operator Sobel, digunakan fungsi edge dengan parameter Sobel. Citra awal dan citra hasil deteksi selanjutnya ditampilkan. Gambar 10.5 memperlihatkan hasilnya.

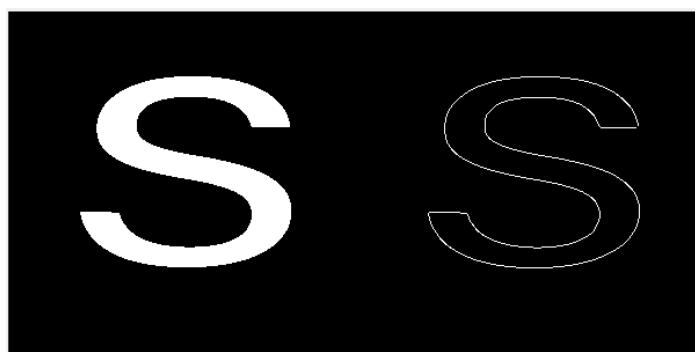


Gambar 10.5 Implementasi Operator Sobel pada Citra S  
Sumber: dokumen penulis

### Operator Prewitt

Persamaan gradien pada operator Prewitt mirip dengan operator Sobel, tetapi menggunakan nilai  $c = 1$ .

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ dan } \begin{bmatrix} -1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$



Gambar 10.6 Implementasi Operator Prewitt pada Citra s  
Sumber: dokumen penulis

### Operator Kirsch

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 10	Pendeteksian Tepi		100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 139 dari 183

Deteksi tepi operator Kirsch diperkenalkan oleh Kirsch pada tahun 1971. Operator ini identik dengan bentuk matriks 3x3 atau jendela ukuran 3x3 piksel dengan  $k_0$  hingga  $k_7$  dihitung menggunakan kernel (mask) sebagai berikut (Haryanto 2015).

$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$ <p><math>k_0</math> Timur (east)</p>	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$ <p><math>k_1</math> Timur laut (north east)</p>	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$ <p><math>k_2</math> Utara (north )</p>
---	--	---

$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$ <p><math>k_3</math> Barat Laut (north west)</p>	$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$ <p><math>k_4</math> Barat (west )</p>	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$ <p><math>k_5</math> Barat Daya (south west)</p>
--	--	--

$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$ <p><math>k_6</math> Selatan (south )</p>	$\begin{bmatrix} -3 & -3 & 3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$ <p><math>k_7</math> Tenggara (south east)</p>
---	---

Gambar 10.7 Mask Operator Kirsch

Sumber: Google Images

### Deteksi Tepi Turunan Kedua

Operator turunan kedua disebut juga operator Laplace. Operator Laplace mendeteksi lokasi tepi lebih akurat khususnya pada tepi yang curam. Pada tepi yang curam, turunan keduanya mempunyai persilangan nol (zero-crossing), yaitu titik di mana terdapat pergantian tanda nilai turunan kedua sedangkan pada tepi yang landai tidak terdapat persilangan nol. Persilangan nol merupakan lokasi tepi yang akurat. Dapat dinyatakan sebagai mask:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

## PENGENALAN CITRA DAN POLA

Pertemuan : 10

Pendeteksian Tepi

100 menit

Kode MK: TIW-043

Rev : 00

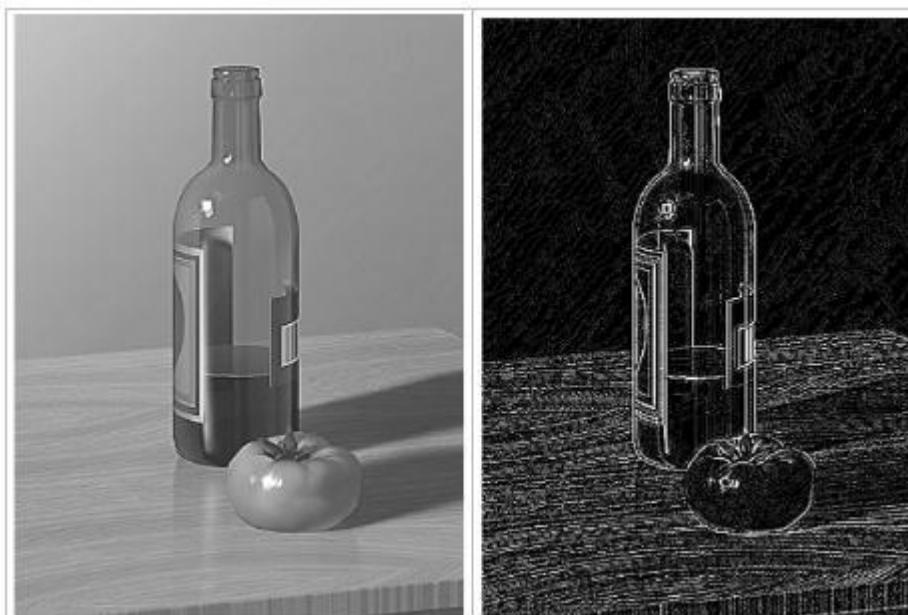
Tgl : 01/03/2022

Hal 140 dari 183

Fungsi Laplace merupakan turunan kedua dari fungsi Gauss, kadang-kadang disebut juga fungsi Laplacian of Gaussian (LoG). Contoh penapis LoG yang berukuran 5 x 5:

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Hasil pendektsian tepi dengan operator Laplacian of Gaussian diperlihatkan pada Gambar 10.8.



Gambar 7. Contoh Implementasi LoG

Sumber: Google Images

### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

### F. Keselamatan Kerja

## **PENGENALAN CITRA DAN POLA**

Pertemuan : 10

Pendeteksian Tepi

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 141 dari 183

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporan ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik
9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

### G. Lembar Kerja

#### Kegiatan Praktikum 1: Deteksi Tepi Robert

Siapkan sebuah gambar untuk digunakan dalam melakukan tahap deteksi tepi. Pada praktikum ini dibutuhkan sebuah library tambahan yaitu *scipy*. Silahkan melakukan instalasi library terlebih dahulu dengan perintah berikut.

```
pip install scipy
```

Selanjutnya silahkan mengetikkan kode berikut dan jalankan program. Pastikan path gambar disesuaikan dengan lokasi gambar yang akan digunakan.

```
import cv2
import numpy as np
from scipy import ndimage

roberts_cross_v = np.array( [[1, 0 ],
                           [0,-1 ] ] )

roberts_cross_h = np.array( [[ 0, 1 ],
                           [-1, 0 ] ] )

img = cv2.imread("gambar/bunga.jpg",0).astype('float64')
img /= 255.0

vertical = ndimage.convolve( img, roberts_cross_v )
horizontal = ndimage.convolve( img, roberts_cross_h )
edged_img = np.sqrt( np.square(horizontal) + np.square(vertical))

cv2.imshow("output.jpg", edged_img)
cv2.waitKey()
```

Hasil ditunjukkan pada gambar berikut ini.

## PENGENALAN CITRA DAN POLA

Pertemuan : 10

Pendeteksian Tepi

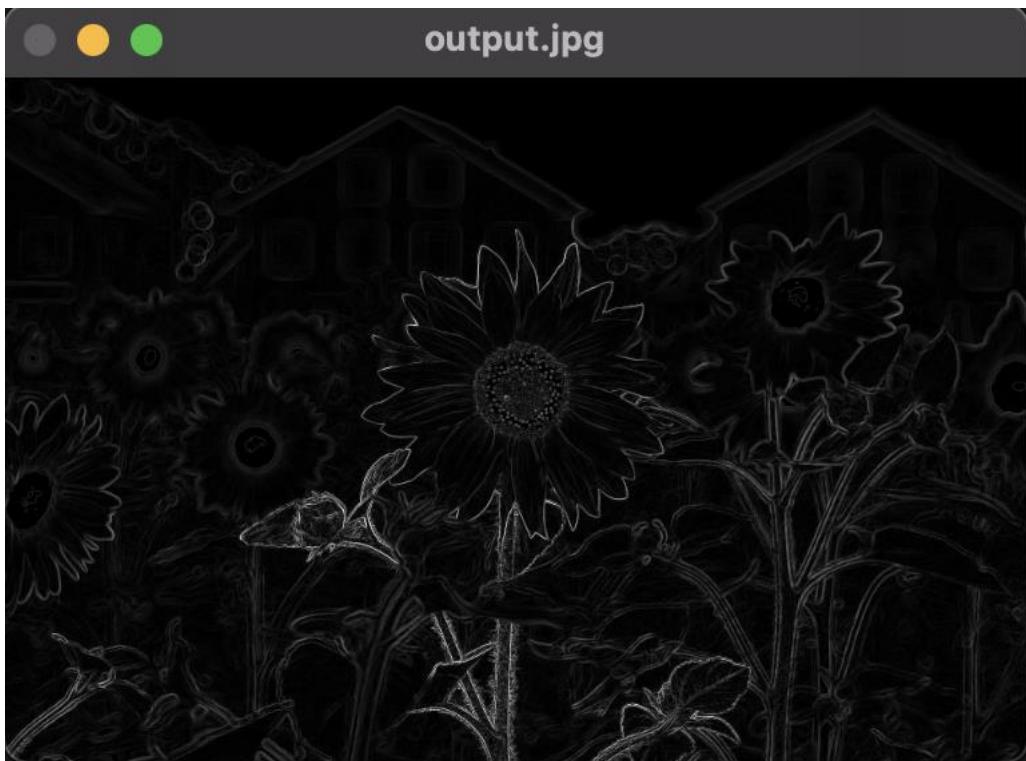
100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 142 dari 183



### Kegiatan Praktikum 2: Deteksi Tepi Sobel

Siapkan sebuah gambar untuk digunakan dalam melakukan tahap deteksi tepi. Selanjutnya copas kode berikut dan jalankan program. Pastikan path gambar disesuaikan dengan lokasi gambar yang akan digunakan.

```
import cv2 as cv
import matplotlib.pyplot as plt

# fetch image
img = cv.imread('gambar/bunga.jpg', cv.COLOR_BGR2GRAY)
rgb_img = cv.cvtColor(img, cv.COLOR_BGR2RGB)

# Grayscale image processing
grayImage = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

# Sobel operator
x = cv.Sobel(grayImage, cv.CV_16S, 1, 0)
y = cv.Sobel(grayImage, cv.CV_16S, 0, 1)

# turn to uint8, image fusion
absX = cv.convertScaleAbs(x)
absY = cv.convertScaleAbs(y)
Sobel = cv.addWeighted(absX, 0.5, absY, 0.5, 0)
```

## PENGENALAN CITRA DAN POLA

Pertemuan : 10

Pendeteksian Tepi

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 143 dari 183

```
# Display graphics
titles = ['Original image', 'Sobel operator']
images = [rgb_img, Sobel]

for i in range(2):
    plt.subplot(1, 2, i + 1), plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([]), plt.yticks([])
plt.show()
```

Hasil menggunakan operator sobel ditunjukkan pada gambar berikut.

