

PCD Buku

by Suhendro Irianto

Submission date: 16-Feb-2020 04:01PM (UTC+0800)

Submission ID: 1258149655

File name: PCD_buku.pdf (3.19M)

Word count: 14885

Character count: 87606

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/311708107>

PENGOLAHAN CITRA DIGITAL

Book · March 2014

CITATIONS
0

READS
4,741

1 author:



Suhendro Irianto
Darmajaya Informatics and Business Institute

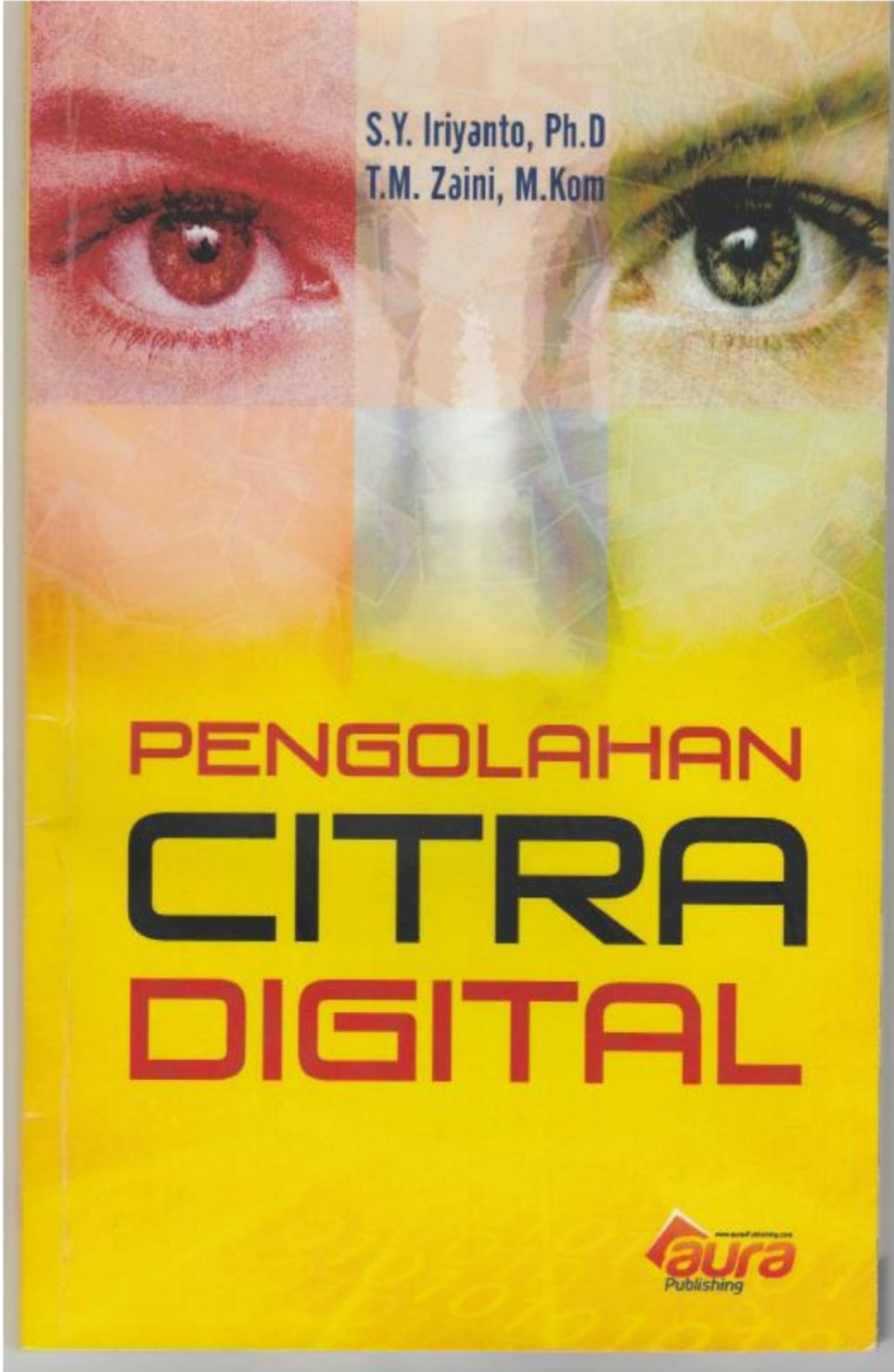
14 PUBLICATIONS 12 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Content Based Image Retrieval View project



S.Y. Iriyanto, Ph.D
T.M. Zaini, M.Kom

PENGOLAHAN
CITRA
DIGITAL

 **laura**
Publishing

Perpustakaan Nasional RI: Katalog Dalam Terbitan (KDT)

S.Y. Iriyanto, Ph.D
T.M. Zaini, M.Kom

PENGOLAHAN CITRA DIGITAL

Rancang Sampul & Penata Isi
YansDesign

ISBN : 978-602-1297-22-3

Cetakan Maret 2014
viii + 94 hlm. ; 15,5 x 23 cm

Penerbit
Anugrah Utama Raharja (AURA)
Printing & Publishing
ANGGOTA IKAPI
No. 003 / LPU / 2013

Alamat
Jl. Prof. Dr. Soemantri Brojonegoro, Komplek Unila
Raja Basa Bandar Lampung 081281430268
www.aura-publishing.com

Hak Cipta dilindungi Undang-Undang
All Rights Reserved.

Dilarang mengutip atau memperbanyak sebagian
atau seluruh isi buku ini tanpa izin tertulis dari penerbit

Prakata

Buku ajar ini untuk memperkenalkan kepada mahasiswa tentang pengetahuan yang berhubungan dengan image processing dan aplikasinya dalam kehidupan sehari-hari. Buku ini menyajikan pemahaman dan konsep dan praktek pengolahan citra. Tujuan dari buku ini adalah memberikan kepada mahasiswa pengetahuan dasar tentang pengolahan citra digital dan penerapannya dengan menggunakan programming MatLab.

Bab pertama dan kedua membahas tentang definisi dan pengertian citra digital, noise citra, model warna dan representasi citra digital. Analisa citra digital, jenis-jenis citra digital, dan fitur citra digital dibicarakan pada bab tiga, empat dan lima. Akhirnya pada bab enam dibicarakan tentang JPEG dan JPEG2000, sementara bab terakhir dibicarakan secara rinci teknik Content Based Image Retrieval (CBIR) yang menyajikan banyak contoh-contoh aplikasi yang banyak digunakan oleh para peneliti bidang image processing termasuk yang telah dikerjakan oleh penulis. Dalam buku ini soal-soal latihan dikerjakan dengan menggunakan program MatLab.

Sangat diharapkan buku ajar ini dapat membantu para mahasiswa Sarjana strata satu khususnya dan mahasiswa pascasarjana yang tertarik dengan bidang image processing pada jurusan teknik informatika, sistem informasi, dan bidang-bidang lain seperti teknik sipil, pertanian, kehutanan, dan bidang-bidang lain yang memanfaatkan image processing.

Bandar Lampung, Maret 2014
Penulis



DAFTAR ISI

	Halaman
PENGANTAR	i
DAFTAR ISI	ii
DAFTAR GAMBAR	iii
BAB I	
CITRA DIGITAL	1
1.1 Pengertian dan Definisi Citra Digital	1
1.2 Common Values	4
1.3 Full Color Processing	4
1.4 Konversi RGB ke dan dari YUV	9
1.5 Kromatisitas koordinat CIE	12
1.6 Perhitungan Kromatisitas Koordinat CIE	13
1.7 Pseudocolor processing	16
1.8. Model Warna	17
1.8.1 Model RGB	18
1.8.2 Model CMY dan Gamut	19
1.8.3 Model HIS	20
1.9. Image Noise	22
1.9.1. Impulse Noise	24
1.9.2. Additive With Gaussian Noise	24
BAB II.	
REPRESENTASI CITRA DIGITAL	27
2.1. Terbentuknya Citra Digital	27
2.2. Sampling dan Quantization Citra 2 D	28
2.3. Distorsi dan Blurring	31
2.4. Representasi spatial citra digital	32
BAB III.	
ANALISA CITRA DIGITAL	32
3.1. Edge Detection	32
3.2. Menelusuri Batas	34
3.3. Segmetasi	36

3.4. Content Based Image Retrieval	41
3.5. Piksel dan Koefisien DCT	43
3.6. Median Filter	44
3.7. Mask Citra Digital	45
3.8. Morfologi	46
3.9. Operasi Morfologi	46
3.9.1. Translasi	46
3.9.2. Dilasi	46
3.9.3. Erosi	47
3.9.4. Opening	47
3.10. Structuring Element	47
3.11. Filtering	48
3.12. Analisa Bentuk	49
3.12.1 Including Boxes	50
3.12.2. Fourier Descriptor	51
3.12.3 Pengenalan Bentuk	52
3.13. Representasi Matrik dari bentuk	55
BAB IV	
JENIS-JENIS CITRA DIGITAL	58
4.1 Jenis atau tipe citra digital	58
4.2 Konversi antar citra	60
4.2.1 rgb2gray	60
4.2.2 im2bw	61
4.2.3. gray2ind	61
4.2.4.gray2ind,	62
im2bw,ind2gray,rg2gray,rg2ind,mat2gray	

	4.2.5. rgb2ind	62
	4.26. mat2gray	62
BAB V	FITUR CITRA DIGITAL	64
	5.1 Fitur Tekstur	64
	5.2. Tepi (edge)	67
BAB VI	JPEG VS JPEG2000	68
	6.1. JPEG	68
	6.2. Lossy Compression	68
	6.3. Discrete Cosine Transform	71
	6.4. DCT Analysis	71
	6.5. Zig Zag Scanning	72
	6.6. Modified Huffman Coding	72
	6.7. JPEG 2000	76
BAB VII	CONTENT BASED IMAGE RETRIEVAL	80
	7.1. Database Citra Digital	80
	7.2. Fitur Citra	81
	7.3 Query-by-Example	81
	7.4. Matching Image Query	82
	7.1 CBIR dalam compressed domain	83
DAFTAR PUSTAKA		

CITRA DIGITAL

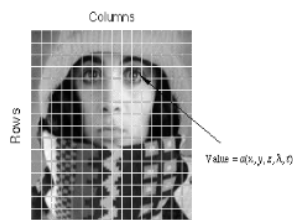
Tujuan intruksional umum : Memahami konsep dan ilmu pegolahan citra digital dan mengerti macam pengolahan citra

Tujuan instruksional khusus :

- Memahami dan mengerti definisi dan konsep citra digital
- Memahami bagaimana proses pengolahan citra dan bagaimamana apalikasi pengolahan citra
- Mengerti perbedaan pengolahan citra secaratradisional dan modern

1.1 Pengertian dan definisi citra digital

Sebuah citra digital $a[m, n]$ dapat dijelaskan sebagai ruang diskrit 2D yang berasal dari sebuah citra analog $a(x, y)$ menjadi citra kontinu 2D melalui proses sampling yang sering disebut sebagai digitalisasi, proses digitalisasi ditunjukkan pada Gambar 1.1. Jika citra 2D kontinu $a(x, y)$ dibagi menjadi N baris dan M kolom, maka perpotongan baris dan kolom disebut piksel. Nilai yang diberikan ke integer koordinat $[m, n]$ dengan $\{m = 0, 1, 2, \dots, M-1\}$ dan $\{n = 0, 1, 2, \dots, N-1\}$ adalah $[m, n]$. Dalam banyak hal suatu citra (x, y) , kita dapat menganggap sebagai sinyal fisik sensor 2D - sebenarnya adalah fungsi dari banyak variabel termasuk kedalaman (z), warna (λ), dan waktu (t).



Gambar 1.1: Digitasi dari citra kontinu. Piksel pada koordinat $[m = 10, n = 3]$ memiliki nilai integer kecerahan 110.

Citra yang ditunjukkan pada gambar 1.1 telah dibagi menjadi $N = 16$ baris dan $M = 16$ kolom. Nilai yang diberikan untuk setiap piksel (kecerahan rata-rata) dalam piksel dibulatkan ke nilai integer terdekat. Proses

ini mewakili amplitudo dari sinyal 2D pada koordinat diberikan sebagai nilai integer dengan tingkat L abu-abu yang berbeda dan biasanya disebut sebagai proses kuantisasi amplitudo atau disingkat disebut kuantisasi.

Citra atau *image* adalah suatu cahaya pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, *scanner*, dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam. Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat:

- optik berupa foto
- analog berupa sinyal video seperti citra pada monitor televisi
- digital yang dapat langsung disimpan pada suatu pita magnetik.



(a) Lena



(b) Kapal

Gambar 1.2. Contoh citra digital

Citra juga dapat dianggap sebagai objek dua dimensi yang dihasilkan dari citra analog yang kontinu dan kemudian dapat diubah menjadi citra diskrit melalui proses *sampling*. *Sampling* adalah pemilihan titik-titik

yang dianggap mewakili citra digital yang diberikan. Sampling dapat juga didefinisikan sebagai pengurangan signal kontinu menjadi signal diskrit. Citra digital dapat juga didefinisikan sebagai fungsi dua dimensi $f(x,y)$ dimana x dan y adalah koordinat dan amplitudo gelombang dimana f adalah merupakan suatu intensitas Gray (Gy) level dalam citra pada titik. Citra digital terdiri dari sejumlah elemen, dimana setiap elemen disebut sebagai elemen citra atau pels atau piksel.



Citra digital asli

Citra yang telah di-sampling

Gambar 1.3: Contoh citra sampling

Pada umumnya citra digital berbentuk empat persegi panjang dengan lebar dan panjang tertentu. Ukuran citra biasanya diukur dalam jumlah titik atau piksel, dimana setiap piksel mempunyai koordinat menurut letaknya di dalam citra digital. Koordinat ini biasanya dinyatakan dalam bilangan integer yang besarnya dari 0 sampai dengan 1 tergantung dari sistem yang digunakan. Setiap piksel berisi nilai yang mewakili informasi apa yang ada dalam piksel. Citra (*image*) juga dapat didefinisikan sebagai fungsi dua dimensi $f(x,y)$ di mana x dan y adalah koordinat spasial dan amplitudo f pada setiap pasang (x,y) disebut intensitas (*gray level*) citra pada titik tersebut. Jika x dan y berhingga (*finite*) dan diskrit (tdk kontinu) maka disebut citra digital. Citra digital terdiri dari sejumlah elemen berhingga yang masing-masing mempunyai lokasi dan nilai, Elemen-elemen x dan y disebut elemen citra atau *pels* atau *piksel*.

1.2 Common Values

Terdapat beberapa nilai parameter standar yang ada dalam pengolahan citra digital. Nilai-nilai standar ini dapat disebabkan oleh standar video, parameter dimaksudkan agar circuit citra digital tetap sederhana.