Original image



Sobel operator



### Kegiatan Praktikum 3: Deteksi Tepi Prewitt

Siapkan sebuah gambar untuk digunakan dalam melakukan tahap deteksi tepi. Siapkan nilai kernel sesuai dengan ketetuan. Lalu lakukan proses konvolusi pada citra. Pastikan path gambar disesuaikan dengan lokasi gambar yang akan digunakan.

```
import cv2 as cv
import numpy as np
import matplotlib.pyplot as plt

# fetch image
img = cv.imread('gambar/bunga.jpg', cv.COLOR_BGR2GRAY)
rgb_img = cv.cvtColor(img, cv.COLOR_BGR2RGB)

# Prewitt operator
kernelx = np.array([[1,1,1], [0,0,0], [-1,-1,-1]], dtype=int)
kernely = np.array([[-1,0,1], [-1,0,1], [-1,0,1]], dtype=int)

x = cv.filter2D(img, cv.CV_16S, kernelx)
y = cv.filter2D(img, cv.CV_16S, kernely)

# turn to uint8, image fusion
absX = cv.convertScaleAbs(x)
absY = cv.convertScaleAbs(y)
```

## PENGENALAN CITRA DAN POLA

Pertemuan : 10

**Pendeteksian Tepi**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 144 dari 183

```

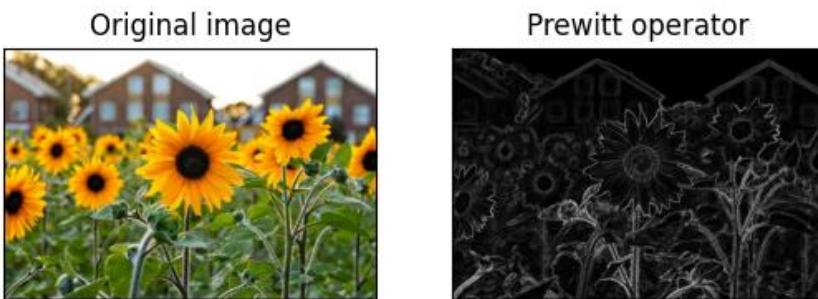
Prewitt = cv.addWeighted(absX, 0.5, absY, 0.5, 0)

# Display graphics
titles = ['Original image', 'Prewitt operator']
images = [rgb_img, Prewitt]

for i in range(2):
    plt.subplot(1, 2, i + 1), plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([]), plt.yticks([])
plt.show()

```

Hasil menggunakan operator prewitt ditunjukkan pada gambar berikut.



### Kegiatan Praktikum 4: Deteksi Tepi Laplacian

Siapkan sebuah gambar untuk digunakan dalam melakukan tahap deteksi tepi. Deteksi tepi menggunakan fungsi pada openCV yaitu cv.Laplacian. Fungsi tersebut sudah disediakan dan bisa langsung digunakan dengan menyesuaikan parameter yang dikirim. Pastikan path gambar disesuaikan dengan lokasi gambar yang akan digunakan.

```

import cv2 as cv
import matplotlib.pyplot as plt

# fetch image
img = cv.imread('gambar/bunga.jpg', cv.COLOR_BGR2GRAY)
rgb_img = cv.cvtColor(img, cv.COLOR_BGR2RGB)

# Grayscale image processing
grayImage = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

# Laplacian
dst = cv.Laplacian(grayImage, cv.CV_16S, ksize = 3)
Laplacian = cv.convertScaleAbs(dst)

# Display graphics
titles = ['Original image', 'the Laplacian operator']
images = [rgb_img, Laplacian]

```

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 10	<b>Pendeteksian Tepi</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 145 dari 183

```
for i in range(2):
    plt.subplot(1, 2, i + 1), plt.imshow(images[i], 'gray')
    plt.title(titles[i])
    plt.xticks([]), plt.yticks([])
plt.show()
```

Hasil dari Operator Laplacian ditunjukkan pada hasil berikut.

Original image



the Laplacian operator



## H. Tugas Praktikum

Buatlah sebuah program sederhana dengan menggunakan Operator Kirsch untuk melakukan proses deteksi tepi. Sebelum di proses silahkan untuk melakukan proses blurring dengan menggunakan gaussian blur!

## I. Daftar Referensi

- Haryanto, Edy Victor. 2015. "Penerapan Metode Kirsch dalam Mendeteksi Tepi Objek Citra Digital." In Konferensi Nasional Sistem & Informatika, Bali: STMIK STIKOM Bali.
- Hermawati, Fajar Astuti. 2013. Pengolahan Citra Digital. Yogyakarta: AndiPublisher.

**PETUNJUK PRAKTIKUM #11**

**SEGMENTASI DAN THRESHOLDING**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 11	<b>Segmentasi dan Thresholding</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 147 dari 183

## A. Capaian Pembelajaran Mata Kuliah

1. Mampu menerapkan algoritma/metode dalam perangkat lunak berbasis komputer
2. Mampu menerapkan konsep teoritis dalam bidang rekayasa perangkat lunak

## B. Sub Capaian Pembelajaran Mata Kuliah

Mahasiswa mampu memahami dan mengaplikasikan metode-metode image segmentation

## C. Tujuan

Tujuan mahasiswa mampu menjelaskan mengenai konsep pengambangan (thresholding) dan segmentasi berbasis clustering pada citra digital.

## D. Dasar Teori

Segmentasi citra adalah pemisahan objek yang satu dengan objek yang lain dalam suatu citra atau antara objek dengan latar yang terdapat dalam sebuah citra. Dengan proses segmentasi tersebut, masing-masing objek pada citra dapat diambil secara individu sehingga dapat digunakan sebagai input bagi proses lain (Cahyan, Aswin, dan Mustofa 2014).

### Pengambangan (Thresholding)

Thresholding merupakan salah satu metode segmentasi citra yang memisahkan antara objek dengan background dalam suatu citra berdasarkan pada perbedaan tingkat kecerahannya atau gelap dan terang. Region citra yang cenderung gelap akan dibuat semakin gelap (hitam sempurna dengan nilai intensitas sebesar 0), sedangkan region citra yang cenderung terang akan dibuat semakin terang (putih sempurna dengan nilai intensitas sebesar 1). Dengan Persamaan secara matematis sebagai berikut.

Metode thresholding ada dua, yaitu :

$$g(x,y) = \begin{cases} 1, & f(x,y) \geq T \\ 0, & f(x,y) < T \end{cases}$$

### Global Thresholding

Teknik thresholding yang paling sederhana adalah mempartisi histogram dengan menggunakan sebuah threshold global dengan memeriksa setiap piksel semi piksel dari citra dan melabeli setiap pixel sebagai objek atau sebagai background, tergantung pada tingkat keabuan dari piksel tersebut, apakah lebih besar atau lebih kecil dari threshold sukses dari metode ini tergantung pada seberapa bagus partisi dari histogram (Hermawati 2013).

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

## PENGENALAN CITRA DAN POLA

Pertemuan : 11

Segmentasi dan Thresholding

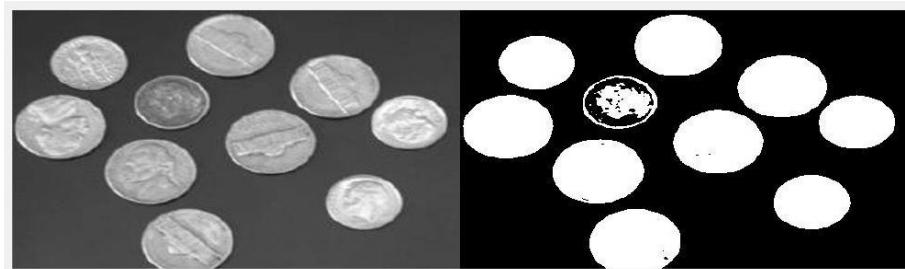
100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 148 dari 183



Gambar 11.1 Implementasi Global Thresholding pada Citra koin

Sumber: dokumen penulis

### Local Adaptive Thresholding

Metode thresholding global dapat gagal jika kontras latar belakang tidak merata. Thresholding akan dikatakan sebagai thresholding lokal jika nilai T (nilai ambang) bergantung pada nilai gray level  $f(x,y)$  dan nilai properti lokal citra  $p(x,y)$ . Dalam thresholding lokal citra akan dibagi ke dalam bagian yang lebih kecil-kecil dan proses pengembangan akan dilakukan secara lokal. Kelebihan yang dimiliki thresholding adalah secara subyektif, citra yang dihasilkan akan lebih bagus. Thresholding lokal dapat ditunjukkan bahwa proses ini adalah setara dengan thresholding  $f(x,y)$  dengan fungsi lokal yang bervariasi  $T$  ambang  $(x,y)$ . Citra dibagi menjadi blok-blok kecil dan kemudian dilakukan thresholding lokal pada setiap blok dengan nilai  $T$  yang berbeda. Secara matematis, dapat dirumuskan sebagai berikut.

$$T = \text{median}\{f(x,y), (x,y) \in W\}$$

$$T = \frac{\max\{f(x,y), (x,y) \in W\} + \min\{f(x,y), (x,y) \in W\}}{2}$$

Perbedaan antara global thresholding dan local adaptive thresholding dapat dilihat pada Gambar 11.2.



 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 11	<b>Segmentasi dan Thresholding</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 149 dari 183

Gambar 2. Hasil Thresholding: Global (Kiri) dan Local Adaptive (Kanan)

Sumber: Google Images

### Penandaan Komponen Terhubung

Pelabelan komponen terhubung memindai gambar dan mengelompokkan pikselnya ke dalam komponen berdasarkan koneksi piksel, yaitu semua piksel dalam komponen yang terhubung memiliki nilai intensitas piksel yang serupa dan dalam beberapa cara terhubung satu sama lain. Setelah semua grup ditentukan, setiap piksel diberi label dengan tingkat abu-abu atau warna (label warna) sesuai dengan komponen yang ditetapkan.

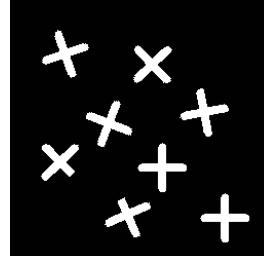
Pelabelan komponen terhubung bekerja dengan memindai gambar, piksel demi piksel (dari atas ke bawah dan kiri ke kanan) untuk mengidentifikasi kawasan piksel yang terhubung, yaitu kawasan piksel yang berdekatan yang memiliki kumpulan nilai intensitas  $V$  yang sama (Untuk a gambar biner  $V = \{1\}$ ; namun, dalam gambar tingkat abu-abu  $V$  akan memiliki rentang nilai, misalnya:  $V = \{51, 52, 53, \dots, 77, 78, 79, 80\}$ ).