Tabel 1. 1: Nilai-nilai standar parameter digital yang ada

<i>Parameter</i>	<i>Symbol</i>	<i>Typical values</i>
Baris	<i>N</i>	256,512,525,625,1024,1035
Kolom	<i>M</i>	256,512,768,1024,1320
Tingkat warna keabuan	<i>L</i>	2,64,256,1024,4096,16384

Jumlah tingkat keabuan yang berbeda biasanya merupakan pangkat dari 2, yaitu $L = 2^\beta$ dimana β adalah jumlah bit dalam representasi biner dari tingkat kecerahan. Ketika $\beta > 1$, maka citra yang dimaksud adalah citra *gray-level*, ketika $\beta = 1$ maka citra biner. Dalam sebuah citra biner hanya ada dua tingkat abu-abu yang ada, yaitu hitam dan putih atau "0" dan "1".

1.3 Full Color Processing

Cahaya *achrominance* merupakan cahaya yang tidak berwarna, hanya menggunakan intensitas yang diukur dengan tingkat keabuan atau *grayscale*. Misalnya cahaya atau warna TV hitam-putih dan citra monokrom. Sedangkan cahaya atau warna kromatik mempunyai panjang gelombang antara 400~700 nm. Cahaya *chrominance* adalah cahaya yang terdiri dari tiga satuan yang digunakan untuk mendeskripsikan kualitas dari sumber cahaya akromatik, yaitu: *Radiance*, *Luminance* dan *Brightness*.



Gambar 1.4. Komponen Luminance, Hue, Saturation

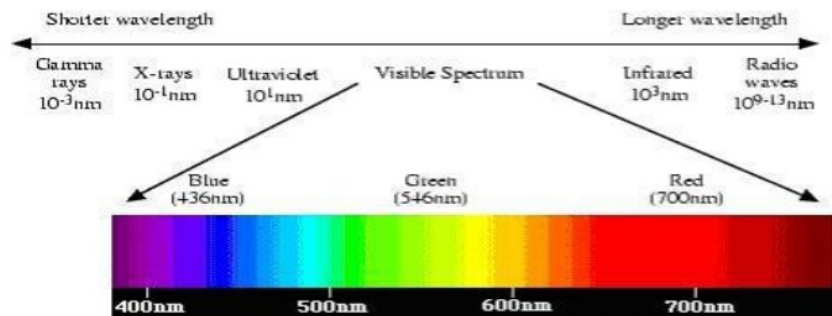
Radiance adalah jumlah energi yang memancar dari sumber cahaya (dalam satuan watt). *Luminance* adalah jumlah energi yang diterima oleh observer dari sumber cahaya (dalam satuan lumens, lm). contoh: sinar inframerah memiliki radiansi yang besar tapi nyaris tidak dapat dilihat oleh mata manusia. *Brightness* adalah deskriptor yang subjektif, mirip dengan pengertian intensitas pada akromatik, adalah satu faktor penentu dalam menggambarkan warna.

Sistem visual manusia dapat membedakan ratusan ribu warna dan intensitas, tetapi hanya 100 macam citra *gray scale* yang dapat dibedakan. Oleh sebab itu, dalam suatu citra, masih banyak informasi lainnya yang ada pada warna, dan informasi tersebut juga dapat digunakan untuk menyederhanakan analisis citra, misalkan identifikasi objek dan ekstraksi warna. Terdapat tiga kuantisasi untuk mencitrakan warna, yaitu:

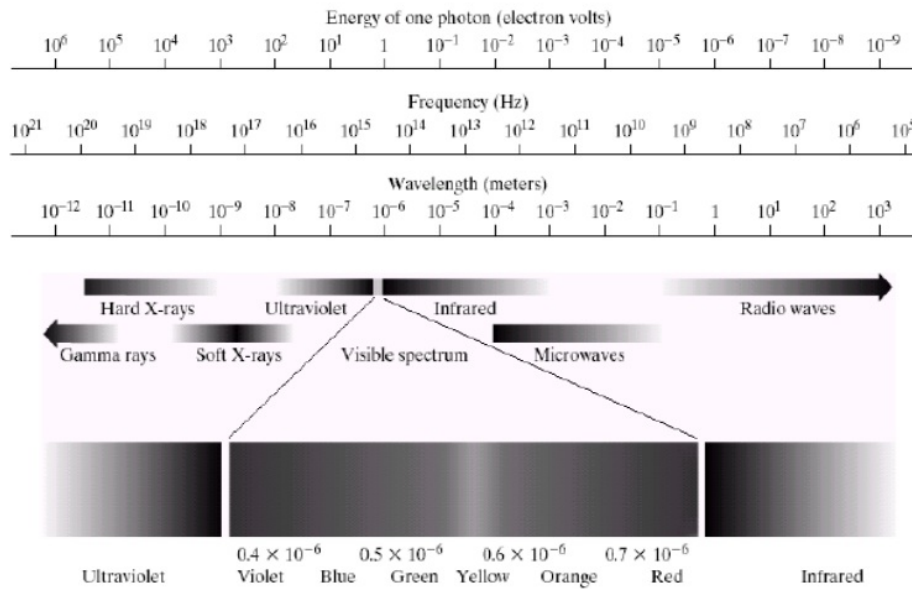
- **Hue** ditentukan oleh dominan panjang gelombang. Warna yang dapat dilihat oleh mata memiliki panjang gelombang antara 400 nm (violet) - 700 nm (red) pada spektrum electromagnetic seperti pada Gambar 1.5.
- **Saturation** ditentukan oleh tingkat kemurnian, dan tergantung pada jumlah sinar putih yang **tercampur** dengan *hue*. Suatu warna *hue* murni adalah secara penuh tersaturasi, yaitu tidak ada sinar putih yang tercampur. Hue dan saturation digabungkan menentukan chromaticity suatu

warna. Intensitas ditentukan oleh jumlah sinar yang diserap. Semakin banyak sinar yang diserap semakin banyak tinggi intensitas warnanya.

- **Chromatic** tidak memiliki warna, tetapi mempunyai atribut intensitas. Tingkat keabuan (*Greylevel*) merupakan ukuran intensitas energi, sehingga merupakan suatu kuantitas fisik. Sementara itu *brightness* atau *luminance* ditentukan oleh persepsi warna penglihatan mata manusia. Apabila diberikan sinar biru dan hijau dengan intensitas yang sama, sinar biru diterima lebih gelap dibandingkan sinar hijau. Sehingga dapat dikatakan bahwa persepsi intensitas manusia adalah non-linear, misalkan perubahan intensitas yang dinormalisasi dari 0.1 ke 0.11 dan 0.5 ke 0.55 akan diterima dengan perubahan tingkat kecerahan (*brightness*) yang sama.



Gambar 1.5: Spektrum yang dapat dilihat oleh mata kita



Gambar 1.6: Spektrum elektromagnetik. Spektrum yang dapat dilihat (visible) ditunjukkan dengan citra yang di-zoom.

Dari gambar 1.6 dapat disimpulkan bahwa cahaya yang dapat dilihat (*visible spectrum*) mempunyai ciri-ciri sebagai berikut:

- Karakteristik kisaran antara: $0.43\mu\text{m}$ (violet)- $0.78\mu\text{m}$ (red)
- Enam bands: *violet, blue, green, yellow, orange, red*
- Warna dari suatu objek ditentukan oleh sifat dari cahaya yang direfleksikan oleh objek.
- Cahaya *monochromatic (gray level)*
- Tiga unsur ukuran cahaya chromatic (*Radiance, luminance and brightness*)

Pada kenyataan citra digital kebanyakan tidak monokrom, tetapi citra berwarna. Sebagai contoh citra spesimen jaringan yang mempunyai banyak warna digunakan untuk mengidentifikasi struktur atau

melokalisasi aktivitas kimia yang terjadi. Untuk bahan anorganik, penggunaan terpolarisasi cahaya atau permukaan oksidasi menghasilkan citra warna untuk mencitrakan struktur.

Penggunaan skala warna sebagai pengganti nilai kecerahan memungkinkan kita untuk menampilkan dan melihat perubahan kecil, dan mengidentifikasi kecerahan yang sama nilai-nilai global dalam citra digital. Beberapa sumber citra dapat menggunakan warna untuk mengkodekan berbagai jenis informasi, seperti intensitas dan polarisasi gelombang radio, namun sejauh ini yang paling umum jenis citra warna yang dihasilkan dengan merekam intensitas pada tiga panjang gelombang cahaya yang berbeda. Sistem Televisi Nasional Committee (NTSC) mempunyai skema encoding warna yang digunakan seperti di Amerika Serikat yang dikembangkan agar kompatibel dengan televisi monokrom. Konversi dari RGB ke YIQ dapat dikerjakan dengan menggunakan rumus:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0.956 & 0.621 \\ 1.0 & -0.272 & -0.647 \\ 1.0 & -1.105 & 1.702 \end{pmatrix} \begin{pmatrix} Y \\ I \\ Q \end{pmatrix}$$

Baris pertama $Y = (0.299, 0.587, 0.114)$ (R,G,B) juga dapat digunakan untuk mengubah RGB menjadi Black and White. Sedangkan konversi dari YIQ ke RGB yang merupakan transformasi matriks inverse yang mengubah dari YIQ ke RGB digunakan rumus sebagai berikut:

$$\begin{pmatrix} Y \\ I \\ Q \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.212 & -0.523 & 0.311 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Konversi dari RGB (kecerahan dari komponen merah, hijau, dan biru pada sinyal didefinisikan sebagai panjang gelombang, seperti yang ditangkap oleh kamera dan disimpan di komputer) untuk YIQ / YUV. Komponen I dan Q (atau U dan V) dari sinyal warna yang dipilih untuk kompatibilitas dengan perangkat keras yang digunakan. sinyal I pada dasarnya cyan minus merah, sedangkan Q adalah magenta minus hijau.

1.4 Konversi RGB ke dan dari YUV

Komponen YUV mirip dengan komponen matriks transformasi dari RGB ke YUV dapat ditulis sbb:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1.0 & 0.0 & 1.140 \\ 1.0 & -0.394 & -0.581 \\ 1.0 & 2.028 & 0.0 \end{pmatrix} \begin{pmatrix} Y \\ U \\ V \end{pmatrix}$$

Instrumen yang digunakan untuk pengukuran warna adalah *primary* CIE, yang menentukan panjang gelombang yang dominan dan kemurnian warna apapun. Mencampur dua warna sesuai dengan memilih sebuah titik yang baru dalam diagram sepanjang garis lurus antara dua warna asli. Ini berarti bahwa sebuah segitiga pada CIE diagram dengan sudutnya di lokasi merah (*Red*), hijau (*Green*), dan biru (*Blue*) merupakan fosfor emisi seperti yang digunakan dalam katoda *ray tube* (CRT) yang mendefinisikan semua warna yang dapat ditampilkan dalam tabung warna.

Diagram CIE menyediakan alat untuk mendefinisikan warna. Pendekatan yang dilakukan adalah diwujudkan dalam sistem HSV (hue, saturasi, dan nilai), HSI (hue, saturasi, dan intensitas) dan HLS (*Hue, Ligth*, dan saturasi). Hal ini terkait erat satu sama lain dan dengan konsep warna, naungan, dan nada. Dalam sistem ini, hue adalah warna seperti yang dijelaskan oleh panjang gelombang, misalnya perbedaan antara daerah merah dan kuning. Saturation adalah jumlah warna yang hadir, misalnya perbedaan antara merah dan pink. Sumbu ketiga (disebut *ligth*, intensitas, atau nilai) adalah jumlah cahaya, perbedaan antara merah dan terang gelap, atau antara abu-abu gelap dan abu-abu terang.

Ruang di mana ketiga nilai diplot dapat ditunjukkan sebagai kerucut melingkar atau hexagonal atau kerucut ganda, atau kadang-kadang sebagai suatu silinder. Hal ini berguna untuk membayangkan ruang sebagai kerucut ganda. Sumbu kerucut yang merupakan perkembangan skala abu-abu dari kejauhan hitam menjadi putih, di pusat sumbu adalah saturasi, sedangkan pada dasarnya adalah hue. Di sisi lain, karena komponen HSI tidak sesuai dengan sebagian besar perangkat keras (baik untuk akuisisi atau layar), maka diperlukan perhitungan untuk mengkonversi RGB-encode citra untuk HSI dan kembali. Untuk melakukan konversi dari RGB ke HIS dengan dikerjakan dengan tahapan sebagai berikut:

- i. Pertama ubah ruang warna RGB ke ruang HSI dengan melakukan normalisasi nilai-nilai dari RGB:

$$r = \frac{R}{R+G+B}, g = \frac{G}{R+G+B}, b = \frac{B}{R+G+B}.$$

- ii. Untuk setiap komponen H, S dan I dinormalisasikan dengan menggunakan perhitungan sebagai berikut:

$$H = h \times 180/\pi, S = s \times 1000, \text{ dan } I = l \times 255$$

Ruang HSI berguna untuk mengolah citra karena mereka memisahkan informasi warna dalam cara-cara yang sesuai dengan respon sistem visual manusia. Sumbu yang berjalan dari merah (+a) menjadi hijau (-a) dan sumbu b dari kuning (+b) ke biru (-b). Konversi dari RGB bidang ruang dan koordinat hue-saturation-intensitas dapat dilakukan dengan beberapa cara, tergantung pada bentuk ruang HSI yang digunakan. Pada umumnya berbentuk bola, silinder, atau kerucut ganda. Dalam semua hal, intensitas sumbu sejajar sepanjang diagonal kubus, tapi tidak ada geometri ruang HSI tepat sesuai dengan bentuk kubus RGB. Ini berarti dalam untuk menampilkan warna dalam ruang baik nilai saturasi sedikit terdistorsi dalam proses konversi.

Ledley *et al.* (1990) mengatakan bahwa perbedaan antara kedua konversi RGB-HSI terletak terutama dalam nilai-nilai saturasi. Perangkat keras untuk digitalisasi citra berwarna menerima baik langsung sinyal RGB dari kamera, atau komponen sinyal YC, atau sinyal komposit (misalnya, NTSC) dan menggunakan filter elektronik untuk memisahkan setiap komponen dan ekstrak sinyal merah, hijau, dan biru. Seperti untuk kasus monokrom dibahas di atas, sinyal-sinyal ini kemudian digitalisasi untuk menghasilkan nilai-nilai, biasanya yang 8-bit mulai dari 0 sampai 255 masing-masing untuk R, G, dan B. Ini memerlukan 3 byte penyimpanan per piksel, sehingga citra 640 piksel akan membutuhkan hampir 1 megabyte ruang penyimpanan di komputer.

Warna citra digital biasanya biasanya merupakan RGB 24-bit, yang berarti bahwa 8 bit atau 256 (linear) tingkat kecerahan untuk merah, hijau, dan biru disimpan. Kondisi ini cukup untuk memungkinkan menampilkan video pada layar komputer, atau untuk keperluan percetakan, namun, tergantung pada rentang dinamis data. Lebih lanjut pengurangan citra warna 256 warna dengan menggunakan tabel lookup di mana warna 256 terbaik dicocokkan dengan isi citra umumnya yang digunakan untuk citra digital. Hal ini akan mengurangi ukuran file. 256 byte untuk menentukan nilai **R**, **G**, dan **B**. Hanya satu byte per piksel diperlukan untuk memilih palet warna. Teknik yang paling umum untuk memilih palet yang

optimum adalah Heckbert atau algoritma median, yang digunakan untuk membagi ruang warna didasarkan pada nilai-nilai RGB yang sebenarnya dari piksel pada citra asli. Berbagai jenis pemisahan warna yang tersedia. Perhitungan komponen RGB (atau CMY sebagai nilai pelengkap) yang umum digunakan dalam desktop publishing, tapi tidak sering berguna untuk analisa citra, yang juga dapat memisahkan warna citra menjadi komponen hue, saturasi, dan intensitas.

1.5. Kromatisitas koordinat CIE

Kromatisitas Koordinat adalah mencitrakan informasi warna sebagai: Koordinat kromatisitas merah diberikan oleh x dan koordinat kromatisitas hijau oleh y . Nilai *tristimulus linear* dalam I dan dengan demikian informasi intensitas mutlak telah hilang dalam perhitungan kromatisitas {koordinat x, y }. Semua distribusi warna, I , yang tampak memiliki warna yang sama akan memiliki kromatisitas koordinat yang sama pula.

Jika kita menggunakan warna murni (seperti *laser dye*), maka intensitas dapat dimodelkan seperti $I = d$ (-o) dengan d (*) berfungsi sebagai *impuls*. Pengumpulan kromatisitas {koordinat x, y } yang akan dihasilkan dengan memvariasikan 0 memberikan Kromatisitas CIE segitiga seperti yang ditunjukkan pada Gambar 1.7.

Perubahan kecil hampir di mana saja dalam (u', v') ruang Kromatisitas sama menghasilkan perubahan-perubahan kecil dalam warna yang dirasakan *Optical Illusions* Citra dari sistem visual manusia yang ada. Hal ini bisa membawa kita untuk menyimpulkan bahwa ada pengetahuan yang cukup dari sistem visual manusia untuk memungkinkan pemodelan sistem visual dengan teknik standar analisis sistem. Warna secara utuh bergantung pada sifat pantulan (*reflectance*) suatu objek. Warna yang dilihat merupakan yang dipantulkan, sedangkan yang lainnya diserap. Sehingga sumber sinar perlu diperhitungkan begitu pula sifat alami *sistem* visual manusia ketika menangkap suatu warna. Sebagai contoh, suatu objek yang

memantulkan sinar merah dan hijau akan tampak berwarna hijau apabila benda tersebut disinari oleh sinar hijau (tanpa adanya sinar merah). Demikian juga sebaliknya, objek akan tampak berwarna merah apabila tidak terdapat sinar hijau. Apabila benda tersebut disinari oleh sinar putih, maka objek tersebut berwarna kuning (merupakan gabungan warna hijau + merah).

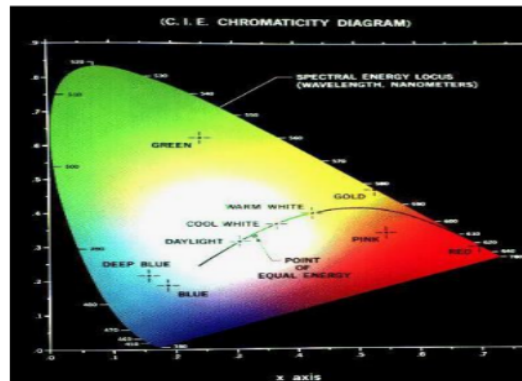
1.6. Perhitungan Kromatisitas Koordinat CIE

Perkiraan komponen warna dapat diberikan ke daerah diagram Kromatisitas CIE . Setiap upaya untuk mencitrakan keseluruhan dari penglihatan warna manusia pada monitor komputer harus disertai dengan berbagai kualifikasi dan pengecualian. Berbagai persepsi warna manusia pada monitor RGB - keseluruhan dari penglihatan manusia normal meliputi seluruh diagram CIE, sedangkan gamut dari RGB monitor dapat ditampilkan sebagai kawasan segitiga dalam diagram CIE. Kualifikasi lain adalah bahwa *hue* dan *saturasi* yang terkait dengan nama warna yang diberikan dapat bervariasi pada rentang yang cukup.

Diagram kromatisitas CIE diperkenalkan oleh Foto Research, Inc. Perlu dicatat bahwa satu nilai pada sekitar tengah dan saturasi warna berkisar dipilih untuk setiap bagian dari diagram. Titik yang dipilih adalah hanya penilaian visual dari warna perwakilan dalam jangkauan. Nilai-nilai RGB yang diperoleh tercantum dalam tabel di sebelah kanan. Setiap orang berbeda dalam memilih titik yang berbeda untuk mewakili nama-nama warna, tapi setidaknya nilai-nilai tersebut mungkin memberikan titik awal untuk variasi pilihan yang sama.

Teori tritimus persepsi warna dapat dilihat bahwa suatu warna dapat diperoleh dari suatu campuran tiga warna utama: merah, hijau dan biru (*Red Green Blue*). Meskipun hampir setiap warna yang tampak dapat ditentukan sesuai dengan tiga komponen diatas, tetapi masih terdapat beberapa warna yang tidak dapat diuraikan sebagai kombinasi dari ketiga warna dasar tersebut. Bagaimanapun juga apabila salah satu dari

ketiga komponen warna dasar tersebut ditambahkan ke warna yang tidak dapat dicocokkan tadi, maka warna yang tidak dapat dicocokkan tersebut dapat dicocokkan dengan campuran dari dua warna dasar lain. Hal ini menunjukkan bahwa warna dapat memiliki nilai bobot negatif dari ketiga komponen warna dasar tersebut. Pada tahun 1931, Commission Internationale de l'Eclairage (CIE) mendefinisikan tiga standar komponen warna utama : X, Y dan Z. Yang dapat ditambahkan untuk membentuk semua kemungkinan warna. Warna utama Y dipilih sedemikian rupa sehingga fungsi kecocokan warnanya secara tepat mencocokkan fungsi *luminous* efisiensi mata manusia berdasarkan penjumlahan ketiga warna seperti pada gambar 1.9 . Diagram *Chromaticity* menunjukkan semua warna yang dapat dilihat oleh manusia. Sumbu x dan y merupakan nilai normalisasi warna utama X dan Y untuk suatu warna, dan $z = 1-x-y$ menyatakan jumlah Z utama yang diperlukan. *Chromaticity* bergantung pada panjang gelombang dan dominan saturation, dan tidak bergantung pada energi luminan. Warna dengan nilai *chromaticity* yang sama tetapi dengan luminan berbeda akan terpetakan pada titik yang sama di daerah tersebut.



Gambar 1.7. Diagram CIE Chromaticity menunjukkan semua *visible colours*. x dan y adalah jumlah normalisasi kemunculan primer X and Y, $z = 1-x-y$ menentukan jumlah primer Z yang dibutuhkan.

Warna spektrum utama berada pada bagian kurva batas daerah, dan suatu sinar putih standar memiliki warna yang didefinisikan berada dekat (tetapi tidak) di titik dengan persamaan energi $x = y = z = 1/3$.

Komplementer warna , yaitu warna yang ditambahkan ke warna putih, berada di titik akhir suatu garis yang melewati titik tersebut. Sebagai ilustrasi pada gambar 1.10, semua warna yang berada di dalam segitiga dapat dibentuk dari campuran warna yang berada pada verteks (garis) segitiga. Dari ilustrasi grafik tersebut, semua warna visible tidak dapat diperoleh dari campuran warna utama R, G dan B (atau dari tiga visible warna lainnya), karena bentuk diagramnya bukan segitiga.

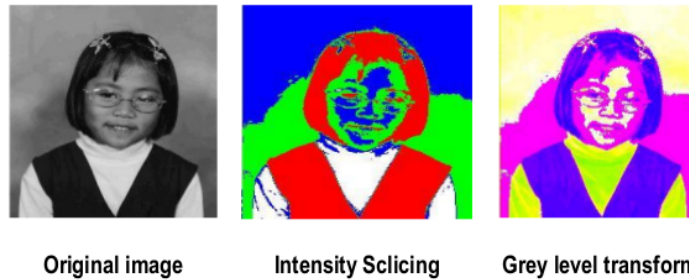
Denoising citra diasumsikan dikenal sebagai *Aditif Gaussian White Noise* (AWGN) Namun, dalam aplikasi nyata kebisingan tidak diketahui dan non-aditif. Banyak algoritma di bidang computer vision secara eksplisit maupun implisit mengasumsikan jenis dan tingkat kebisingan citra. Deviasi standar s dari kebisingan adalah fungsi dari kecerahan citra I yang terukur dengan memperbaiki kamera dan mengambil citra statis. Untuk mean dari setiap piksel, dan Standar deviasi I , NLF tergantung pada kamera, ISO, kecepatan warna, dan diafragma. Tujuan untuk memperkirakan NLF dari satu citra adalah bagaimana memperkirakan kebisingan tanpa memisahkan noise dan sinyal dari citra.

1.7. Pseudocolor processing

Pseudocolor (false color) image processing merupakan perubahan nilai warna menjadi nilai warna grayscale berdasarkan pada criteria tertentu. Istilah "pseudocolor" menekankan pada warna buatan yang sangat berbeda dengan warna yang sebenarnya. Pada prinsipnya pseudocolors merupakan visualisasi dan intepretasi dari citra gray scale secara rinci pada citra. Contoh proses pseudocolor dari citra digital, adalah perubahan citra berwarna ke citra nonochrom yang dapat dilakukan dengan *Intensity slicing* dan citra gray level kedalam transformasi warna dan dengan teknik filtering.

Dalam transformasi warna dari Gray-level menjadi full color terdapat tiga transformasi yang dikerjakan pada citra grey scale. Dimana hasilnya digabung dalam warna merah, hijau dan blue. Untuk mengubah

citra dengan graylevel ke citra warna, maka masing-masing channel RGB diproses dengan cara yang berbeda.



Gambar 1.18 Hasil penerapan intemcity slicing dan grey level transform

1.8. Model Warna

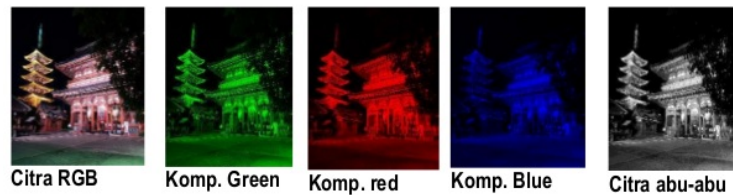
Model warna merupakan cara standar untuk menspesifikasikan suatu warna tertentu, dengan mendefinisikan suatu sistem koordinat 3D, dan suatu ruang bagian yang mengandung semua warna yang dapat dibentuk ke dalam suatu model tertentu. Suatu warna yang dapat dispesifikasikan menggunakan suatu model akan berhubungan ke suatu titik tunggal dalam suatu ruang bagian yang didefinisikannya. Masing-masing warna diarahkan ke salah satu standard *hardware* tertentu (RGB, CMY, YIQ), atau aplikasi pengolahan citra (HSI). Warna adalah spektrum tertentu yang terdapat di dalam suatu cahaya sempurna (berwarna putih). Identitas suatu warna ditentukan oleh panjang gelombang cahaya tersebut. Sebagai contoh warna biru memiliki panjang gelombang 460 nanometer. Panjang gelombang warna yang masih bisa ditangkap mata manusia berkisar antara 380-780 nanometer. Dalam peralatan optis, warna bisa pula berarti interpretasi otak terhadap campuran tiga warna utama cahaya: **R**(red=merah), **G**(green=hijau), **B**(blue=biru) yang digabungkan dalam komposisi tertentu. Misalnya pencampuran 100% merah, 0% hijau, dan 100% biru akan menghasilkan interpretasi warna *magenta*.

Dalam bidang seni , warna bisa berarti pantulan tertentu dari cahaya yang dipengaruhi oleh pigmen yang terdapat di permukaan benda. Misalnya pencampuran pigmen *magenta* dan *cyan* dengan proporsi tepat dan disinari cahaya putih sempurna akan menghasilkan sensasi mirip warna merah. Setiap warna mampu memberikan kesan dan identitas tertentu sesuai kondisi sosial pengamatnya. Misalnya warna putih akan memberi kesan suci dan dingin di daerah negara-negara barat karena berasosiasi dengan salju. Di dalam ilmu warna, hitam dianggap sebagai ketidakhadiran seluruh jenis gelombang warna. Sementara putih dianggap sebagai representasi kehadiran seluruh gelombang warna dengan proporsi seimbang.

Didalam pengolahan citra pada dasarnya warna terdiri dari tiga komponen, yaitu komponen **RGB** (*Red, Green, Blue*). RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, yang jika digabungkan akan membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang-nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak $256 \times 256 \times 256 = 16777216$ jenis warna. Satu jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang 3 dimensi yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z. Misalkan sebuah vektor dituliskan sebagai $r = (x,y,z)$. Untuk warna, komponen-komponen tersebut digantikan oleh komponen R(red), G(green), B(blue). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = RGB(30, 75, 255). Putih = RGB (255,255,255), sedangkan untuk hitam= RGB(0,0,0).

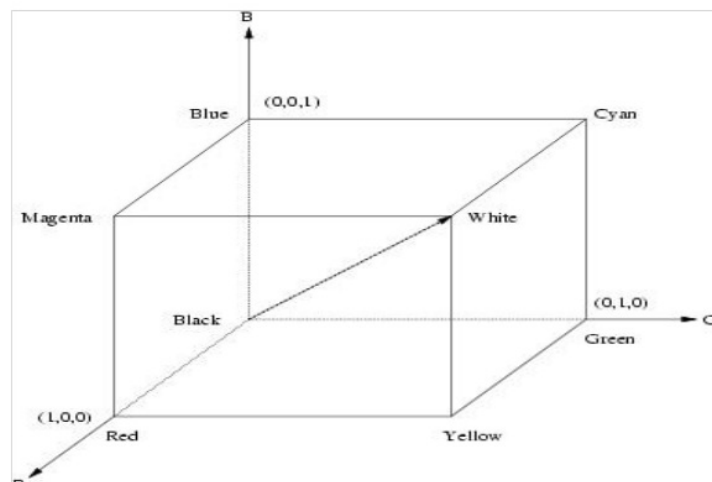
1.8.1 Model RGB

suatu citra dalam model RGB terdiri dari tiga bidang citra yang saling lepas,masing-masing terdiri dari warna utama: merah, hijau dan biru. Suatu warna dispesifikasikan sebagai campuran sejumlah komponen warna utama. Contoh citra dengan individu komponen warna dapat dilihat pada gambar 1.12.