Pelabelan komponen yang terhubung berfungsi pada gambar biner atau tingkat abu-abu dan ukuran koneksi yang berbeda dimungkinkan. Operator pelabelan komponen yang terhubung memindai gambar dengan bergerak sepanjang baris hingga mencapai titik  $p$  (di mana  $p$  menunjukkan piksel yang akan diberi label pada setiap tahap dalam proses pemindaian) dengan  $V = \{1\}$ . Jika ini benar, ia memeriksa empat tetangga  $p$  yang telah ditemukan dalam pemindaian (yaitu tetangga (i) di sebelah kiri  $p$ , (ii) di atasnya, dan (iii dan iv) dua suku diagonal atas). Berdasarkan informasi ini, pelabelan  $p$  terjadi sebagai berikut:

1. Jika keempat tetangga adalah 0, tetapkan label baru ke  $p$ .
2. Jika hanya satu tetangga yang memiliki  $V = \{1\}$ , tetapkan labelnya ke  $p$ .
3. jika lebih dari satu tetangga memiliki  $V = \{1\}$ , tetapkan salah satu label ke  $p$  dan catat persamaannya.

Setelah menyelesaikan pemindaian, pasangan label yang setara diurutkan ke dalam kelas kesetaraan dan label unik diberikan untuk setiap kelas. Sebagai langkah terakhir, pemindaian kedua dilakukan melalui gambar, di mana setiap label diganti dengan label yang ditetapkan ke kelas ekuivalennya. Untuk tampilan, label mungkin memiliki tingkat atau warna abu-abu yang berbeda.

Untuk mengilustrasikan pelabelan komponen yang terhubung, dimulai dengan gambar biner sederhana yang berisi beberapa objek buatan yang berbeda.



## PENGENALAN CITRA DAN POLA

Pertemuan : 11

Segmentasi dan Thresholding

100 menit

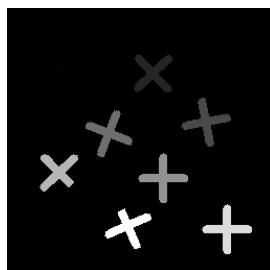
Kode MK: TIW-043

Rev : 00

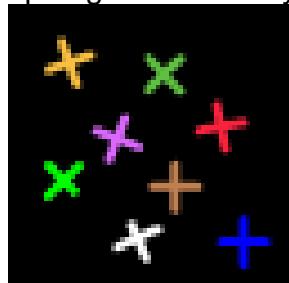
Tgl : 01/03/2022

Hal 150 dari 183

Setelah memindai gambar ini dan memberi label pada kelas piksel yang berbeda dengan nilai abu-abu yang berbeda, didapatkan gambar keluaran berlabel.



Perhatikan bahwa gambar diskalakan, karena nilai abu-abu awal (1-8) semuanya akan tampak hitam di layar. Namun, piksel yang awalnya ditetapkan ke kelas bawah (1 dan 2) masih tidak terlihat dari latar belakang. Jika kita menetapkan warna yang berbeda untuk setiap tingkat abu-abu yang kita peroleh.



Kegunaan penuh dari pelabelan komponen terhubung dapat direalisasikan dalam skenario analisis citra dimana citra diproses sebelumnya melalui beberapa segmentasi (misalnya thresholding) atau skema klasifikasi. Salah satu aplikasinya adalah menggunakan pelabelan komponen terhubung untuk menghitung objek dalam gambar. Misalnya, dalam adegan sederhana di atas, 8 objek menghasilkan 8 kelas yang berbeda.

Kumpulan operator morfologi ada untuk mengekstraksi komponen yang terhubung dan melabelinya dengan berbagai cara. Metode sederhana untuk mengekstraksi komponen terhubung dari suatu citra menggabungkan dilasi dan operasi perpotongan matematis. Yang pertama mengidentifikasi piksel yang merupakan bagian dari kawasan kontinu yang berbagi sekumpulan nilai intensitas umum  $V = \{1\}$  dan yang terakhir menghilangkan dilatasi yang berpusat pada piksel dengan  $V = \{0\}$ . Elemen penataan yang digunakan menentukan konektivitas yang diinginkan.

Varian yang lebih canggih dari ini mencakup serangkaian fungsi geodesik untuk mengukur bentuk yang tepat dari objek berbeda dalam sebuah gambar. Operator ini didasarkan pada pengertian jarak geodesik  $d$  yang didefinisikan sebagai jarak terpendek antara dua titik yang terletak di dalam objek gambar sehingga seluruh jalur antar titik termasuk dalam objek. Misalnya, perhatikan gambar berikut.

## PENGENALAN CITRA DAN POLA

Pertemuan : 11

**Segmentasi dan Thresholding**

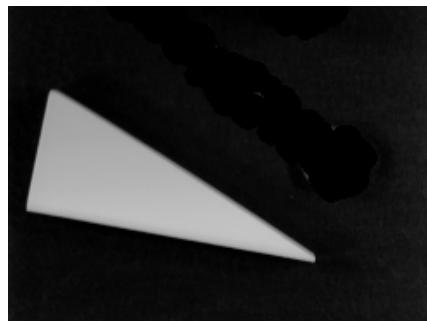
100 menit

Kode MK: TIW-043

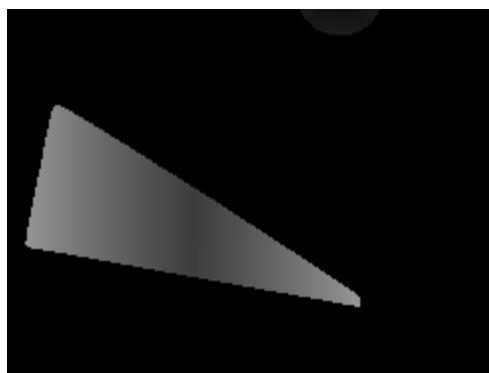
Rev : 00

Tgl : 01/03/2022

Hal 151 dari 183



Pelabelan intensitas tingkat abu-abu di seluruh permukaan blok mengkodekan jarak geodesik, yaitu piksel cahaya menunjukkan jarak yang lebih besar.



Algoritma penandaan komponen terhubung:

1. Apakah pixel sebelah kiri memiliki nilai yang sama dengan pixel saat ini? Jika:
  - Ya, berikan label yang sama pada pixel saat ini
  - Tidak, cek langkah selanjutnya
2. Apakah pixel sebelah kiri dan pixel di atas memiliki nilai yang sama, dan berbeda dengan pixel saat ini? Jika:
  - Ya, berikan label baru untuk pixel saat ini
  - Tidak, cek langkah selanjutnya
3. Apakah pixel di atas memiliki nilai yang sama dengan pixel saat ini? Jika:
  - Ya, berikan label yang sama pada pixel saat ini
  - Tidak, cek langkah selanjutnya
4. Apakah pixel sebelah kiri dan atas memiliki nilai yang berbeda dengan pixel saat ini?
  - Ya, berikan label baru untuk pixel saat ini

## PENGENALAN CITRA DAN POLA

Pertemuan : 11

Segmentasi dan Thresholding

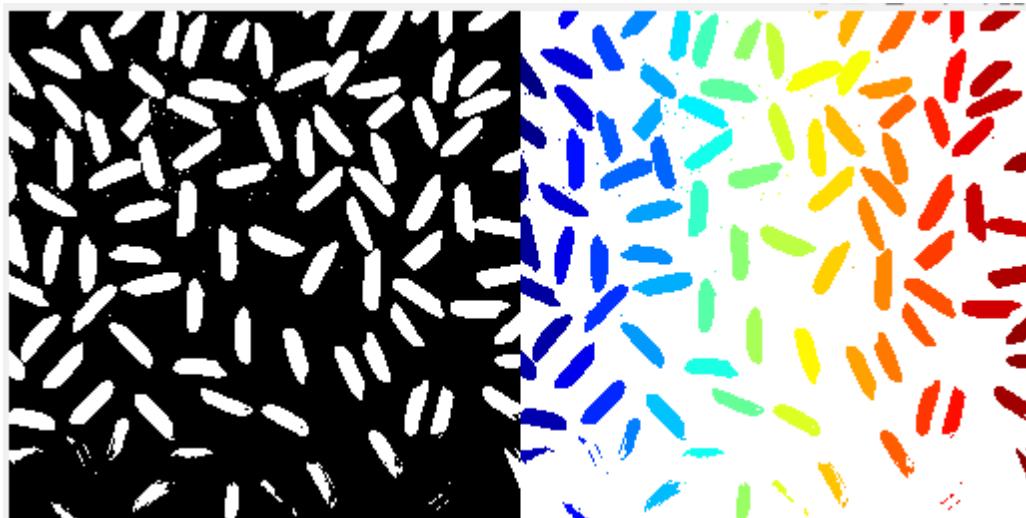
100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 152 dari 183



Gambar 11. Implementasi Penandaan Komponen Terhubung pada Citra rice  
Sumber: dokumen penulis

### Segmentasi Berbasis Clustering

Segmentasi objek pada citra adalah K-Means Clustering, yang termasuk dalam hard- clustering. Hard-clustering merupakan teknik pengelompokan yang membagi gambar menjadi beberapa kelompok dimana setiap piksel hanya masuk ke dalam satu kelompok (cluster).

Algoritma pengelompokan K-Means adalah algoritma tanpa pengawasan dan digunakan untuk melakukan segmentasi area minat dari latar belakang. Ini mengelompokkan, atau mempartisi data yang diberikan ke dalam K-cluster atau bagian berdasarkan K-centroids. Algoritma digunakan jika data tidak memiliki label. Tujuannya adalah untuk menemukan kelompok tertentu berdasarkan kemiripan datanya dengan jumlah kelompok yang direpresentasikan oleh K.