Gambar 1.9. Citra digital masing-masing komponen R,G,B dan gray scale

Sedangkan gambar 1.9. menunjukkan bentuk geometri dari model warna RGB untuk menspesifikasikan warna menggunakan sistem koordinat Cartesian. Spektrum greyscale (tingkat keabuan) yaitu warna yang dibentuk dari gabungan tiga warna utama dengan jumlah yang sama, berada pada garis yang menghubungkan titik hitam dan putih.



Gambar 1.10 Koordinat warna RGB

1.8.2 Model CMY dan gamut

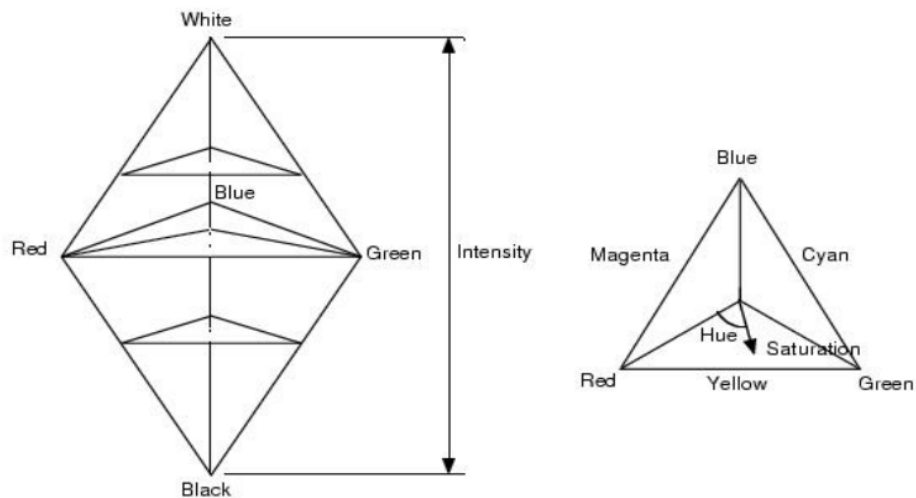
Model CMY (*Cyan, Magenta dan Yellow*) adalah suatu model *subtractive* yang berhubungan dengan penyerapan warna. Suatu permukaan yang dicat warna *cyan* kemudian diiluminasi sinar putih, maka tidak

ada sinar merah yang dipantulkan, dan similar untuk warna magenta dengan hijau, dan kuning dengan biru. Relasi model CMY adalah sebagai berikut:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

1.8.3 Model HSI

Sebagaimana yang sudah dijelaskan, warna juga dapat dispesifikasikan oleh tiga kuantisasi hue, saturation, intensity (disebut model HSI) seperti pada gambar 1.14. Pada gambar sebelah kiri merupakan bentuk solid HSI dan sebelah kanan adalah model segitiga HSI yang merupakan bidang datar dari pemotongan model solid HSI secara horisontal pada tingkat intensitas tertentu. Hue ditentukan dari warna merah, saturation ditentukan berdasarkan jarak dari sumbu. Warna pada permukaan model solid dibentuk dari saturasi penuh, yaitu warna murni, dan spektrum tingkat keabuan,



Gambar 1.11 Model warna HIS

Konversi nilai antar model RGB dan HSI adalah sebagai berikut:

$$I = (R+G+B)/3$$

dimana kuantitas R, G, dan B adalah jumlah komponen warna merah, hijau, biru dan dinormiliasi ke [0,1].

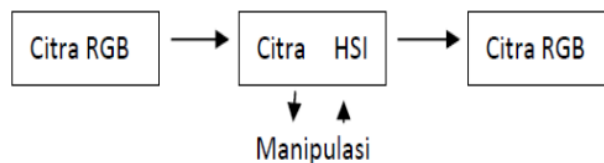
Intensitas adalah nilai rata-rata komponen merah, hijau dan biru. Nilai saturation ditentukan sebagai:

$$S = 1 - \frac{\min(R,G,B)}{I} = 1 - \frac{3}{(R+G+B)}\min(R, G, B)$$

Komponen luminance (Y) berisi semua informasi yang diperlukan untuk televisi hitam dan putih, dan menerjemahkan persepsi kita tentang *brithgness* warna tertentu. Dalam persepsi kita warna hijau lebih enak dipandang dari pada warna merah, sedangkan warna merah lebih enak dipandang dari pada warna biru, hal ini disebabkan karena nilai masing-masing wama adalah 0.587, 0.299 dan 0.114 pada baris pertama dari konversi matrik di atas. Bobot nilai harus digunakan untuk mengkonversi citra menjadi citra abu-abu (grayscale image) dengan mempertahankan kecerahan.

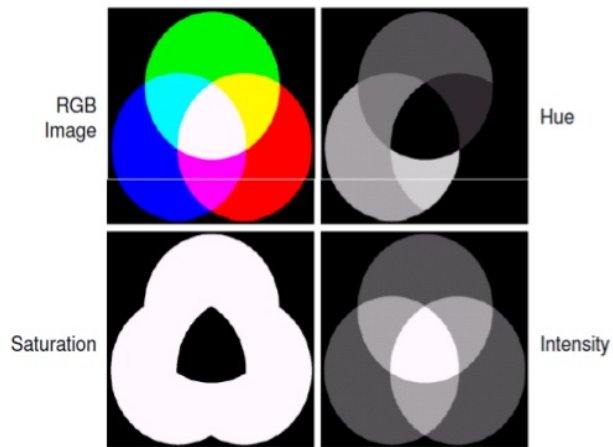
Untuk memanipulasi model HSI maka harus dikerjakan bebarapa tahapan:

- Pertama konvesi RGB kedalam HSI
- Proses manipulasi dikerjakan untuk HSI
- Akhirnya konversi kembali citra dari HSI kedalam RGB



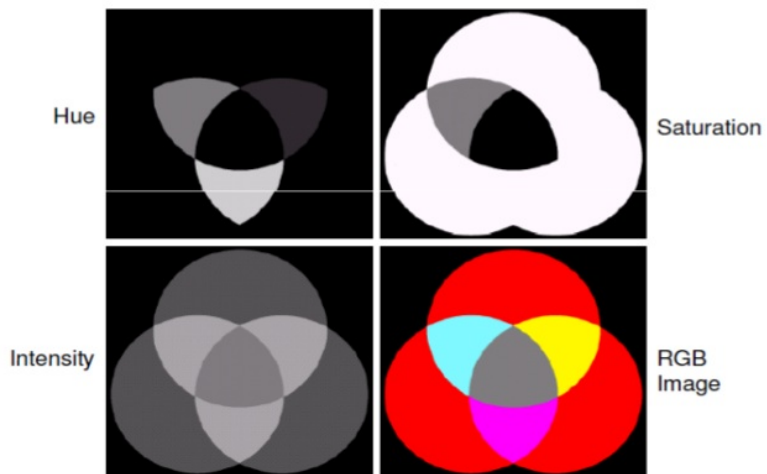
Gambar 12. Konversi RGB ke HSI

RGB -> HSI -> RGB



Gambar 1.13. Konversi citra RGB ke citra HSI

Selanjutnya untuk mengubah kembali dari HIS ke RGB dapat diilustrasikan pada gambar 1.14



Gambar 1.14. Konversi citra HIS ke RGB

1.9. Image noise atau kebisingan Citra

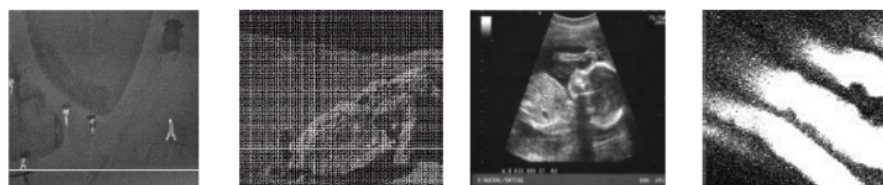
Image noise adalah adanya informasi *brighthness* atau warna yang mengganggu pada sebuah citra digital, biasanya dikarenakan oleh sensor atau circuit dari scanner atau digital kamera. Ada beberapa contoh dari citra noise, antara lain: *Amplifier noise (Gaussian noise)*, *Salt-and-pepper noise*, *Shot noise*, dan *Quantization noise (uniform noise)*. Misalnya $f(x,y)$ merupakan piksel dari citra digital $\eta(x,y)$ adalah image noise dan $g(x,y)$ adalah piksel dari hasil noise. Jika dapat membuat model noise dari suatu citra berdasarkan fungsi tersebut, maka dapat diterangkan bagaimana citra digital dikembalikan ke citra sebelumnya (original image). Noise dapat diartikan sebagai kehadiran signal yang tidak diinginkan. Noise tidak selamanya terjadi secara random atau tidak disengaja, tetapi dapat dibuat secara sengaja sehingga noise tidak selalu jelek atau tidak bermanfaat contohnya pada *Stochastic Resonance*.



Citra tanpa noise Citra dengan sedikit noise Citra dengan banyak noise

Gambar 1.15. Citra noise (*Stochastic Resonance*)

Stochastic resonance merupakan noise yang justru menambah atau meningkatkan kualitas dari suatu citra digital. Sedangkan beberapa contoh dari noise yang mengurangi kualitas dari citra antara lain:



thermal imaging electrical interference ultrasound imaging physical interference

Gambar 1.16. Contoh citra noise yang mengurangi kualitas gambar

Terdapat beberapa jenis atau tipe noise, yaitu:

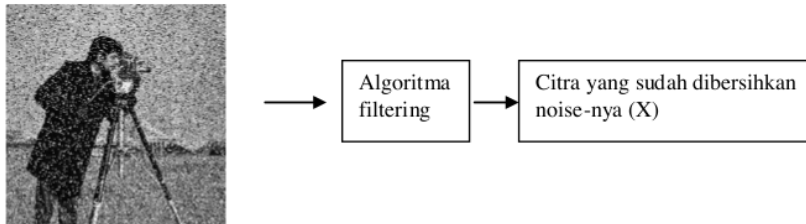
1.9.1 Impulse noise

Pada noise tipe ini setiap piksel dalam suatu citra digital mempunyai probabilitas $p/2$ ($0 < p < 1$) untuk terganggu dengan titik putih (*white dot*) atau titik hitam (*black dot*)

$$Y = \left\{ \begin{array}{l} 255 \text{ dengan probabilitas } p/2 \text{ ----} \rightarrow \text{ piksel - piksel noise} \\ 0 \text{ dengan probabilitas } p/2 \text{ ----} \rightarrow \text{ piksel - piksel noise} \\ X(i, j) \text{ dengan probabilitas } 1 - p \text{ ---} \rightarrow \text{ piksel - piksel tanpa noise} \end{array} \right\}$$

Dimana $1 \leq i \leq H, 1 \leq j \leq W$

Dalam beberapa aplikasi piksel noise tidak hanya hitam dan putih yang dapat membuat penghilangan impulse noise menjadi lebih sulit. Berikut adalah teknik untuk menghilangkan noise (*denoise*)



Gambar 1.17. Teknik menghilangkan noise

1.9. 2.Additive white Gaussian noise

Setiap piksel di dalam suatu citra digital biasanya diganggu dengan adanya noise yang disebut dengan additive white Gaussian Noise yang mempunyai zero mean dan variance σ^2 .

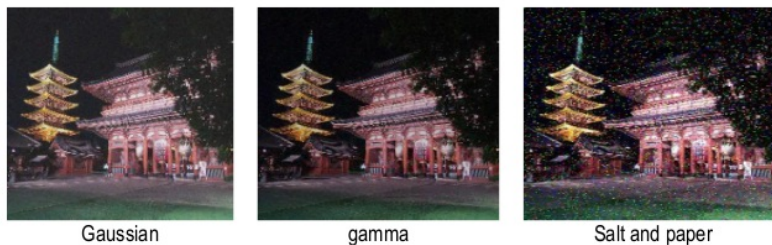
$$Y(i, j) = X(i, j) + N(i, j),$$

$$N(i, j) \sim N(0, \sigma^2), 1 \leq i \leq H, 1 \leq j \leq W$$

Dimana, X: citra tanpa noise sementara Y: citra dengan noise

Citra yang diperoleh melalui sensor modern mungkin terkontaminasi oleh berbagai sumber kebisingan. Dengan kebisingan kita lihat variasi stokastik sebagai kebalikan dari distorsi deterministik seperti bayangan atau kurangnya fokus. Kami akan menganggap untuk bagian ini bahwa kita berhadapan dengan citra yang terbentuk dari cahaya menggunakan elektro-optik modern. Secara khusus kita akan menganggap penggunaan perangkat modern, charge-coupled (CCD) kamera menghasilkan elektron yang biasa disebut sebagai foto elektron. Namun demikian, sebagian besar pengamatan tentang kebisingan dan yang lain dapat digunakan pengolahan citra lainnya.

Sementara teknologi modern telah memungkinkan untuk mengurangi tingkat kebisingan yang terkait dengan perangkat elektro-optik untuk hampir berbagai tingkat diabaikan, salah satu sumber kebisingan tidak dapat dihilangkan dan dengan demikian bentuk kasus pembatas ketika semua sumber kebisingan lainnya adalah "dihilangkan".



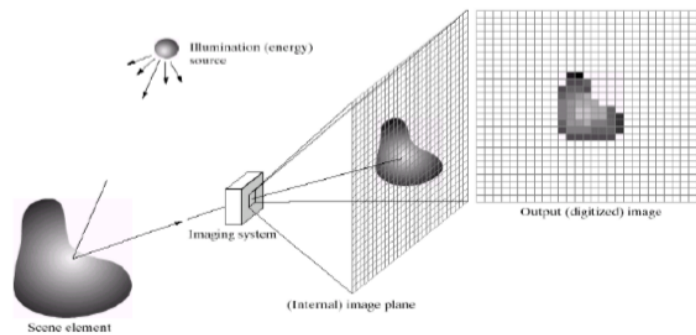
Gambar 1.18. Berbagai tipe kebisingan (noise)

BAB - II

REPRESENTASI CITRA DIGITAL

- Tujuan intruksional umum : Mampu memahami bagaimana dan menjelaskan citra direpresentasikan
- Tujuan instruksional khusus :
- Bagaimana citra digital terbentuk
 - Konsep Sampling dan Kuantisasi
 - Mengerti apa distorsi dan debluring

2.1 Terbentuknya citra digital



Gambar 2.1. Pembentukan suatu citra digital

Pembentukan suatu citra digital dapat dimodelkan dengan persamaan, sebagai berikut:

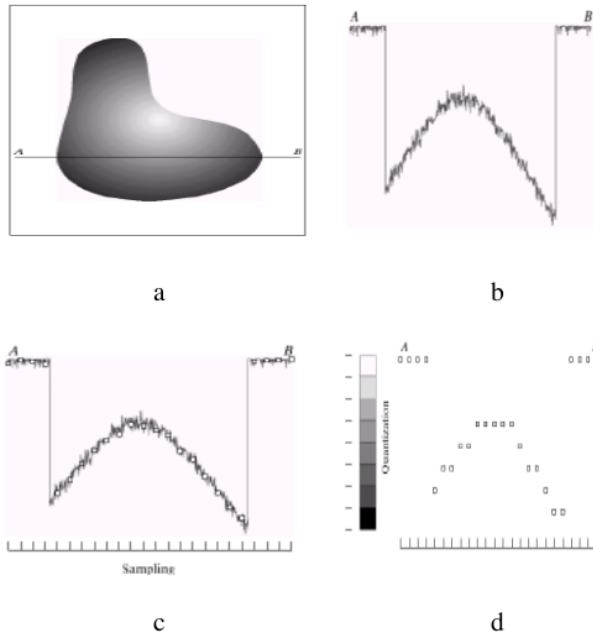
$$f(x,y) = i(x,y)r(x,y) + n(x,y)$$

Dimana : $0 < f(x,y) < \infty$, proposional untuk energi radiasi

$0 < i(x,y) < \infty$, illumination

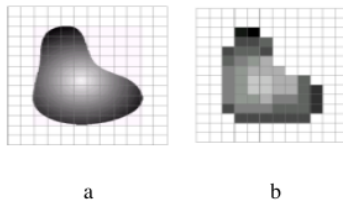
$0 < r(x,y) < 1$, reflectance

$n(x,y)$, noise (derau)



Gambar 2.2. Generating citra digital. a. citra kontinu ,b.scanning dari A ke B pada citra kontinu untuk menggambarkan konsep sampling dan kuantisasi,c. sampling dan kuantisasi, d. garis scanning digital

2.2. Sampling and Quantization citra 2D

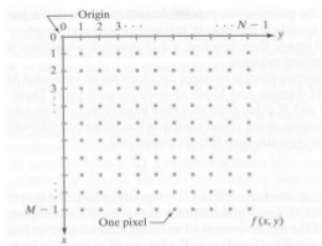


Gambar 2.3. a). Citra kontinu diproyeksikan suatu sensor matrik, b). hasil dari citra sampling dan kuantisasi

Proses konversi dari analog ke digital dapat dikerjakan dengan dua teknik yaitu teknik sampling dan kuantisasi, tahapan proses Sampling adalah sebagai berikut:

1. Digitalisasi koordinat spasial
2. Kemudian nilai-nilai dalam citra kontinyu $f(x,y)$ didekati dengan nilai-nilai diskrit dalam array $N \times M$; biasanya $N = 2^n$ & $M = 2^m$
3. Setiap elemen array kemudian dikonversi menjadi picture element (piksel)

Untuk membuat citra digital, kita perlu mengkonversi terus menerus ke dalam bentuk digital lihat gambar 2.2. dan 2.3. Gambar 2.3 (a) menunjukkan gambar terus menerus, $f(x,y)$ dimana ditunjukkan konversi citra analog ke bentuk digital. Untuk mengubahnya menjadi bentuk digital, kita harus men-sampling fungsi baik dalam koordinat maupun dalam amplitudo. Suatu citra dapat merupakan fungsi continue baik dengan x dan y maupun dengan koordinat dan juga dalam *amplitude*. Digitisasi nilai koordinat disebut sebagai *sampling* sedangkan digitisasi nilai amplitude disebut sebagai kuantisasi. Hasil sampling dan kuantisasi adalah suatu matrik bilangan riil spt ditunjukkan pada gambar 2. 4



2.4.a. Koordinat digunakan untuk mewakili citra digital.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

2.4.b. Citra digital den ukuran $M \times N$

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \dots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \dots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \dots & a_{M-1,N-1} \end{bmatrix}$$

2.4.c. Citra digital

Gambar 2.4. Sampling dan kuantisasi adalah suatu matrik

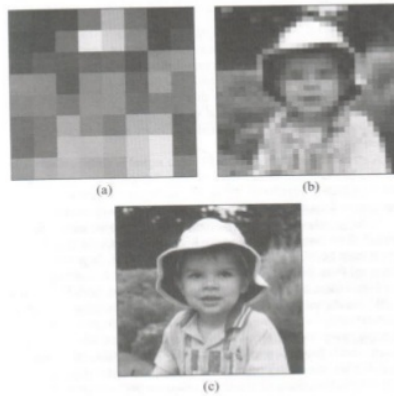
Nilai koordinal awal adalah $(x,y) = (0,0)$. Nilai-nilai koordinat selanjutnya pada baris pertama adalah $(x,y) = (0,1)$, notasi $(0,1)$ digunakan untuk menunjukkan pentingnya sample ke dua sepanjang baris ke dua. Jumlah bit yang diperlukan untuk menyimpan citra yang telah digitalisasi adalah:

$$b = M \times N$$

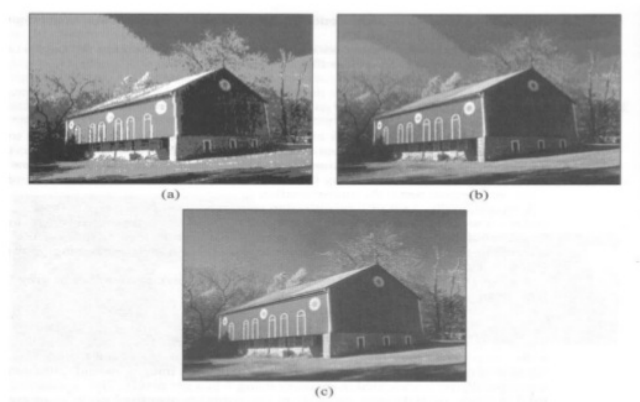
Dimana **M** dan **N** adalah jumlah baris dan kolom, sedangkan jumlah gray level adalah pangkat dua dari bilangan integer:

$$L = 2^k, \text{ dimana } k = 1, 2, \dots, 24$$

Citra seperti ini biasa disebut sebagai citra **k** bit.



Gambar 2.5. Pengaruh resolusi pada citra interpretasi citra a. citra 8x8, b. 32 x 32, c. 256x256.



Gambar 2.6. Pengaruh kuantisasi nada citra interpretasi. (a) 4 levels. (b) 16 levels. (c) 256 levels.

Sementara itu tahapan proses kuantisasi adalah sebagai berikut:

1. digitalisasi amplitudo
2. Jumlah gray level yang diperbolehkan untuk tiap elemen array = $G = 2^n \rightarrow$ berjarak sama pada rentang $[0,L]$

Kuatisasi adalah Konversi dari amplitudo tak terhingga yang tepat untuk suatu bilangan biner;

2.3 Distorsi dan blurring

Efek blur adalah distorsi gambar yang umum terjadi di bidang fotografi. Gambar yang blur (kabur) bisa muncul karena berbagai alasan, seperti lensa kamera yang tidak fokus, intensitas cahaya yang sangat ekstrim, ketidak sempurnaan fisik lensa yang mengakibatkan deviasi optis, dan gerakan relatif objek terhadap lensa kamera. Gerakan relatif objek terhadap lensa, misalnya, diketahui menyebabkan pergeseran posisi rincian objek pada gambar sehingga tampilan tekstur dan tepian dari objek pada gambar menjadi tidak jelas. Efek *motion blur* bisa juga terjadi secara tidak sengaja sebagai akibat proses pengambilan gambar digital yang tidak sempurna (misalnya karena adanya pergerakan sekecil apapun).

Sementara itu efek blur dapat disebabkan adanya gerakan yang disengaja yang ditambahkan pada gambar untuk menimbulkan sensasi gerak cepat dari objek gambar. Fotografer sering menggunakan efek ini untuk menghasilkan gambar yang dramatis sehingga gambar yang dihasilkan akan dianggap memiliki daya tarik lebih.



Gambar 2.4. Beberapa contoh artifak distorsi blur pada gambar: (a) Efek blur yang menimbulkan sensasi gerakan cepat; (b) Efek blur yang tidak diinginkan pada seluruh area gambar; (c) Efek motion blur lokal pada sebagian area gambar

Gambar 2.4 menampilkan beberapa contoh efek blur gerak (*motion blur*). Gambar 2.7(a) adalah contoh efek blur gerak yang diinginkan untuk memberikan sensasi gerakan cepat; gambar ini dibuat dengan menggunakan kecepatan warna yang lambat pada kamera apabila dibandingkan dengan pergerakan objek ketika gambar diambil. Efek blur pada Gambar 2.7(b) adalah distorsi yang tidak diinginkan karena membuat detail gambar menjadi kabur. Distorsi ini mungkin terjadi karena proses pemindahan gambar yang kurang sempurna, atau karena goyangan kamera pada saat pemotretan (*camera shake*). Berbeda dengan Gambar 2.7(c) yang mengandung efek blur lokal pada sebagian daerah gambar, distorsi blur pada Gambar 2.7(b) terjadi secara merata pada semua area.

Citra yang diperoleh melalui sensor modem sering terganggu dengan oleh berbagai sumber kebisingan θ . Dengan kebisingan kita lihat variasi stokastik sebagai lawan distorsi deterministik seperti bayangan atau kurangnya fokus. Secara khusus kita akan menganggap penggunaan perangkat modem, *charge-coupled digital* (CCD) kamera dimana foton menghasilkan elektron yang biasa disebut sebagai foto elektron.

2.4 Representasi spasial citra digital

Supaya citra dapat direpresentasikan secara numerik, maka sebuah citra harus digitalisasi, baik terhadap ruang (koordinat (x,y)) maupun terhadap derajat keabuannya ($f(x,y)$). Proses digitalisasi koordinat (x,y) dikenal sebagai pencuplikan citra (*image sampling*), sedangkan proses digitalisasi derajat keabuan $f(x,y)$ disebut sebagai "Kuantisasi Derajat Keabuan" (*graylevel quantization*). Citra digital dapat dinyatakan dalam matriks dua dimensi $f(x,y)$ dimana (x,y) merupakan koordinat piksel dalam matriks dan f merupakan derajat intensitas piksel tersebut. Citra digital berbentuk matriks dengan ukuran $M \times N$ akan tersusun sebagai berikut:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M) \\ \dots & & & \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1, M-1) \end{bmatrix}$$

Berarti penyimpanan untuk citra digital yg disampling dengan N x M piksel dan dikuantisasi menjadi 2^m level derajat keabuannya membutuhkan memori N x M x m. Contoh, citra yang berukuran 512 x 512 dengan 256 derajat keabuan membutuhkan memori sebesar 512 x 512 x 8 bit = 2048.000 bit. Resolusi gambar ditentukan oleh N dan m. Makin tinggi nilainya maka citra yg dihasilkan makin bagus kualitasnya(mendekati citra kontinu).

ANALISA CITRA DIGITAL

- Tujuan intruksional umum* : *Mampu memahami bagaimana proses analisa citra dan peranannya dalam image processing*
- Tujuan instruksional khusus* :
- *Mengerti konsep dan definisi analisa citra*
 - *Memahami contoh-contoh analisa citra (edge detection)*
 - *Mengerti penggunaan matlab dalam analisa citra*

Analisa citra adalah ekstrasi informasi yang penting dari citra. Contoh sederhana dari analisa citra adalah pembacaan barcode sedangkan yang lebih canggih adalah proses identifikasi seseorang berdasarkan wajah (face detection). Yang termasuk analisa citra antara lain: deteksi pinggir (edge detection), segmentasi (segmentation), pattern recognition, dan content based retrieval.

3.1 Edge detection

Deteksi pinggir atau edge detection merupakan teknik yang sering digunakan pada bidang pengolahan citra dan computer vision, terutama pada bidang *features detection dan feature extraction*. Dalam suatu citra digital, tepi atau edge adalah sebuah kurva yang mengikuti yang berubah dengan cepat berdasarkan intensitasnya. Edge sering dikaitkan dengan batas-batas objek dalam sebuah objek. Deteksi tepi digunakan untuk mengidentifikasi tepi dalam citra digital.

Metode deteksi tepi-tepi yang paling sering digunakan adalah metode Canny. Metode Canny berbeda dari metode deteksi tepi yang lain karena menggunakan dua ambang batas yang berbeda (untuk mendeteksi tepi yang jelas dan tepi yang tidak jelas). Berikut penerapan dari detector metode Canny dengan dibandingkan dengan metode sobel dengan menggunakan Matlab:

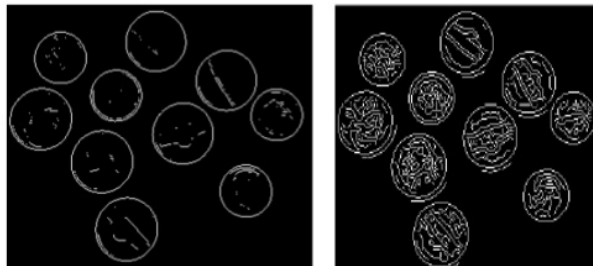
Membaca citra coin dan menampilkannya.
`I = imread('coins.png');`
`imshow(I)`



Gambar 3.1 Citra coins

Berikut adalah perintah di Matlab untuk metode Sobel and Canny

```
BW1 = edge(I, 'sobel');  
BW2 = edge(I, 'canny');  
imshow(BW1)  
figure, imshow(BW2)
```



Sobel

Canny

Gambar 3.2. Hasil edge detection dengan metode sobel dan canny

Metode sobel mencari tepi dengan menggunakan turunannya, dimana dengan turunan tersebut kembali ke tepi pada titik-titik dimana gradientnya maksimum 1.

$$\frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad s_x$$

$$\frac{1}{8} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad s_y$$

$$\begin{bmatrix} 0 & 1 & & & 1 & 0 \\ -1 & 0 & & & 0 & -1 \end{bmatrix}$$

a

$$\begin{bmatrix} -1 & 0 & 1 & & 1 & 1 & 1 \\ -1 & 0 & 1 & & 0 & 0 & 0 \\ -1 & 0 & 1 & & -1 & -1 & -1 \end{bmatrix}$$

b

$$\begin{bmatrix} -1 & 0 & 1 & & & 1 & 2 & 1 \\ -2 & 0 & 2 & & & 0 & 0 & 0 \\ -1 & 0 & 1 & & & -1 & -2 & -1 \end{bmatrix}$$

c

$$\begin{bmatrix} -3 & -1 & 1 & 3 & & & 3 & 3 & 3 & 3 \\ -3 & -1 & 1 & 3 & & & 1 & 1 & 1 & 1 \\ -3 & -1 & 1 & 3 & & & -1 & -1 & -1 & -1 \\ -3 & -1 & 1 & 3 & & & -3 & -3 & -3 & -3 \end{bmatrix}$$

d

Gambar 3.3. (a):operator roberts' (b): operator prewitt 3x3 prewitt , (c): operator sobel , (d) Operator Prewitt 4x4

Metode Roberts digunakan untuk mencari tepi dengan menggunakan perkiraan turunan algoritma Roberts. Metode Robert kembali pada nilai titik yang nilai maksimumnya 1.

3.2 Menelusuri Batas

Dalam MatLab Tool box Image processing mempunyai dua fungsi yaitu fungsi **bwtraceboundary** dan **bwboundaries**. Berikut adalah contoh penggunaan fungsi **bwtraceboundary** to digunakan untuk men-trace batas (border) suatu objek pada citra biner dan kemudian menggunakan fungsi **bwboundaries** untuk men-trace batas dari batas semua objek dalam citra. Berikut perintah (command) dalam MatLab.