## PENGENALAN CITRA DAN POLA

Pertemuan : 11

Segmentasi dan Thresholding

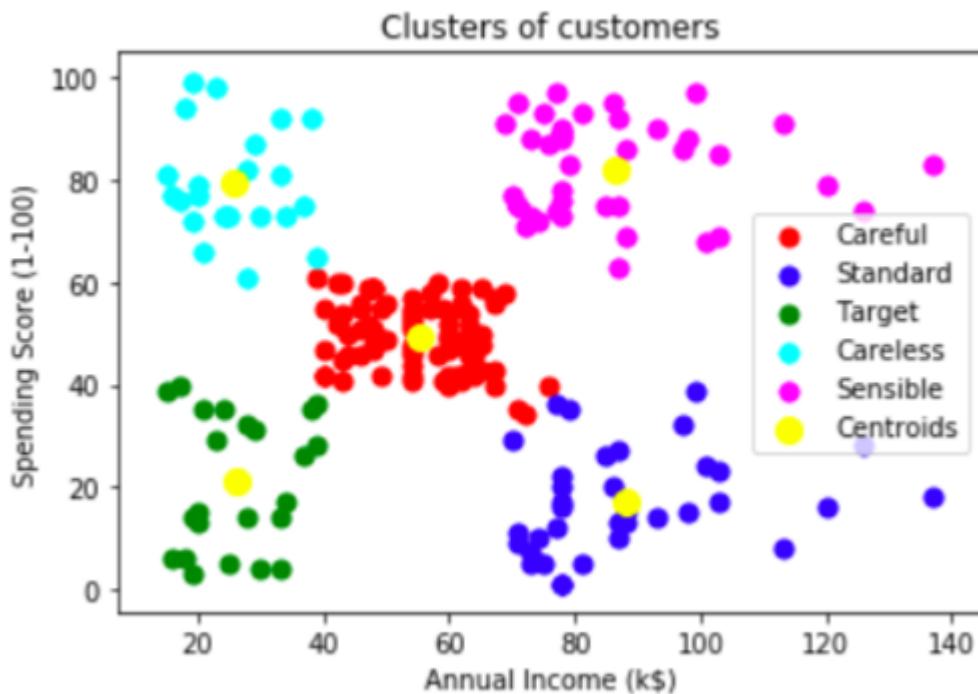
100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 153 dari 183



Gambar 11.4 Ilustrasi Clustering pada Data Pelanggan  
Sumber: Google Images

Gambar 11.4 menunjukkan pelanggan pusat perbelanjaan telah dikelompokkan menjadi lima cluster berdasarkan skor pendapatan dan pengeluarannya. Titik kuning mewakili centroid dari setiap cluster. Tujuan dari pengelompokan K-Means adalah untuk meminimalkan jumlah kuadrat jarak antara semua titik dan pusat cluster. Dalam matematika, K-Means Clustering dapat dituliskan sebagai berikut.

$$\text{objective function } \leftarrow J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2$$

number of clusters      number of cases      centroid for cluster  $j$   
 $k$        $n$        $c_j$   
 case  $i$   
 Distance function

Langkah-langkah dalam algoritma:

1. Pilih jumlah cluster K.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 11	<b>Segmentasi dan Thresholding</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 154 dari 183

2. Pilih secara acak K poin, sentroid (tidak harus dari kumpulan data).
3. Tetapkan setiap titik data ke pusat massa terdekat → yang membentuk kluster K.
4. Hitung dan tempatkan sentroid baru dari setiap cluster.
5. Tetapkan ulang setiap titik data ke pusat massa terdekat yang baru. Ulangi dari langkah 3 jika titik sentroid mengalami perubahan posisi.



Gambar 11.5 Implementasi K-Means Clustering pada Citra koala:  
 Citra Awal (Kiri), Cluster 3 (Tengah), Cluster 4 (Kanan)

Sumber: dokumen penulis

#### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

#### F. Keselamatan Kerja

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporan ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika  Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 11	<b>Segmentasi dan Thresholding</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 155 dari 183

9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

## G. Lembar Kerja

### Kegiatan Praktikum 1: Thresholding

Siapkan sebuah gambar yang akan digunakan untuk proses thresholding. Thresholding adalah teknik segmentasi yang sangat populer, digunakan untuk memisahkan objek yang dianggap sebagai latar depan (foreground) dari latar belakangnya (background). Nilai ambang batas diperlukan dalam proses ini. Ambang batas adalah nilai yang memiliki dua wilayah di kedua sisinya yaitu di bawah ambang batas atau di atas ambang batas. Dalam Computer Vision, teknik thresholding ini dilakukan pada citra grayscale. Jadi awalnya, gambar harus dikonversi dalam ruang warna skala abu-abu. Pada library OpenCV untuk melakukan thresholding menggunakan fungsi `cv2.threshold`. Ketikkan kode program berikut ini pada code editor.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('gambar/bunga.jpg', 0)
ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)
ret,thresh5 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)

titles = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]

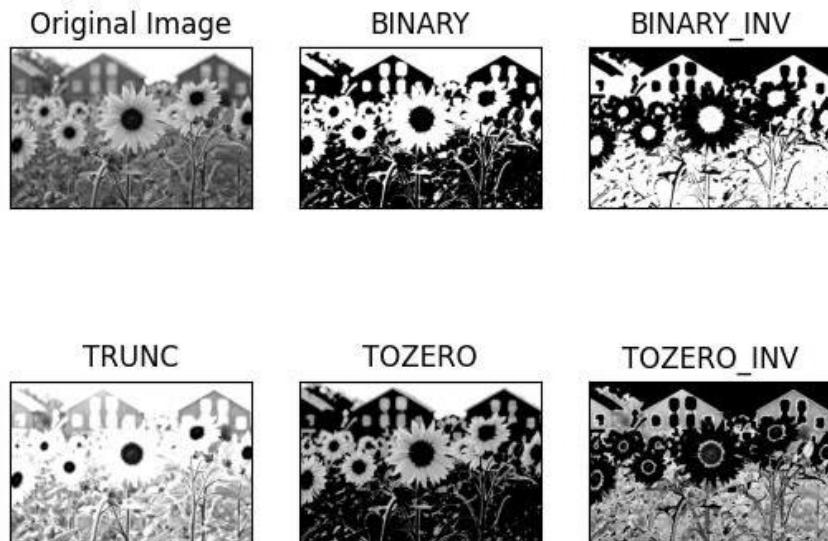
for i in range(6):
    plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')
    plt.title(titles[i])
    plt.xticks([]),plt.yticks([])

plt.show()
```

Hasil program ditunjukkan pada gambar berikut.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 11	<b>Segmentasi dan Thresholding</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 156 dari 183



Untuk masing-masing fungsi memiliki proses yang berbeda. Berikut beberapa teknik yang dapat digunakan dalam proses thresholding.

- cv2.THRESH\_BINARY: Jika intensitas piksel lebih besar dari ambang batas yang ditetapkan, nilai disetel ke 255, jika tidak, setel ke 0 (hitam).
- cv2.THRESH\_BINARY\_INV: Kasus terbalik atau kebalikan dari cv2.THRESH\_BINARY.
- cv.THRESH\_TRUNC: Jika nilai intensitas piksel lebih besar dari ambang batas, maka akan terpotong ke ambang batas. Nilai piksel diatur agar sama dengan ambang batas. Semua nilai lainnya tetap sama.
- cv.THRESH\_TOZERO: Intensitas piksel diatur ke 0, untuk semua intensitas piksel, kurang dari nilai ambang batas.
- cv.THRESH\_TOZERO\_INV: Kasus terbalik atau kebalikan dari cv2.THRESH\_TOZERO.

## PENGENALAN CITRA DAN POLA

Pertemuan : 11

Segmentasi dan Thresholding

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 157 dari 183

Binary

$$dst(x, y) = \begin{cases} \text{maxval} & \text{if } src(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

Inverted Binary

$$dst(x, y) = \begin{cases} 0 & \text{if } src(x, y) > \text{thresh} \\ \text{maxval} & \text{otherwise} \end{cases}$$

Truncated

$$dst(x, y) = \begin{cases} \text{threshold} & \text{if } src(x, y) > \text{thresh} \\ src(x, y) & \text{otherwise} \end{cases}$$

To Zero

$$dst(x, y) = \begin{cases} src(x, y) & \text{if } src(x, y) > \text{thresh} \\ 0 & \text{otherwise} \end{cases}$$

To Zero Inverted

$$dst(x, y) = \begin{cases} 0 & \text{if } src(x, y) > \text{thresh} \\ src(x, y) & \text{otherwise} \end{cases}$$

### Kegiatan Praktikum 2: Adaptive Thresholding

Pada proses adaptive thresholding pada openCV dapat dilakukan dengan 2 cara, yaitu mean dan gaussian.

- cv2.ADAPTIVE\_THRESH\_MEAN\_C : nilai ambang batas adalah rata-rata area tetangga.
- cv2.ADAPTIVE\_THRESH\_GAUSSIAN\_C : nilai ambang batas adalah jumlah bobot nilai tetangga di mana bobot menggunakan kernel gaussian.

Siapkan sebuah file dan ketikkan kode program berikut ini.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

img = cv2.imread('gambar/bunga.jpg', 0)
img = cv2.medianBlur(img,5)

ret,th1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
th2 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY,11,2)
th3 = cv2.adaptiveThreshold(img,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,
cv2.THRESH_BINARY,11,2)
```

## PENGENALAN CITRA DAN POLA

Pertemuan : 11

Segmentasi dan Thresholding

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 158 dari 183

```
titles = ['Original Image', 'Global Thresholding (v = 127)',  
         'Adaptive Mean Thresholding', 'Adaptive Gaussian Thresholding']  
images = [img, th1, th2, th3]  
  
for i in range(4):  
    plt.subplot(2,2,i+1),plt.imshow(images[i],'gray')  
    plt.title(titles[i])  
    plt.xticks([]),plt.yticks([])  
plt.show()
```

Hasil adaptive thresholding ditunjukkan pada gambar berikut.

Original Image



Global Thresholding (v = 127)



Adaptive Mean Thresholding



Adaptive Gaussian Thresholding



### Kegiatan Praktikum 3: Otsu's Binarization

Untuk threshold dengan otsu menggunakan fungsi cv2.threshold(), tetapi berikan tanda tambah (+), cv2.THRESH\_OTSU. Untuk nilai ambang batas, diset nol. Kemudian algoritma menemukan nilai ambang optimal dan mengembalikan sebagai keluaran kedua. Jika ambang batas Otsu tidak digunakan, akan menggunakan nilai ambang batas yang digunakan.

```
import cv2  
import numpy as np  
from matplotlib import pyplot as plt  
  
img = cv2.imread('gambar/bunga.jpg', 0)  
  
# global thresholding
```

## **PENGENALAN CITRA DAN POLA**

Pertemuan : 11

**Segmentasi dan Thresholding**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 159 dari 183

```

ret1,th1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)

# Otsu's thresholding
ret2,th2 = cv2.threshold(img,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)

# Otsu's thresholding after Gaussian filtering
blur = cv2.GaussianBlur(img,(5,5),0)
ret3,th3 = cv2.threshold(blur,0,255,cv2.THRESH_BINARY+cv2.THRESH_OTSU)

# plot all the images and their histograms
images = [img, 0, th1,
          img, 0, th2,
          blur, 0, th3]
titles = ['Original Noisy Image','Histogram','Global Thresholding (v=127)',
          'Original Noisy Image','Histogram',"Otsu's Thresholding",
          'Gaussian filtered Image','Histogram',"Otsu's Thresholding"]

for i in range(3):
    plt.subplot(3,3,i*3+1),plt.imshow(images[i*3],'gray')
    plt.title(titles[i*3]), plt.xticks([]), plt.yticks([])
    plt.subplot(3,3,i*3+2),plt.hist(images[i*3].ravel(),256)
    plt.title(titles[i*3+1]), plt.xticks([]), plt.yticks([])
    plt.subplot(3,3,i*3+3),plt.imshow(images[i*3+2],'gray')
    plt.title(titles[i*3+2]), plt.xticks([]), plt.yticks([])
plt.show()

```

Gambar masukan adalah gambar dengan noise. Dalam kasus pertama, menerapkan ambang batas global untuk nilai 127. Dalam kasus kedua, menerapkan ambang batas Otsu secara langsung. Dalam kasus ketiga, gambar difilter dengan kernel gaussian 5x5 untuk menghilangkan noise, lalu menerapkan ambang batas Otsu. Hasil ditunjukkan pada gambar berikut.

## PENGENALAN CITRA DAN POLA

Pertemuan : 11

Segmentasi dan Thresholding

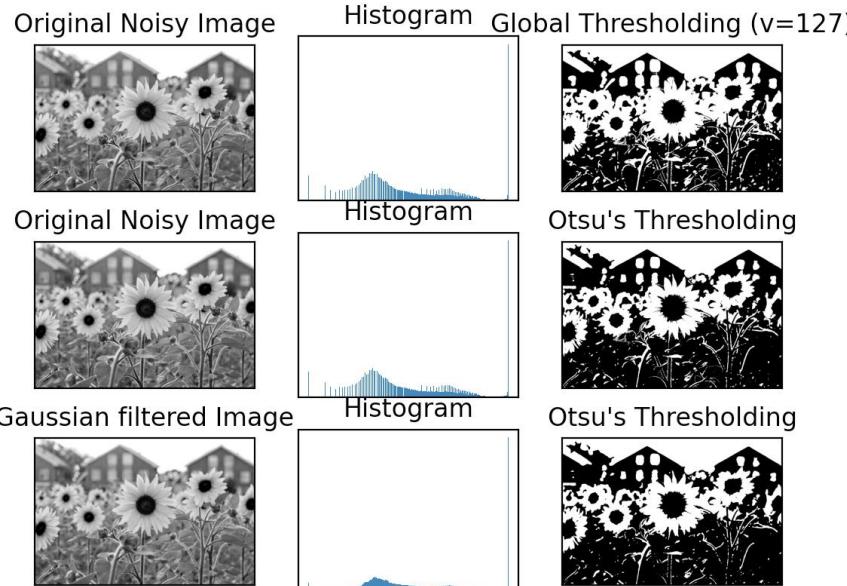
100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 160 dari 183



### Kegiatan Praktikum 4: Image Segmentation menggunakan K-Means

Siapkan sebuah file dan ketikan kode program berikut pada code editor yang digunakan. Gambar dirubah menjadi format RGB terlebih dahulu. Untuk proses segmentasi menggunakan fungsi `cv2.kmeans()`. Fungsi ini mengambil array 2D sebagai input, dan karena gambar asli adalah 3D (lebar, tinggi, dan kedalaman 3 nilai RGB), maka perlu meratakan tinggi dan lebar menjadi vektor tunggal piksel (3 nilai RGB):

```
import numpy as np
import matplotlib.pyplot as plt
import cv2

# Read in the image
image = cv2.imread('gambar/forest.jpg')

# Change color to RGB (from BGR)
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image)

# Reshaping the image into a 2D array of pixels and 3 color values (RGB)
pixel_vals = image.reshape((-1,3))

# Convert to float type
pixel_vals = np.float32(pixel_vals)

#the below line of code defines the criteria for the algorithm to stop running,
#which will happen if 100 iterations are run or the epsilon (which is the required accuracy)
```

## PENGENALAN CITRA DAN POLA

Pertemuan : 11

**Segmentasi dan Thresholding**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 161 dari 183

```
#becomes 85%
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.8)

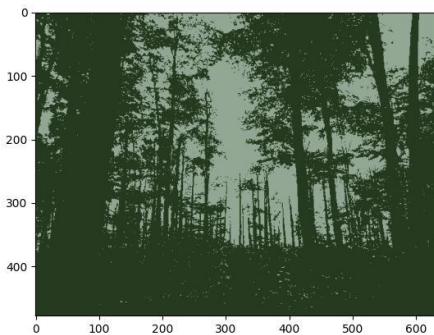
# then perform k-means clustering with h number of clusters defined as 5
#also random centres are initially choosed for k-means clustering
k = 5
retval, labels, centers = cv2.kmeans(pixel_vals, k, None, criteria, 10,
cv2.KMEANS_RANDOM_CENTERS)
# number of clusters (K)

# convert data into 8-bit values
centers = np.uint8(centers)
segmented_data = centers[labels.flatten()]

# reshape data into the original image dimensions
segmented_image = segmented_data.reshape((image.shape))

plt.imshow(segmented_image)
plt.show()
```

Hasil dari proses segmentasi ditunjukkan pada gambar berikut.



Gambar kiri menggunakan k=2 dan kanan menggunakan k=5

### H. Tugas Praktikum

Buatlah sebuah pemrosesan image segmentasi menggunakan kmeans dengan menganalisis jumlah iterasi dan kluster. Jumlah iterasi yang digunakan mulai dari 20, 50 dan 100 serta nilai kluster yaitu 2, 5, dan 7. Bandingkan hasil citra yang dihasilkan.

### I. Referensi

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 11	<b>Segmentasi dan Thresholding</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 162 dari 183

- Cahyan, Pramuda Akariusta, Muhammad Aswin, dan Ali Mustofa. 2014. “Segmentasi Citra Digital Dengan Menggunakan Algoritma Watershed dan Lowpass Filter Sebagai Proses Awal.” Jurnal Mahasiswa TEUB 1(1). <http://elektro.studentjournal.ub.ac.id/index.php/teub/article/view/142> (Agustus 18, 2020).
- Hermawati, Fajar Astuti. 2013. Pengolahan Citra Digital. Yogyakarta: AndiPublisher.

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

**PETUNJUK PRAKTIKUM #12**

**PENGENALAN POLA BERBASIS JARAK**

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 12	<b>Pengenalan Pola Berbasis Jarak</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 164 dari 183

## A. Capaian Pembelajaran Mata Kuliah

1. Mampu mengaplikasikan konsep rekayasa perangkat lunak
2. Mampu membuat program untuk mengimplementasikan pemecahan masalah dengan membangun program komputer, dan dengan memanfaatkan framework, atau teknologi informasi yang terkini (up to date)

## B. Sub Capaian Pembelajaran Mata Kuliah

1. Mahasiswa mampu merancang dan membuat sebuah aplikasi dengan mengkombinasikan beberapa teknik pengolahan citra digital

## C. Tujuan

Tujuan pada pertemuan ini adalah mahasiswa mampu memahami sebuah alur dalam pengenalan pola dan menganalisis metode pengenalan berbasis jarak pada pengenalan pola

## D. Dasar Teori

Pola adalah bentuk atau model yang memiliki keteraturan, baik dalam desain maupun gagasan abstrak. Unsur pembentuk pola disusun secara berulang dalam aturan tertentu sehingga dapat diprakirakan kelanjutannya. Pola dapat dipakai untuk menghasilkan sesuatu atau bagian dari sesuatu, contoh dalam dunia desain adalah seperti kertas dinding dan corak kain. Pola yang paling sederhana didasarkan pada pengulangan: beberapa tiruan sejenis digabungkan tanpa modifikasi.

### Pengertian Pola dan Ciri

Pola adalah entitas yang terdefinisi dan dapat diidentifikasi melalui ciri-cirinya (features). Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi. Sebagai contoh.

Pola	Ciri
Huruf	tinggi, tebal, titik sudut, lengkungan garis, dll
Suara	amplitudo, frekuensi, nada, intonasi, warna, dll
Tanda tangan	panjang, kerumitan, tekanan, dll
Sidik jari	lengkungan, jumlah garis, dll

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 12	<b>Pengenalan Pola Berbasis Jarak</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 165 dari 183

Ciri pada suatu pola diperoleh dari hasil pengukuran terhadap objek uji. Khusus pada pola yang terdapat di dalam citra, ciri-ciri yang dapat diperoleh berasal dari informasi:

- a. Spasial: intensitas pixel, histogram, ...
- b. Tepi: arah, kekuatan, ...
- c. Kontur: garis, elips, lingkaran, ...
- d. Wilayah/bentuk: keliling, luas, pusat massa, ...
- e. Hasil transformasi Fourier: frekuensi, ...

### Sistem Pengenalan Pola

Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu objek dengan objek lain. Terdapat dua pendekatan yang dilakukan dalam pengenalan pola: pendekatan secara statistik dan pendekatan secara sintaktik atau struktural.

### Pengenalan Pola secara Statistik

Pendekatan ini menggunakan teori-teori ilmu peluang dan statistik. Ciri-ciri yang dimiliki oleh suatu pola ditentukan distribusi statistiknya. Pola yang berbeda memiliki distribusi yang berbeda pula. Dengan menggunakan teori keputusan di dalam statistik, digunakan distribusi ciri untuk mengklasifikasikan pola. Contoh teori keputusan: misalkan terdapat N pola yang dikenali, yaitu  $w_1, w_2, \dots, w_N$  dan fungsi peluang atau kerapatan dari ciri-ciri pada pola diketahui. Jika  $\vec{x}$  merupakan hasil pengukuran ciri-ciri, maka

$$P(\vec{x} | w_i), i = 1, 2, \dots, N$$

dapat dihitung. Sebagai contoh, misalkan diketahui fungsi kerapatan dari diameter buah jeruk dan apel yang diperlihatkan sebagai berikut.