1. Baca citra dan display.

```
I = imread('coins.png');  
imshow(I)
```

2. Konversi citra ke dalam citra biner.

`bwtraceboundary` dan `bwboundaries` hanya bisa digunakan pada citra biner.

```
BW = im2bw(I);  
imshow(BW)
```

3. Menentukan koordinat baris dan kolom dari piksel pada batas objek yang akan di-trace. Fungsi

`bwboundary` menggunakan titik ini sebagai titik awal dalam mem-tracing batas.

```
dim = size(BW)  
col = round(dim(2)/2)-90;  
row = min(find(BW(:,:col)))
```

4. Fungsi `bwtraceboundary` men- trace batas dari titik tertentu ,langkah pertama adalah citra biner

harus ditentukan dulu koordinat kolom dan barisnya dari sebagai titik awal dan arahnya.

```
boundary = bwtraceboundary(BW,[row, col],'N');
```

5. Menampilkan citra asli grayscale dan menggunakan koordinat `bwtraceboundary` untuk mem-plot to batas tepi citra.

```
imshow(I)  
hold on;  
plot(boundary(:,2),boundary(:,1),'g','LineWidth',3);
```

6. Untuk men-trace batas semua citra coins. Dengan menggunakan fungsi `bwboundaries`. Secara otomatis, fungsi `bwboundaries` mencari batas semua objek suatu citra termasuk objek yang ada didalam objek lain. Dalam buku ini digunakan citra biner, dalam kasus ini daerah – daerah yang berwarna hitam oleh `bwboundaries` interpretasikan sebagai batas dari objek-objek yang ada. Oleh

Karen itu dalam program MatLab untuk menyakinkan bahwa `bwboundaries` hanya me-traces uang logam (coins) digunakan fungsi ***imfill*** untuk mengisi daerah yang ada didalam setiap coin.

```
BW_filled = imfill(BW,'holes');  
boundaries = bwboundaries(BW_filled);
```

Disini fungsi `bwboundaries` kembali ke sel, dimana setiap koordinat sel terdiri dari baris dan kolom the row/column coordinates suatu citra objek.

7. Plot batas semua coins pada citra grayscale yang merupakan citra awal (original) dengan menggunakan fungsi ***bwboundaries***.

```
for k=1:10  
    b = boundaries{k};  
    plot(b(:,2),b(:,1),'g','LineWidth',3);  
end
```

3.3 Segmentasi

Segmentasi merupakan proses mempartisi citra menjadi beberapa daerah atau objek . Segmentasi citra pada umumnya berdasar pada sifat *discontinuity* atau *similarity* dari intensitas piksel

- Pendekatan *discontinuity*: mempartisi citra bila terdapat perubahan intensitas secara tiba-tiba (*edge based*).
- Pendekatan *similarity*: mempartisi citra menjadi daerah-daerah yang memiliki kesamaan sifat tertentu (*region based*)

Contoh: *thresholding, region growing, region splitting and merging*

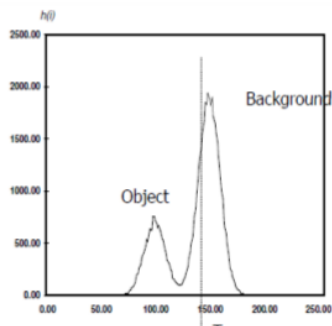
Citra digital merupakan atau berisi piksel-piksel atau citra digital = { piksels }, segmentasi merupakan partisi atau membagi citra kedalam beberapa bagian yang lebih kecil (segmen). Segmentasi dilakukan dengan menggabung piksel-piksel yang mirip atau similar (piksel i and j), $S_{ij} = S_j \geq 0$. Tujuan segmentasi adalah menggabungkan piksel yang mirip dengan nilai dari nilai S_j yang besar, harus pada segmen yang sama, sedangkan piksel yang tidak mirip pada segmen yang berbeda-beda. Segmentasi

dapat juga dianggap sebagai proses pembagian citra ke dalam kelompok-kelompok yang berarti yang saling berkaitan.

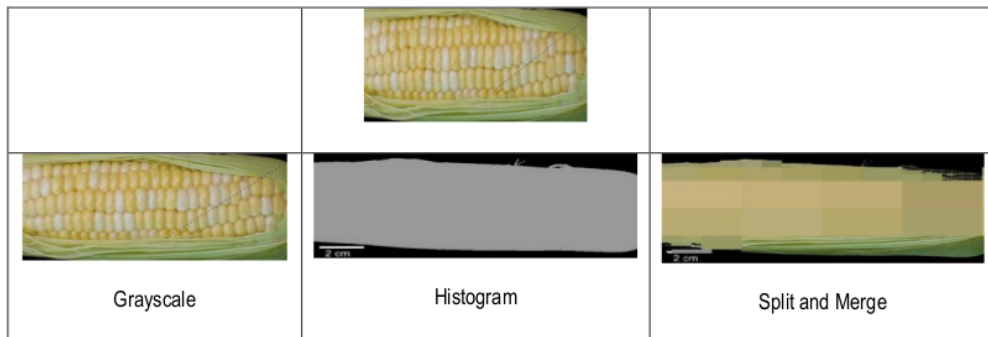
Tujuan dari segmentasi adalah membagi-bagi dalam bagian – bagian atau region yang penting untuk aplikasi tertentu. Segementasi dikerjakan dengan berbasis pada: *greylevel*, *colour*, *texture*, *depth* or *motion*. Segmentasi merupakan proses awal dari process pengolahan citra selanjutnya, contoh aplikasi segmentasi antara lain: Indefikasi objek berdasarkan ukuran dan bentuk dan Identifikasi objek dari moving picture (*object-based video compression (MPEG4)*). Terdapat beberapa macam segmentasi, anantara lain Greylevel histogram-based segmentation Thresholding dan Greylevel histogram-based segmentation Clustering. Didalam tipe segmentasi , terlebih dulu didefinisiakann citra input sebagai citra input *signal-to-noise ratio* dalam hal rata-rata (mean) dari niai greylevel piksel objek dan piksel background dan standard deviasi noise.

$$S / N = \frac{|\mu_b - \mu_o|}{\sigma}$$

Dapat mudah dipahami bahwa segemetasi berbasis throsolding dengan memeriksa histogram pada citra dengan *low noise* objek dan *background*, seperti pada gambar berikut:



Gambar 3.4. Histogram dengan 1 “lembah” diantara 2 puncak.



Gambar 3.5. Contoh segmentasi dengan histogram dan region

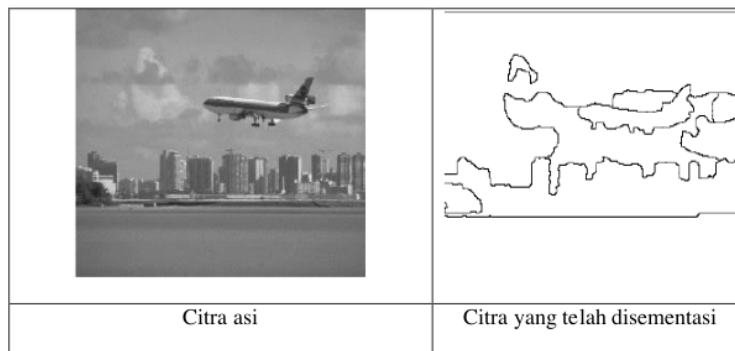
Dalam melakukan pengenalan sebuah objek di antara banyak objek dalam citra, komputer harus melakukan proses segmentasi terlebih dahulu. Segmentasi adalah memisahkan citra menjadi bagian-bagian yang diharapkan merupakan objek-objek tersendiri atau membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan tertentu antara derajat keabuan suatu piksel dengan derajat keabuan piksel-piksel tetangganya.

Berdasarkan cara kerjanya, terdapat dua jenis teknik segmentasi citra, yaitu :

1. Segmentasi berdasarkan intensitas warna (derajat keabuan). Berasumsi bahwa objek-objek yang akan dipisahkan cenderung memiliki intensitas warna yang berbeda-beda dan masing-masing objek memiliki warna yang hampir seragam. Salah satu teknik segmentasi berdasarkan intensitas warna adalah *mean clustering*. Pada *mean clustering* dilakukan pembagian citra dengan membagi histogram citra. Kelemahan segmentasi berdasarkan intensitas warna (derajat keabuan) antara lain adalah harus tahu dengan tepat berapa jumlah objek yang ada pada citra serta citra hasil kurang bagus jika pada citra terdapat beberapa objek dengan warna pada masing-masing objeknya bervariasi atau pada setiap objek memiliki warna yang sama.

2. Segmentasi berdasarkan karakteristik. Yaitu mengelompokkan bagian-bagian citra yang memiliki karakteristik yang sama berupa perubahan warna antara titik yang berdekatan, nilai rata-rata dari bagian citra tersebut. Untuk menghitung atau menentukan karakteristik digunakan perhitungan statistik, misalnya varian, standard deviasi, teori probabilitas, transformasi fourier. Salah satu teknik segmentasi berdasarkan karakteristik adalah *split and merge*. Proses tersebut dilakukan secara rekursif karena pada setiap saat dilakukan proses yang sama tetapi dengan data yang selalu berubah.

Sedangkan berdasarkan metodenya, terdapat dua pendekatan utama dalam segmentasi citra yaitu didasarkan pada tepi (*edge-based*) dan didasarkan pada wilayah (*region-based*). Segmentasi didasarkan pada tepi membagi citra berdasarkan diskontinuitas di antara sub-wilayah (*sub-region*), sedangkan segmentasi yang didasarkan pada wilayah bekerjanya berdasarkan keseragaman yang ada pada sub-wilayah tersebut. Hasil dari segmentasi citra adalah sekumpulan wilayah yang melingkupi citra tersebut, atau sekumpulan kontur yang diekstrak dari citra (pada deteksi tepi). Contoh segmentasi dapat dilihat dalam gambar 3.1. Tiap piksel dalam suatu wilayah mempunyai kesamaan karakteristik atau properti yang dapat dihitung (*image properties*), seperti : warna (*color*), intensitas (*intensity*), dan tekstur (*texture*).



Gambar 3.6. Segmentasi citra pesawat

Segmentasi wilayah merupakan pendekatan lanjutan dari deteksi tepi. Dalam deteksi tepi segmentasi citra dilakukan melalui identifikasi batas-batas objek (*boundaries of object*). Batas merupakan lokasi dimana terjadi perubahan intensitas. Dalam pendekatan didasarkan pada wilayah, maka identifikasi dilakukan melalui wilayah yang terdapat dalam objek tersebut. Salah satu cara untuk mendefinisikan segmentasi citra adalah sebagai berikut. Sekumpulan wilayah $\{ R_1, R_2, R_3, \dots, R_n \}$ merupakan suatu segmentasi citra R ke dalam n wilayah jika :

i. $\bigcup_{i=1}^n R_i = R$

ii. $R_i \cap R_k = \phi, i \neq k$

iii. R_i terhubung $i= 1, 2, \dots, n$

iv. Terdapat suatu predikat P yang merupakan ukuran homogenitas wilayah:

a. $P(R_i) = \text{TRUE}, i=1, 2, \dots, n$

b. $P(R_i \cup R_k) = \text{FALSE}, i \neq k$ dan R_i adjacent R_k

3.4 Content Based Image Retrieval

Content-based image retrieval system (CBIRS) adalah sistem dan teknik untuk melakukan pencarian citra (*image retrieval*) berdasarkan isi visualnya (*content-based*) antara lain seperti warna, tekstur, bentuk, tepian, dan karakteristik citra lainnya. Sedangkan MPEG-7 *Color Descriptor* digunakan untuk mengekstrak fitur citra secara otomatis berdasarkan deskripsi content yang terdapat dalam citra.

Ide aplikasi pencarian citra yang dibangun terinspirasi oleh kelebihan *Lucene* yang mampu menerima segala jenis data untuk diindekskan terlebih dahulu. Untuk itu dalam aplikasi pencarian citra, harus dibuat dahulu suatu cara untuk mengubah citra dengan merepresentasikan citra ke dalam bilangan vektor. Kemudian membandingkan vektor citra dengan data yang telah diindeks, lalu dihitung perbandingannya. Proses selanjutnya adalah mencari relevansi dari query citra dengan indeks yang ada. Hasil yang didapat

berupa jarak vektor citra yang menunjukkan kesamaan dengan data yang sebelumnya diindeks terlebih dahulu. Semakin kecil jarak yang didapat dari proses perbandingan maka tingkat kesamaan semakin tinggi dengan query yang diberikan. Aplikasi pencarian citra ini dapat dimanfaatkan dalam pengidentifikasian arsip digital image, pencocokan warna, pencarian duplikasi image, bidang disain grafis dan lain-lain. Pencarian citra merupakan permasalahan yang menarik untuk dicari alternatif penyelesaiannya. Berikut ini adalah penelitian yang sebelumnya dilakukan tentang pencarian dan pencocokan citra.

- Pencarian citra dengan teknik multiresolusi dengan memanfaatkan ciri tepi suatu citra, teknik ini dikenal dengan QVE (*Query by Visual Example*).
- Pencarian citra dengan memanfaatkan warna, tekstur dan bentuk citra yang dikenal dengan metode QBIC (*Query By Image Content*). Metode ini telah dimanfaatkan oleh IBM untuk membangun perangkat lunak pencarian citra yang bersifat komersial. Dalam perangkat lunak tersebut user dapat membuat *query* berdasarkan pada berbagai variasi perbedaan atribut visual seperti komposisi warna dan ciri bentuk.
- Pencarian Citra yang didasarkan pada tekstur, dimana citra *query* dibangun berdasarkan berdasarkan bagianbagian tertentu dari suatu citra. Citra yang diambil dari basis data kemudian diurut berdasarkan tingkat kemiripan dengan citra *query*. Dengan teknik ini digunakan (filter) yang diperoleh dari turunan *Gaussian*. Filter ini berguna dalam menghitung tingkat kemiripan antara citra *query* dengan citra dalam database. Metode ini masih tetap efektif walaupun telah terjadi perubahan skala yang cukup besar terhadap ukuran citra *query*.
- Pencarian citra dengan menggunakan metode dekomposisi *wavelet*. Ciri-ciri suatu citra yang disebut dengan *signature* dipilih berdasar koefisienkoefisien *wavelet* yang memiliki magnitude terbesar hasil

proses transformasi *wavelet*. Sedangkan tingkat kemiripan citra *query* dengan citra database dihitung dengan sebuah metrika citra multiresolusi. Metrika ini memberikan sebuah nilai yang menyatakan tingkat kemiripan antara citra *query* dengan citra database. Citra database yang memberikan nilai paling kecil berarti citra tersebut paling mirip dengan citra *query*.