## PENGENALAN CITRA DAN POLA

Pertemuan : 12

Pengenalan Pola Berbasis Jarak

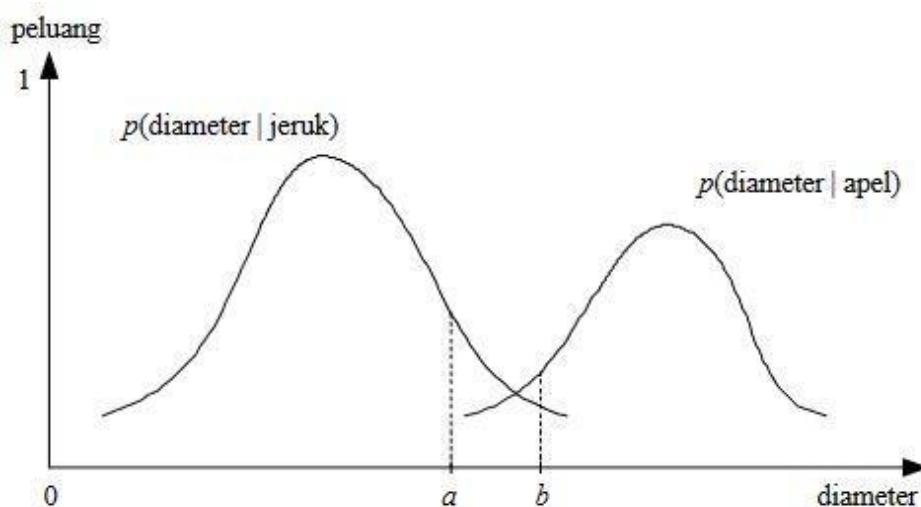
100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 166 dari 183



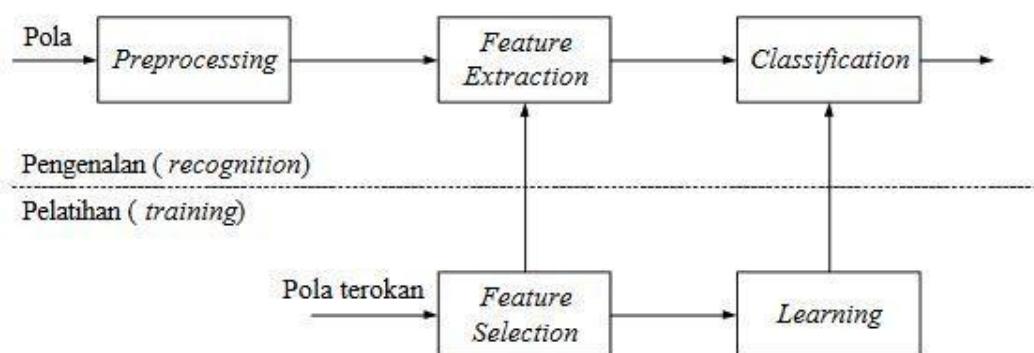
Jika sebuah objek diukur dan diperoleh diameternya adalah a cm, maka kita mengklasifikasikan objek tersebut sebagai "jeruk", karena

$$p(a|\text{jeruk}) > p(a \vee \text{apel})$$

Jika hasil pengukuran diameter adalah b cm, kita mengklasifikasikan objek tersebut sebagai "apel", karena

$$p(b|\text{jeruk}) < p(b \vee \text{apel})$$

Sistem pengenalan pola dengan pendekatan statistik ditunjukkan oleh diagram berikut.



Ada dua fase dalam sistem pengenalan pola: (i) fase pelatihan dan (ii) fase pengenalan. Pada fase pelatihan, beberapa contoh citra dipelajari untuk menentukan ciri yang akan digunakan dalam proses pengenalan serta prosedur

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <h1 style="margin: 0;">PENGENALAN CITRA DAN POLA</h1>		
Pertemuan : 12	<b>Pengenalan Pola Berbasis Jarak</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 167 dari 183

klasifikasinya. Pada fase pengenalan, citra diambil cirinya kemudian ditentukan kelas kelompoknya.

### Preprocessing

Proses awal yang dilakukan untuk memperbaiki kualitas citra (edge enhancement) dengan menggunakan teknik-teknik pengolahan citra yang sudah diejelaskan sebelumnya.

### Feature Extraction

Untuk mengenali obyek dalam citra dibutuhkan parameter-parameter yang mencirikan obyek tersebut. Ciri yang dapat digunakan untuk membedakan obyek satu dengan obyek lainnya di antaranya adalah ciri bentuk, ciri ukuran, ciri geometri, ciri tekstur, dan ciri warna. Masing-masing obyek diekstrak cirinya berdasarkan parameter-parameter tertentu dan dikelompokkan pada kelas tertentu. Misalnya untuk mencirikan ukuran suatu obyek yang termasuk dalam kelas ukuran besar maka digunakan parameter luas dan keliling. Nilai dari parameter-parameter tersebut kemudian dijadikan sebagai data masukan dalam proses identifikasi/ klasifikasi. Pada proses pengenalan pola yang kompleks dibutuhkan ciri yang kompleks pula, oleh sebab itu perlu dilakukan kajian mengenai ciri apa yang benar-benar dapat membedakan antara obyek satu dengan obyek yang lain.

### Classification

Dalam proses ini, nilai parameter-parameter yang merepresentasikan ciri obyek pada masing-masing kelas dijadikan sebagai data masukan. Data tersebut kemudian diolah sehingga diperoleh suatu rumusan untuk dapat mengenali obyek. Dalam tahapan identifikasi, umumnya dilakukan dua proses utama yaitu proses pelatihan dan proses pengujian. Proses pelatihan dilakukan menggunakan sekumpulan data latih yang memuat parameter ciri/ feature yang digunakan untuk membedakan antara objek satu dengan objek lainnya. Proses pelatihan memetakan data latih menuju target latih melalui suatu rumusan (algoritma identifikasi/ klasifikasi).

### Feature Selection

Proses memilih ciri pada suatu objek agar diperoleh ciri yang optimum, yaitu ciri yang dapat digunakan untuk membedakan suatu objek dengan objek lainnya.

### Learning

Proses belajar membuat aturan klasifikasi sehingga jumlah kelas yang tumpang tindih dibuat sekecil mungkin. Kumpulan ciri dari suatu pola dinyatakan sebagai vektor ciri dalam ruang bahanmata (multi dimensi). Jadi, setiap pola

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 12	<b>Pengenalan Pola Berbasis Jarak</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 168 dari 183

dinyatakan sebagai sebuah titik dalam ruang bahanmatra. Ruang bahanmatra dibagi menjadi sejumlah uparuang (sub-ruang). Tiap uparuang dibentuk berdasarkan pola-pola yang sudah dikenali kategori dan ciri-cirinya (melalui fase pelatihan). Proses selanjutnya yaitu proses pengujian, pada proses ini rumusan yang dihasilkan dari proses pelatihan digunakan untuk memetakan data uji sehingga diperoleh data keluaran yang kemudian dibandingkan dengan target uji sehingga dapat diperoleh tingkat akurasi dari proses pengujian.

Hal-hal yang umumnya digunakan dalam proses pengenalan pola adalah sebagai berikut:

1. Persentase pembagian data untuk data latih dan data uji adalah 50%:50%, 60%:40%, 70%:30%, dan 80%:20%.
2. Tingkat akurasi proses pelatihan umumnya lebih tinggi dibandingkan dengan tingkat akurasi proses pengujian.

### Color Moments

Color moments adalah suatu metode yang digunakan untuk membedakan citra berdasarkan fitur warnanya. Dasar dari metode ini adalah asumsi bahwa distribusi warna pada sebuah citra dapat dinyatakan sebagai distribusi probabilitas. Oleh sebab itu, akurasi yang dihasilkan adalah konstan walaupun ukuran citra berubah.

Color moments disebut compact karena dapat memampatkan informasi warna citra menjadi beberapa nilai. Ekstraktor warna ini tidak memerlukan kuantisasi pada tahapan pra proses karena color moments hanya menyimpan fitur dominan pada distribusi warna di dalam database. Metode ini menggunakan tiga momen utama dari distribusi warna citra, yaitu mean, standard deviation, dan skewness, sehingga metode ini menghasilkan tiga nilai untuk masing-masing komponen warna.

Didefinisikan i komponen warna pada j piksel citra sebagai  $P_{ij}$ . Ketiga momen tersebut dapat didefinisikan sebagai berikut:

Moment 1 – Mean :

$$E = \frac{1}{N} \sum_{j=1}^N P_{ij} \quad (12.1)$$

Mean dapat dikatakan sebagai rata-rata nilai warna pada citra.

Moment 2 – Standard Deviation :

$$\sigma_i = \sqrt{\frac{1}{n} \sum_{j=1}^N (P_{ij} - E_i)^2} \quad (12.2)$$

Standard deviation adalah akar dua dari variance pada distribusi atau jangkauan tersebarnya data dari mean.

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 12	<b>Pengenalan Pola Berbasis Jarak</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 169 dari 183

Moment 3 – Skewness :

$$s_i = \sqrt[3]{\frac{1}{n} \sum_{j=1}^N (P_{ij} - E_i)^3} \quad (12.3)$$

Skewness merupakan sebuah ukuran asimetri data disekitar mean. Jika nilai skewness adalah negatif, berarti data lebih banyak tersebar di sebelah kiri rata-rata (mean) dibandingkan di sebelah kanan rata-rata mean dan begitu pula sebaliknya. Skewness dari distribusi normal adalah nol.

Adapun keterangan dari persamaan (12.1), (12.2), dan (12.3) adalah sebagai berikut :

- $E$  : Mean
- $\sigma$  : Standard deviation
- $S$  : Skewness
- $N$  : Jumlah piksel
- $i$  : indeks komponen warna sekarang (contoh : 1=H, 2=S, 3=V)
- $j$  : Urutan piksel
- $P_{ij}$  : mendefinisikan nilai i-th komponen warna pada j-th piksel citra  $ij$

Jarak dari distribusi warna citra query dengan citra database dapat dihitung dengan persamaan (12.4) di bawah ini :

$$d_{mom}(H, I) = \sum_{i=1}^r w_{i1} |E_i^1 - E_i^2| + w_{i2} |\sigma_i^1 - \sigma_i^2| + w_{i3} |s_i^1 - s_i^2| \quad (12.4)$$

Dimana:

- ( $H, I$ ) : 2 distribusi citra yang sedang dibandingkan
- $i$  : indeks komponen warna sekarang (contoh : 1=H, 2=S, 3=V)
- $r$  : jumlah komponen warna (contoh : 3)
- $E_i^1, E_i^2$  : momen pertama (mean) dari dua distribusi citra
- $\sigma_i^1, \sigma_i^2$  : momen kedua (standard deviation) dari dua distribusi citra
- $s_i^1, s_i^2$  : momen ketiga (skewness) dari dua distribusi citra
- $w_i$  : bobot (weight) dari setiap momen yang dispesifikasikan oleh user.

Citra diekstrak menggunakan color moments untuk memperoleh informasi ciri warna dari citra tersebut.

### Konversi ke HSV

Citra terlebih dahulu dikonversi ke dalam ruang warna HSV sebelum diekstraksi dengan color moments. Hal ini dilakukan karena HSV lebih sesuai dengan persepsi visual manusia daripada RGB.