- Pengenalan citra wajah menggunakan Metode Alihragam Gelombang Singkat dan Metrika. Alihragam gelombang singkat dilakukan dengan mengkonvolusi sinyal dengan data filter atau dengan proses perata-rataan dan pengurangan secara berulang, yang sering disebut metode *filter bank*. Gelombang singkat *Coflet 6*, *Daubechies 8* dan *Symlet 8* merupakan jenis alihragam gelombang singkat yang baik
- digunakan untuk pengenalan citra wajah. Waktu pengenalan yang diperlukan sangat pendek dan waktu pengenalan bersifat linear terhadap ukuran basis data, sehingga metode ini dapat digunakan untuk pengenalan citra wajah dengan ukuran basis data yang besar. Hasil lain yang menarik dari penelitian ini adalah tingkat kesuksesan pencarian citra *query* yang bersumber dari sketsa pensil warna sangatlah baik.
- Pencarian citra berbasis DCT ,citra database ataupun citra *query* bersumber dari scanner berupa sidik jari Hasil dari penelitian ini menemukan bahwa pencarian citra sidik jari menggunakan alihragam DCT sangat baik hasilnya jika di *query* menggunakan citra *blur* dan citra *query* asli. Tingkat kesuksesan pencarian citra sangat dipengaruhi oleh batasan *threshold score* yang digunakan. Semakin besar *threshold score* yang digunakan maka tingkat kesuksesan yang dihasilkan akan semakin tinggi.

3.5 Piksel dan koefisien DCT

Piksel neighborhood merupakan istilah dalam *image processing* yang artinya *piksel* tetangga. Maksud dari *piksel* tetanggaini adalah *piksel* yang terdapat di sekeliling *piksel* yang akan diproses. Penggunaan *piksel-piksel* lain yang ada di sekeliling suatu *piksel* yang akan diproses ini bertujuan untuk membuat nilai dari *piksel* yang berada di tengah-tengah *piksel* lainnya tersebut menjadi berubah sesuai dengan keinginan *user*. Matriks *piksel neighborhood* ada yang memakai model *rectangle* juga ada yang memakai model *star*. Tetapi, yang paling sering digunakan adalah model *rectangle*.

3.6 Median filter

Sesuai dengan namanya, *median filter* merupakan suatu metode yang menitik beratkan pada nilai *median* atau nilai tengah dari jumlah total nilai keseluruhan *piksel* yang ada di sekelilingnya. Dimisalkan terdapat data A=1, B=5, C=2, D=9, dan E=7, maka *median filter* akan mencari nilai tengah dari semua data yang telah diurutkan terlebih dahulu dari yang paling kecil hingga pada data yang paling besar dan kemudian diambil nilai tengahnya (1, 2, 5, 7, 9). *Median* dari deret tersebut adalah 5. Pemrosesan *median filter* ini dilakukan dengan cara mencari nilai tengah dari nilai *piksel* tetangga yang mempengaruhi *piksel* tengah. Teknik ini bekerja dengan cara mengisi nilai dari setiap *piksel* dengan nilai *median* tetangganya. Proses pemilihan *median* ini diawali dengan terlebih dahulu mengurutkan nilai-nilai *piksel* tetangga, baru kemudian dipilih nilai tengahnya.

Metode *median filter* ini pernah digunakan sebagai metode untuk mengurangi dampak *noise* pada *file* suara. Pengujian dilakukan dengan sampel suara 8 bit dengan 22,05 kHz dan memiliki *noise* 5%. Dari hasil penelitian, penggunaan *median filter* terhadap *noise* yang terdapat pada *file* suara tersebut memberikan hasil yang baik dan dapat mengurangi *noise* yang terdapat pada *file* suara dengan tidak terlalu mengurangi

kualitas suara aslinya. Kasus mengurangi *noise* terhadap *file* citra sebelumnya pernah diteliti juga dengan meto

de *average filter*. Selain menggunakan metode *average filter*, kasus ini juga pernah diteliti dengan metode *Diffusion Filter*. Dari penelitian tersebut dapat dilihat hasil yang diuji melalui citra awal dan citra akhir setelah proses. Dari hasil penelitian, didapat bahwa *average filter* dan *diffusion filter* memberikan hasil yang cukup baik dalam mereduksi *noise* tetapi dengan hasil citra yang cukup *blur* atau kabur, sehingga hasil yang didapat tampak sedikit berbeda dengan citra aslinya. Dalam hal ini mengurangi *noise* dengan teknik *salt pepper* terhadap *file* citra sebelumnya juga pernah diuji. Pengujian dilakukan dengan tingkat *noise* 10% sampai 90%. Dari pengujian didapat semakin tinggi tingkat *noise* yang terdapat pada sebuah citra, semakin rendah pula tingkat keberhasilannya dan kualitas citra yang didapat. Dalam penelitian yang lain, *median filter* pernah diuji tingkat keberhasilan deteksi dan tingkat keberhasilan dalam mereduksi *noise*. Dari pengujian didapat tingkat rata-rata keberhasilan median filter adalah 97,7%.

3.6 Mask sutau citra digital

- Jika pada point processing kita hanya melakukan operasi terhadap masing-masing piksel, maka pada mask processing kita melakukan operasi terhadap suatu jendela ketetangaan pada citra.
- Kemudian kita menerapkan (mengkonvolusikan) suatu *mask* terhadap jendela tersebut. *Mask* sering juga disebut *filter*. Contoh:

W_1	W_2	W_3
W_4	W_5	W_6
W_7	W_8	W_9

Gambar 4.1. Mask citra 3x3

Nilai piksel pada posisi x dipengaruhi oleh nilai ke 8 tetangganya. Perbedaan dengan point processing: pada point processing, nilai suatu piksel tidak dipengaruhi oleh nilai tetangga-tetangganya.

3.7 Morfologi

Matematika morfologi merepresentasikan citra objek dua dimensi sebagai suatu himpunan matematika dalam ruang *Euclidean* dimana dapat berupa ruang kontinyu R atau ruang diskrit Z. Dulu sebuah citra dipandang sebagai suatu fungsi intensitas terhadap posisi (x,y) , sedangkan dengan pendekatan morfologi, suatu citra dipandang sebagai himpunan. Sebuah objek citra A dapat direpresentasikan dalam bentuk himpunan dari posisi-posisi (x,y) yang bernilai 1 atau 0 dimana nilai-nilai tersebut menunjukkan tingkat gray scale setiap posisi. Nilai 1 untuk *gray level* warna putih dan nilai 0 untuk gray level warna hitam. Prinsip dasar dari matematika morfologi adalah penggunaan structuring element yaitu bentuk dasar dari suatu objek yang digunakan untuk menganalisis struktur geometri dari objek lain yang lebih besar dan kompleks. Tujuannya adalah untuk memperoleh informasi mengenai bentuk dari suatu citra dengan mengatur bentuk dan ukuran suatu *structuring element*.

3.8. Operasi-Operasi Morfologi

Dalam morfologi ada beberapa operasi yang dapat dilakukan, yaitu :

3.8.1 Translasi

Translasi artinya sebuah citra dideser pada arah (x,y) dimana (x,y) adalah koordinat matriks.

Operasi Operasi translasi dinyatakan sebagai.

$$A_w = \{(a, b) + (x, y) : (a, b) \in A\}$$

3.8.2. Dilasi

Operasi Dilasi dilakukan untuk memperbesar ukuran segmen objek dengan menambah lapisan di sekeliling objek sehingga citra hasil dilasi cenderung menebal [9]. Operasi dilasi akan melakukan proses pengisian pada citra asal yang memiliki ukuran lebih kecil dibandingkan structuring element (strel). Dilasi A oleh B dinotasikan dengan $A \oplus B$ dan didefinisikan sebagai:

$$A \oplus B = \bigcap_{x \in B} A_x$$

3.8.3. Erosi

Operasi erosi adalah kebalikan dari operasi dilasi. Pada operasi ini, ukuran objek diperkecil dengan mengikis sekeliling objek sehingga citra hasil cenderung diperkecil menipis. Operasi erosi akan melakukan pengurangan pada citra asal yang lebih kecil dibanding elemen penstruktur (strel). Erosi A oleh B dinotasikan $A \ominus B$ didefinisikan sebagai :

$$A \ominus B = \{w : B_w \subseteq A\}$$

$$A \ominus B = \bigcap_{b \in B} A_b$$

3.8.4. Opening

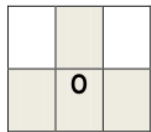
Proses opening pada sebuah citra A oleh strel B dinotasikan dengan $A \circ B$ dan didefinisikan sebagai proses erosi yang dilanjutkan dengan proses dilasi dimana kedua proses tersebut dilakukan secara berulang untuk semua titik (x,y).

$$(A)_{opening} = A \circ B = (A \ominus B) \oplus B$$

3.9. Structuring Element

Structuring element dapat diibaratkan dengan mask pada pemrosesan citra biasa (bukan secara morfologi). Structuring element juga memiliki titik poros (disebut juga titik origin/ titik asal/titik acuan). Di bawah ini adalah contoh structuring element dengan titik poros di (0,0) ditunjukkan dengan huruf "O"

Gambar 4.2.



$$= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$



Gambar 4.2. Structuring element, titik O adalah titik pusat , sedang b adalah representasi biner

Structuring element disajikan dalam bentuk matriks biner seperti pada Gambar 4.2.b, di mana angka "1" dan angka "0" menunjukkan nilai gray level. Dalam morfologi, yang menjadi kunci penting adalah pemilihan structuring element. Structuring element memiliki dua komponen yang penting yaitu bentuk dan ukuran dimana keduanya mempengaruhi hasil pengujian. Pemilihan bentuk structuring element juga mempengaruhi citra hasil operasi morfologi.

3.10. Filtering

3.10.1. Low Pass Filter

Low pass filter, atau dikenal sebagai "pemulusan", digunakan untuk menghilangkan noise frekuensi tinggi spasial dari sebuah citra digital. Kebisingan sering diperkenalkan selama proses konversi analog-ke-digital sebagai efek samping dari konversi fisik pola energi cahaya menjadi pola listrik. Bentuk yang *low-pass filter* biasanya adalah sebagai semacam bergerak operator jendela. Operator biasanya mempengaruhi satu piksel citra pada suatu waktu, mengubah nilai dengan beberapa fungsi suatu daerah "lokal" dari piksel ("tertutup" oleh jendela). Operator "bergerak" di atas citra untuk mempengaruhi semua piksel dalam citra. Beberapa jenis umum adalah dari filtering adalah.

- *Neighborhood-filter* ini rata-rata mengganti nilai setiap piksel, $a[i, j]$ mengatakan, oleh-tertimbang rata-rata piksel dalam beberapa lingkungan di sekitarnya, yaitu jumlah tertimbang dari $[i + p, j + q]$, dengan $k \leq p \leq -k$ untuk k untuk beberapa k positif; bobot adalah non-negatif dengan berat tertinggi pada $p = q = 0$ panjang. Jika semua bobot yang sama maka ini adalah berarti filter.

- *Filter Median* ini menggantikan setiap nilai piksel dengan rata-rata negara tetangga, yaitu nilai tersebut bahwa 50% dari nilai di lingkungan berada di atas, dan 50% di bawah. Ini bisa sulit dan mahal untuk diimplementasikan karena kebutuhan untuk menyortir dari nilai-nilai. Namun, metode ini secara umum sangat baik pada mempertahankan tepi. Dalam Model filter median setiap nilai piksel digantikan oleh tetangga yang paling umum. Ini adalah sangat berguna filter untuk prosedur klasifikasi di mana setiap piksel sesuai dengan obyek yang harus ditempatkan dalam kelas, dalam penginderaan jauh, misalnya, setiap kelas bisa menjadi beberapa jenis medan, jenis tanaman, air, dan lain-lain.

Filter atas semua invariant ruang dalam operasi yang sama diterapkan untuk masing-masing lokasi piksel. Sebuah invariant non-space filtering, dengan menggunakan filter di atas, dapat diperoleh dengan mengubah jenis filter atau bobot yang digunakan untuk piksel untuk bagian yang berbeda dari citra. filter non-linear juga ada yang tidak invarian ruang; upaya ini untuk mencari tepi pada citra bising sebelum menerapkan smoothing, tugas yang sulit yang terbaik, untuk mengurangi kabur karena perataan tepi.

3.11. Analisa bentuk

Analisa bentuk adalah identifikasi suatu objek dengan berdasarkan bentuk objek yang bersangkutan. Proses ini biasanya dikerjakan untuk setelah proses procedure segmentation. Analisa bentuk dapat didefinisikan sebagai analisa bentuk berdasarkan diskripsi bentuk. Terdapat beberapa deskripsi bentuk diantaranya: Including boxes , Fourier descriptors, Guzman polygons, Freeman chain coding, Moment descriptors

3.12.1 Including boxes

Dikerjakan dengan merancang kotak yang mengikut sertakan sumbu bagian objek yang penting dengan sumbu citra digital. Contoh dari deskripsi ini adalah deteksi struktur *filarial*



.Gambar 4.3. Citra Binary setelah segmentasi

Including boxes dapat dilakukan dengan beberapa langkah:

1. objects labeling

- a. Morphological tool (labeling)

```
%load image  
Img=imread('objek.png');  
Img=double(:,:,1);  
%create labeling  
Result = bwlabel(img,8);
```



Gambar 4.4. Morphological tool (labeling)

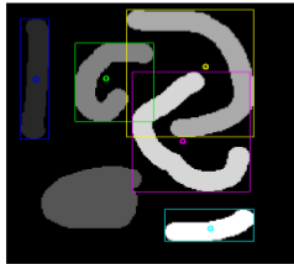
2. Karakterisasi setiap objek, kemudian dilanjutkan dengan karakterisasi menggunakan including boxes.



Gambar 4.5. Karakterisasi citra

3. Perhitungan parameter diskrimnasi

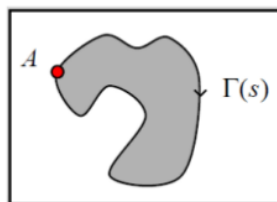
- Ratio antara daerah objek / including square area
- One keeps objects whose corresponding parameter has a value lower than 0.5



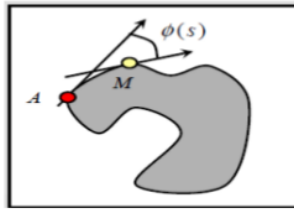
Gambar 4.6. Komputasi diskriminan parameter

3.11.2 Fourier descriptors

Fourier descriptor adalah alat yang digunakan untuk mendeskripsikan bentuk eksternal dari suatu objek yaitu konturnya. Terdapat dua macam *Fourier descriptors*, yaitu: *tangential descriptors* dan *complex descriptors*. Prinsip dari *Tangential descriptor* adalah kontur suatu bentuk digunakan untuk mendiskripsikan sebagai kurva kontinu yang didefinisikan oleh parameter kurva linier s dari A .