## **PENGENALAN CITRA DAN POLA**

Pertemuan : 12

**Pengenalan Pola Berbasis Jarak**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 170 dari 183

Citra awal dengan tiga matriks komponen R, G dan B. Berikut ini contoh adalah perhitungan konversi RGB ke HSV berdasarkan persamaan (12.1):

- Mengubah R,G,B ke dalam range [0,1]

$$R = \frac{185}{255} = 0.7255, \quad G = \frac{113}{255} = 0.4431, \quad B = \frac{57}{255} = 0.2235$$

- Menghitung nilai  $\delta$

$$\text{MAX} = \max(R, G, B) = 0.7255$$

$$\text{MIN} = \min(R, G, B) = 0.2235$$

$$\text{MAX} - \text{MIN} = 0.7255 - 0.2235 = 0.502$$

- Menghitung nilai hue

Karena MAX = R, maka :

$$H = 60 \left( \frac{G - B}{\delta} \right) = 60 \left( \frac{0.4431 - 0.2235}{0.502} \right) = 26.247$$

Nilai hue yang masih dalam range [0,360] diubah ke dalam range [0,1], jadi :

$$H = \frac{26.247}{360} = 0.073$$

- Menghitung nilai saturation

Karena MAX ≠ 0, maka :

$$S = \frac{\delta}{\text{MAX}} = \frac{0.502}{0.7255} = 0.6919$$

- Menghitung nilai value

$$\text{Value} = \text{MAX} = 0.7255$$

Jadi, piksel(1,1) memiliki nilai H=0.073 ; S=0.6919 ; dan V=0.7255. dilakukan pada piksel-piksel berikutnya. Berikut adalah contoh matriks HSV

## **PENGENALAN CITRA DAN POLA**

Pertemuan : 12

**Pengenalan Pola Berbasis Jarak**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 171 dari 183

Tabel 1 Nilai *Hue*

0.073	0.046	0.0629	0.0659	0.069
0.051	0.057	0.0471	0.0448	0.055
0.054	0.058	0.0459	0.0382	0.047
0.053	0.051	0.0429	0.0373	0.033
0.121	0.046	0.0338	0.029	0.029

Tabel 2 Nilai *Saturation*

0.6919	0.8340	0.8193	0.8300	0.7430
0.8155	0.9440	0.9409	0.9784	0.8506
0.7968	0.9160	0.9495	0.9412	0.8589
0.7593	0.8934	0.8800	0.9178	0.8940
0.7692	0.9144	0.9258	0.8812	0.8418

Tabel 3 Nilai *Value*

0.7255	0.9922	0.9765	0.9686	0.7020
0.9137	0.9804	0.9294	0.9098	0.9451
0.9843	0.9804	0.8549	0.8000	0.9451
0.9451	0.9569	0.8824	0.8588	0.8510
0.4078	0.8706	0.8980	0.7922	0.6941

### Perhitungan 3 Momen dari H, S dan V

Dalam ruang warna HSV, rentang hue berada diantara [0..1] dan bersifat sirkular (wrap around) sehingga  $0 \equiv 1$ . Dengan demikian perhitungan rata-rata hue harus dilakukan dengan cara khusus. Mula-mula, nilai hue dipetakan kedalam interval koordinat polar  $(0..2\pi)$  radians, sehingga dapat dianggap sebagai sebuah titik dalam koordinat polar  $(\theta, 1)$ , dimana  $\theta = 2\pi h$  dan  $h$  adalah nilai hue-nya.

Titik dalam koordinat polar tersebut kemudian dikonversikan ke dalam koordinat kartesius sedemikian rupa sehingga untuk setiap hue terdapat sebuah titik  $(x, y)$  dimana  $x = \cos(\theta)$  dan  $y = \sin(\theta)$ .

Rata-rata  $(x, y)$  ini digunakan untuk mendapatkan sudut dalam koordinat polar dengan formula  $\theta = \arctan(y/x)$  dimana kemudian dipetakan ke dalam interval  $(0..2\pi)$  dengan menambahkan atau menguranginya dengan kelipatan dari  $2\pi$ . Selanjutnya dipetakan ke interval  $[0..1]$  dengan cara membaginya dengan  $2\pi$ . Dengan demikian rata-rata hue dihitung sebagai  $h = \theta / 2\pi$ . Sedangkan perhitungan

## PENGENALAN CITRA DAN POLA

Pertemuan : 12

**Pengenalan Pola Berbasis Jarak**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 172 dari 183

mean pada komponen S dan V dilakukan sesuai  $2\pi$  dengan persamaan (12.2). Menghitung mean S berdasarkan pada persamaan (12.1):

$$E = \frac{1}{N} \sum_{j=1}^N P_{ij} = \frac{0.6919 + 0.8155 + 0.7968 + \dots + 0.8418}{25} = 0.8635$$

Menghitung standard deviation S sesuai dengan persamaan (12.2):

$$\begin{aligned} \sigma_i &= \sqrt{\frac{1}{n} \sum_{j=1}^N (P_{ij} - E_i)^3} \\ &= \sqrt{\frac{(0.6919 - 0.8635)^2 + (0.8155 - 0.8635)^2 + \dots + (0.8414 - 0.8635)^2}{25}} = 0.0718 \end{aligned}$$

Menghitung nilai skewness S berdasarkan persamaan (12.3) :

$$\begin{aligned} s_i &= \sqrt[3]{\frac{1}{n} \sum_{j=1}^N (P_{ij} - E_i)^3} \\ &= \sqrt[3]{\frac{(0.6919 - 0.8635)^3 + (0.8155 - 0.8635)^3 + \dots + (0.8414 - 0.8635)^3}{25}} \\ &= -0.0584 \end{aligned}$$

Untuk mendapatkan 3 momen pada komponen hue dan value dilakukan perhitungan yang sama seperti pada komponen saturation. Hasil akhir adalah sembilan nilai vektor ciri warna (tabel 4). Sembilan nilai inilah yang digunakan untuk membandingkan kesamaan (similarity) dengan citra yang lain berdasarkan fitur warna.

Tabel 4 Nilai Vektor Warna Citra *Query*

	Hue	Saturation	Value
<i>mean</i>	0.0515	0.8635	0.8706
<i>std</i>	0.0182	0.0718	0.1282
<i>skewness</i>	0.0232	-0.0584	-0.16

### Metode Perhitungan Jarak Euclidean Distance

Dalam sistem koordinat citra dua dimensi, jarak antara dua objek dapat diukur menggunakan persamaan euclidean distance. Euclidean distance adalah suatu metode pencarian kedekatan nilai jarak dari dua buah variabel. Euclidean

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<b>Fakultas Teknologi dan Informatika</b> <b>Program Studi Teknik Informatika</b> <b>PENGENALAN CITRA DAN POLA</b>		
Pertemuan : 12	<b>Pengenalan Pola Berbasis Jarak</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 173 dari 183

diperkenalkan oleh seorang matematikawan dari Yunani. Untuk mempelajari hubungan antara sudut dan jarak. Euclidean biasanya diterapkan pada dua dimensi dan 3 dimensi tapi juga sederhana jika diterapkan pada dimensi yang lebih tinggi. Jarak euclidean merupakan jarak yang paling umum yang digunakan untuk data numerik, untuk dua titik data  $x$  dan  $y$  dalam ruang  $d$ - dimensi. Metode jarak euclidean berdasarkan lima ciri: intensitas warna ( $\sigma$ ), nilai rata-rata ( $\mu$ ), entropi ( $e$ ), energi ( $E$ ), dan homogeiniti ( $H$ ). Bentuk umum euclidean distance ( $d$ ) dapat diperoleh dengan:

$$x = (x_1, x_2, \dots, x_n)$$

$$y = (y_1, y_2, \dots, y_n)$$

$$d(x, y) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + \dots + (y_n - x_n)^2}$$

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

(a) (b) Gambar 12.1 Contoh Matriks Jarak: Posisi Koordinat (a), Setelah Dihitung (b)

Sumber: Google Images

Pada dua dimensi, misalkan ada dua titik koordinat  $(X_{11}, X_{21})$  dan  $(X_{12}, X_{22})$ , maka dapat diketahui jarak euclidean ( $d$ ) tersebut seperti contoh berikut:

$$d(P_1, P_2) = \sqrt{(x_{21} - x_{11})^2 + (x_{22} - x_{12})^2}$$

Mengacu pada Gambar 12.1 terdapat empat poin koordinat citra, dengan masing-masing koordinat  $(x,y)$  memiliki nilai acak kemudian dicari nilai jarak dengan menggunakan rumus. Hasil perhitungan masing-masing koordinat yang direpresentasikan dalam bentuk matriks (b). Rumus tersebut merupakan rumus untuk dua dimensi.

Proses yang dilakukan dengan membandingkan kedekatan nilai jarak dari dua buah variabel yaitu antara citra uji dengan citra acuan untuk mencari nilai jarak terdekat. Hasil perhitungan yang digunakan sebagai perbandingan adalah nilai yang paling kecil (jarak yang paling dekat).

### Jarak Manhattan (Manhattan Distance )

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	---	------------------

## PENGENALAN CITRA DAN POLA

Pertemuan : 12

**Pengenalan Pola Berbasis Jarak**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 174 dari 183

Manhattan distance/city block distance merupakan salah satu teknik yang sering digunakan untuk menentukan kesamaan antara dua buah objek. Manhattan distance merupakan salah satu formula untuk menghitung jarak terdekat antara dua titik. Pengukuran ini dihasilkan berdasarkan penjumlahan jarak selisih antara dua buah obyek dan hasil yang didapatkan dari manhattan distance bernilai mutlak. Dimana manhattan distance melakukan perhitungan jarak dengan cara tegak lurus.

Manhattan distance juga merupakan salah satu pengukuran yang paling banyak digunakan meliputi penggantian perbedaan kuadrat dengan menjumlahkan perbedaan absolut dari variabel. Prosedur ini disebut juga sebagai blok absolut.

$$d(x, y) = L_p = i(x, y) = \sum_i^n \|x_i - y_i\|$$

### Hamming Distance

Hamming distance adalah jarak antara dua string dengan panjang yang sama dengan jumlah total posisi di mana karakter yang tidak sesuai dalam dua string yang berbeda. Jika ukuran dua string tidak sama maka jarak hamming antara mereka adalah tak terbatas. Hamming distance adalah suatu metode untuk mengukur tingkat error antara histogram citra dengan probabilistik distribusi di mana parameter diperoleh dengan estimasi parameter. Formula yang di bentuk oleh hamming sebagai berikut:

$$d_{ij} = q + r$$

- q = Nilai variabel positif pada objek pertama dan nilai variabel negatif pada objek kedua
- r = Nilai variabel negatif pada objek pertama dan nilai variabel positif pada objek kedua.

### E. Alat/Instrument/Aparatus/Bahan

1. LCD Proyektor
2. Komputer
3. Python
4. Code editor (Visual Studio Code)
5. Python Library

### F. Keselamatan Kerja

1. Sebelum memulai praktikum, mahasiswa memahami tata tertib dan keselamatan di Laboratorium
2. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)

## **PENGENALAN CITRA DAN POLA**

Pertemuan : 12	<b>Pengenalan Pola Berbasis Jarak</b>	100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022

Hal 175 dari 183

3. Menggunakan alat dan bahan sesuai dengan fungsinya
4. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera lapor ke asisten terkait atau dapat langsung melapor ke laboran
5. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai
6. Bacalah petunjuk penggunaan alat dan bahan dengan baik dan benar
7. Bila mendapatkan masalah atau kesulitan saat praktik tanyakanlah kepada dosen atau asisten lab
8. Tidak bercanda atau makan dan minum saat praktik
9. Rapikan dan kembalikan alat dan bahan sesuai tempatnya

### G. Lembar Kerja

#### **Kegiatan Praktikum 1: Implementasi KNN menggunakan Euclidean Distance**

Algoritma k-Nearest Neighbors atau disingkat KNN adalah teknik yang sangat sederhana. Seluruh dataset pelatihan disimpan. Ketika prediksi diperlukan, k-tetangga yang paling mirip dengan data baru dari dataset pelatihan kemudian didapatkan. Dari tetangga ini, prediksi yang diringkas dibuat.

Kesamaan antara data record dapat diukur dengan berbagai cara. Masalah atau metode khusus data dapat digunakan. Umumnya, dengan data tabular, titik awal yang baik adalah metode jarak Euclidean.

Tahap pada KNN dapat dimulai dari menghitung nilai Euclidean Distance lalu mencari data tetangga terdekat dan menentukan nilai prediksi. Siapkan sebuah file python baru dan copy paste kode program berikut.

```
# Example of making predictions
from math import sqrt

# calculate the Euclidean distance between two vectors
def euclidean_distance(row1, row2):
    distance = 0.0
    for i in range(len(row1)-1):
        distance += (row1[i] - row2[i])**2
    return sqrt(distance)

# Locate the most similar neighbors
def get_neighbors(train, test_row, num_neighbors):
    distances = list()
    for train_row in train:
        dist = euclidean_distance(test_row, train_row)
        distances.append((train_row, dist))
    distances.sort(key=lambda tup: tup[1])
    neighbors = list()
    return neighbors[:num_neighbors]
```

## PENGENALAN CITRA DAN POLA

Pertemuan : 12

**Pengenalan Pola Berbasis Jarak**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 176 dari 183

```

for i in range(num_neighbors):
    neighbors.append(distances[i][0])
return neighbors

# Make a classification prediction with neighbors
def predict_classification(train, test_row, num_neighbors):
    neighbors = get_neighbors(train, test_row, num_neighbors)
    output_values = [row[-1] for row in neighbors]
    prediction = max(set(output_values), key=output_values.count)
    return prediction

# Test distance function
dataset = [[2.7810836,2.550537003,0],
           [1.465489372,2.362125076,0],
           [3.396561688,4.400293529,0],
           [1.38807019,1.850220317,0],
           [3.06407232,3.005305973,0],
           [7.627531214,2.759262235,1],
           [5.332441248,2.088626775,1],
           [6.922596716,1.77106367,1],
           [8.675418651,-0.242068655,1],
           [7.673756466,3.508563011,1]]
prediction = predict_classification(dataset, dataset[0], 3)
print('Expected %d, Got %d.' % (dataset[0][-1], prediction))

```

### Kegiatan Praktikum 2: Iris Flower Species Case Study

Download dataset iris pada link berikut, letakan file satu direktori dengan file python.

Dataset iris: <https://drive.google.com/file/d/1DnmCev2T2i31qXz5NkZpUHtE8S-Snx1K/view?usp=sharing>

Copy kode program berikut dan jalankan pada code editor. Lakukan pengujian dengan mencoba data baru.

```

# k-nearest neighbors on the Iris Flowers Dataset
from random import seed
from random import randrange
from csv import reader
from math import sqrt

# Load a CSV file
def load_csv(filename):
    dataset = list()
    with open(filename, 'r') as file:
        csv_reader = reader(file)

```

## **PENGENALAN CITRA DAN POLA**

Pertemuan : 12

**Pengenalan Pola Berbasis Jarak**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 177 dari 183

```

for row in csv_reader:
    if not row:
        continue
    dataset.append(row)
return dataset

# Convert string column to float
def str_column_to_float(dataset, column):
    for row in dataset:
        row[column] = float(row[column].strip())

# Convert string column to integer
def str_column_to_int(dataset, column):
    class_values = [row[column] for row in dataset]
    unique = set(class_values)
    lookup = dict()
    for i, value in enumerate(unique):
        lookup[value] = i
        print('[%s] => %d' % (value, i))
    for row in dataset:
        row[column] = lookup[row[column]]
    return lookup

# Find the min and max values for each column
def dataset_minmax(dataset):
    minmax = list()
    for i in range(len(dataset[0])):
        col_values = [row[i] for row in dataset]
        value_min = min(col_values)
        value_max = max(col_values)
        minmax.append([value_min, value_max])
    return minmax

# Rescale dataset columns to the range 0-1
def normalize_dataset(dataset, minmax):
    for row in dataset:
        for i in range(len(row)):
            row[i] = (row[i] - minmax[i][0]) / (minmax[i][1] - minmax[i][0])

# Split a dataset into k folds
def cross_validation_split(dataset, n_folds):
    dataset_split = list()
    dataset_copy = list(dataset)
    fold_size = int(len(dataset) / n_folds)
    for _ in range(n_folds):
        fold = list()
        index = random.randrange(0, len(dataset_copy), 1)
        fold.append(dataset_copy.pop(index))
        if len(dataset_copy) == 0:
            break
        dataset_split.append(fold)
    return dataset_split

```

## **PENGENALAN CITRA DAN POLA**

Pertemuan : 12

**Pengenalan Pola Berbasis Jarak**

100 menit

Kode MK: TIW-043

Rev : 00

Tgl : 01/03/2022

Hal 178 dari 183

```

fold = list()
while len(fold) < fold_size:
    index = randrange(len(dataset_copy))
    fold.append(dataset_copy.pop(index))
dataset_split.append(fold)
return dataset_split

# Calculate accuracy percentage
def accuracy_metric(actual, predicted):
    correct = 0
    for i in range(len(actual)):
        if actual[i] == predicted[i]:
            correct += 1
    return correct / float(len(actual)) * 100.0

# Evaluate an algorithm using a cross validation split
def evaluate_algorithm(dataset, algorithm, n_folds, *args):
    folds = cross_validation_split(dataset, n_folds)
    scores = list()
    for fold in folds:
        train_set = list(folds)
        train_set.remove(fold)
        train_set = sum(train_set, [])
        test_set = list()
        for row in fold:
            row_copy = list(row)
            test_set.append(row_copy)
            row_copy[-1] = None
        predicted = algorithm(train_set, test_set, *args)
        actual = [row[-1] for row in fold]
        accuracy = accuracy_metric(actual, predicted)
        scores.append(accuracy)
    return scores

# Calculate the Euclidean distance between two vectors
def euclidean_distance(row1, row2):
    distance = 0.0
    for i in range(len(row1)-1):
        distance += (row1[i] - row2[i])**2
    return sqrt(distance)

# Locate the most similar neighbors
def get_neighbors(train, test_row, num_neighbors):
    distances = list()
    for train_row in train:
        distance = euclidean_distance(test_row, train_row)
        distances.append((train_row, distance))
    distances.sort(key=lambda tup: tup[1])
    return distances[:num_neighbors]

```

## **PENGENALAN CITRA DAN POLA**

Pertemuan : 12	<b>Pengenalan Pola Berbasis Jarak</b>	100 menit
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022

Hal 179 dari 183

```

dist = euclidean_distance(test_row, train_row)
distances.append((train_row, dist))
distances.sort(key=lambda tup: tup[1])
neighbors = list()
for i in range(num_neighbors):
    neighbors.append(distances[i][0])
return neighbors

# Make a prediction with neighbors
def predict_classification(train, test_row, num_neighbors):
    neighbors = get_neighbors(train, test_row, num_neighbors)
    output_values = [row[-1] for row in neighbors]
    prediction = max(set(output_values), key=output_values.count)
    return prediction

# kNN Algorithm
def k_nearest_neighbors(train, test, num_neighbors):
    predictions = list()
    for row in test:
        output = predict_classification(train, row, num_neighbors)
        predictions.append(output)
    return(predictions)

# Test the KNN on the Iris Flowers dataset
seed(1)
filename = 'iris.csv'
dataset = load_csv(filename)
for i in range(len(dataset[0])-1):
    str_column_to_float(dataset, i)
# convert class column to integers
str_column_to_int(dataset, len(dataset[0])-1)
# evaluate algorithm
n_folds = 5
num_neighbors = 5
scores = evaluate_algorithm(dataset, k_nearest_neighbors, n_folds, num_neighbors)
print('Scores: %s' % scores)
print('Mean Accuracy: %.3f%%' % (sum(scores)/float(len(scores))))

# define a new record
row = [5.1, 3.5, 1.4, 0.2]
# predict the label
label = predict_classification(dataset, row, num_neighbors)
print('Data=%s, Predicted: %s' % (row, label))

```

 <b>INSTIKI</b> <small>INSTITUT BISNIS DAN TEKNOLOGI INDONESIA</small>	<p style="text-align: center;"><b>Fakultas Teknologi dan Informatika</b></p> <p style="text-align: center;"><b>Program Studi Teknik Informatika</b></p> <p style="text-align: center;"><b>PENGENALAN CITRA DAN POLA</b></p>		
Pertemuan : 12	<b>Pengenalan Pola Berbasis Jarak</b>	100 menit	
Kode MK: TIW-043	Rev : 00	Tgl : 01/03/2022	Hal 180 dari 183

## H. Tugas Praktikum

Berdasarkan data pada kegiatan praktikum 1 gunakan metode Manhattan Distance untuk mencari nilai kedekatan dan menentukan prediksi.

## I. Daftar Referensi

- Munir, Rinaldi. 2019. "Pengantar Pengolahan Citra (Bagian 2) - IF4073 Interpretasi dan Pengolahan Citra." Institut Teknologi Bandung.
- <http://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2019-2020/02-Pengantar-Pengolahan-Citra-Bag2.pdf> (Agustus 10, 2020).

Dibuat oleh :	Dilarang memperbanyak sebagian atau seluruh isi dokumen tanpa ijin tertulis dari Program Studi Teknik Informatika Institut Bisnis dan Teknologi Indonesia (INSTIKI)	Diperiksa oleh :
---------------	--	------------------



**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
PROGRAM STUDI TEKNIK INFORMATIKA**

Jl. Tukad Pakerisan No. 97 Denpasar