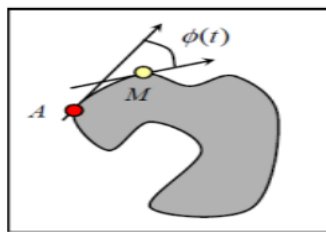


Karakteristik kurva dari sudut $\phi(s)$ yang didefinisikan oleh vektor tangential pada setiap titik point
Dan vector tangential vector pada titik A .



Definisi dari variable t mempunyai nilai-nilai antar 0 dan 2π dapat ditulis dengan persamaan berikut:

$$t = \frac{2\pi \cdot s}{L} \text{ dimana } L \text{ adalah panjang dari kurva.}$$



$$\phi(t) = \phi \left[\frac{2\pi \cdot s}{L} \right] - \left[\frac{2\pi \cdot s}{L} \right]$$

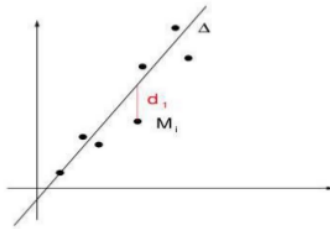
3.11.3. Pengenalan bentuk (*Shape recognition*)

3.13.1 Line approximation

Shape recognition dapat didefinisikan sebagai (piksel) didapat setelah pengolahan citra dikerjakan salah satu tujuannya adalah untuk menemukan bentuk awal yang dapat mendeskripsikan titik terbaik dari point yang diekstrak. Ada dua metode perkiraan garis untuk shape recognition yaitu metode Least square yang diturunkan dari jarak minimum *exact approach* dan *robust estimation approaches*. Pendekatan pasti atau *exact approach* menganggap setiap titik di-ekstrak dianggap mempunyai pengaruh yang sama pada diskripsi garis. Contoh dari *Exact approach* adalah:

1. Pendekatan regresi linier, langkah pertama adalah mencari nilai minimum dari jarak :

$$\Delta: Y = a_0 + a_1 x$$



$$d_i^2 = \sum_{i=1}^N [y_i - (a_0 + a_1 x_i)]^2$$

Selanjutnya metode ini dilanjutkan dengan perhitungan titik-titik yang didapat dari (x_i, y_i) dengan menggunakan persamaan berikut:

$$A = (X^t X)^{-1} X^t Y$$

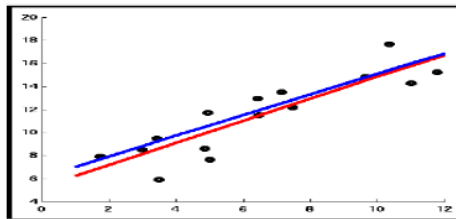
Dimana $X = \begin{bmatrix} 1 & x_1 \\ \dots & \dots \\ 1 & x_N \end{bmatrix}$ $Y = [y_1, y_2, y_3, \dots, y_N]^t$ $A = [a_0, a_1]^t$

3.12. Pendekatan *Robust estimation*

Metode ini adalah mencari garis terbaik yang mewakili titik – titik yang dianggap terganggu oleh noise. Metode ini dapat dibagi menjadi dua cara *median estimation* dan *mixture of lines estimation*. Dalam metode *median estimation* beberapa langkah perlakuan antara lain:

- i. Pilih dua titik M_i dan M_j dari himpunan
- ii. titik-titik yang berhubungan dengan garis yang berkaitan dengan garis yang didefinisikan oleh koefisien a_{ij} dan b_{ij}
- iii. dari gabungan titik titik yang telah diaplikasikan pada titik dari koefisien :
 $a = \{a_{ij}\}$ et $b = \{b_{ij}\}$

- iv. Setelah itu secara terpisah diurutkan dan nilai satu disimpan sebagai nilai median dan yang lain sebagai parameter-parameter untuk mempergunakan garis linier.
- v. Metode ini dapat digunakan untuk memperkirakan sebanyak 50 % titik yang tidak tepat.



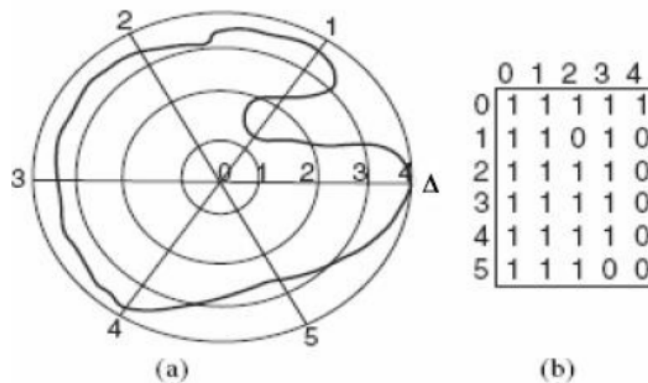
— Regresi linier — Estimasi garis

Shape (bentuk) adalah salah satu atribut yang penting untuk mengenali sebuah objek. Pengenalan bentuk menjadi faktor yang penting dalam pengenalan suatu objek. Sebuah representasi yang baik akan dapat menggambarkan karakteristik intrinsik dari sebuah shape secara eksplisit. Sebuah representasi shape yang bagus harus dapat menggambarkan sebuah objek secara akurat sehingga akan memudahkan menemukan bentuk asli sebuah objek setelah objek tersebut telah direkonstruksi. Pada dasarnya bentuk sebuah objek dapat direpresentasikan dalam bentuk karakteristik internal dan eksternal. Pendekatan shape secara struktural merupakan salah satu cara representasi shape yang umum digunakan pada metode representasi shape pada morfologi.

Shape didefinisikan sebagai sekumpulan titik yang saling terhubung [6]. Jadi sebuah objek dikatakan mempunyai bentuk jika semua titik pada objek tersebut terhubung. Secara taksonomi, ada beberapa teknik dalam representasi shape yaitu dalam bentuk contours, region, dan transforms. Contour-based sama dengan boundary-based yaitu merepresentasikan shape berdasarkan boundary-nya. Regionbased berdasarkan area dari suatu objek, sedangkan teknik transform merepresentasikan shape dalam bentuk koefisien transform, biasa menggunakan transformasi Fourier maupun wavelet.

3.12. Representasi Matrik dari bentuk

Matrik dapat digunakan untuk merepresentasikan sebuah shape. Shape matrix adalah kuantisasi polar dari sebuah bentuk, yang dianalogikan sama dengan sistem koordinat. Titik tengah koordinat (0,0) merupakan titik tengah dari objek shape, dan sumbu axis x merupakan sumbu yang ditarik dari titik tengah menuju titik terjauh dari shape. Jika shape diubah ke dalam bentuk koordinat polar (r, θ) maka shape tidak akan dipengaruhi oleh posisi dan sudut rotasinya. Shape matrix juga tidak dipengaruhi oleh skala dari objek shape tersebut. Shape matrix invariant terhadap translasi, rotasi, dan scaling dari shape tersebut.



Gambar 3.5. (a) Contoh Shape , (b) Matrik Shape

Pada konsep *shape matrix*, sebuah shape akan diubah menjadi bentuk matriks dengan melakukan kuantisasi polar pada shape tersebut (Gambar 5b). Asumsi titik O adalah titik tengah dari shape dan garis OA dengan panjang L adalah radius terjauh shape dihitung dari titik tengah O. Untuk mendapatkan shape matrix $m \times n$, garis OA dibagi menjadi $(n - 1)$ bagian dengan jarak yang sama. Kemudian dibuat lingkaran - dengan titik pusat O - dengan radius lingkaran masing-masing adalah

$$L/(n-1), 2L/(n-1), \dots, (n-1)L/(n-1).$$

Akan didapatkan titik potong antara masing-masing lingkaran dengan garis OA pada $i_1, i_2, i_3, \dots, i_{n-1}$. misalkan didapatkan titik potong (1), (2), (3), dan (4). Kemudian dari setiap titik potong tersebut – dengan arah berlawanan jarum jam – setiap lingkaran dibagi menjadi m busur yang sama dengan sudut $d\theta = 360 / m$ derajat.

Sedangkan untuk menentukan nilai-nilai elemen pada shape matrix dijelaskan pada algoritma di bawah ini:

Buat matrik M dengan ordo $m \times n$

For $i = 0$ to $(n-1)$

For $j = 0$ to $(m-1)$

If titik dengan koordnar polar

$(i.L/(n-1), j(360/m))$

Berada di dalam shape

Then $M(i,j) := 1$

Otherwise $M(i,j) := 0$

BAB IV

JENIS- JENIS CITRA DIGITAL

Tujuan intruksional umum : Mengetahui mengetahui jenis atau tipe citra digital terutama dan aplikasinya di program MatLab.

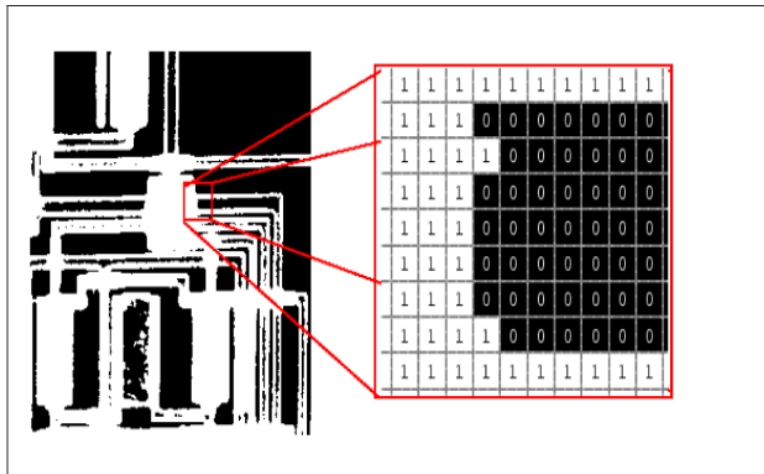
Tujuan instruksional khusus :

- Mengerti dan memahami bagaimana konversi diantara jenis-jenis citra dala MatLab.
- Dapat menjelaskan perintah-perintah di Matab untuk konversi diantara jenis citra

4.1 Jenis atau tipe citra digital

Dalam MatLab dikenal 4 jenis citra digital yaitu: citra Binary , citra Indexed, citra *Grayscale*, dan citra *Truecolor*.

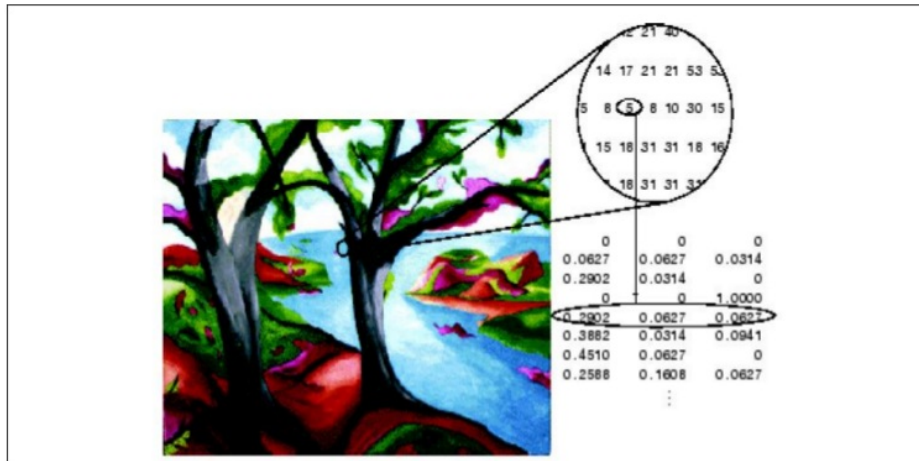
- **Citra biner** dikenal pula sebagai **citra Binary bilevel image**. Dimana matriknya berisi nilai 0 dan 1, mewakili warna putih. Contoh citra biner dapat dilihat pada gambar:



Gambar 4.1. Contoh citra binary/biner

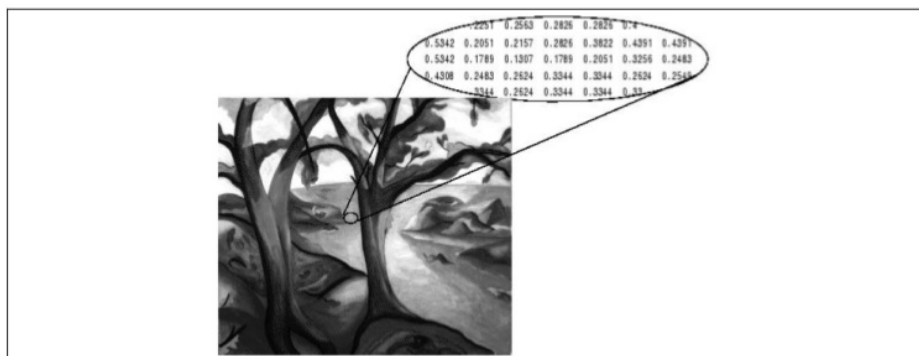
- **Citra indexed** dikenal pula sebagai citra *pseudocolor* . Citra ini merupakan array klas logika single , uint8, uint16 atau double yang nilai-nilai pikselnya di-indeks dalam peta warna. Peta warna

merupakan suatu array kelas double $m \times 3$. Untuk array single atau double kisaran nilai integer dari $[1, p]$. Sedangkan untuk array uint8, atau uint16 mempunyai kisaran nilai $[0, p-1]$.



Gambar 4.2. Contoh citra indexed

- **Citra grayscale** dikenal sebagai citra: *intensity*, *gray scale*, atau *gray level* : Klas array untuk uint8, uint16, int16, single, or double mempunyai nilai piksel tergantung nilai intensitasnya. Untuk array single atau double, mempunyai kisaran nilai $[0, 1]$. Untuk uint 8, mempunyai kisaran nilai dari $[0, 255]$. Untuk uint 16, kisaran nilai adalah dari $[0, 65535]$. Untuk int16, kisaran nilai dari $[-32768, 32767]$.



Gambar 4.3. Citra gray scale

- Citra truecolor dikenal sebagai citra RGB : array $m \times n \times 3$ array kelas uint8, uint16, single, atau double mempunyai nilai menurut nilai intensitasnya. Untuk array single atau double, mempunyai

kisaran nilai dari [0, 1]. Untuk unit 8, kisaran nilai dari [0, 255]. Untuk uint16, kisaran nilai dari [0, 65535].



Gambar 4.4. Contoh citra berwarna (RGB)

Untuk menentukan warna piksel pada (2,3), maka kita harus melihat triplet RGB yang disimpan pada (2,3,1:3). Misal (2,3,1) mengandung nilai 0.5176 , (2,3,2) mengandung 0.1608 dan (2,3,3) berisi 0.0627. Maka warna piksel (2,3) adalah 0.5176, 0.1608,dan 0.0627.

4.2 Konversi antar citra

Pada saat kita mengkonversi jenis citra satu ke yang lain, sering menghasilkan citra yang sedikit berbeda dengan citra asli. Terdapat beberapa fungsi atau commad untuk konversi citra dlam MatLab diantaranya: gray2ind, im2bw, ind2gray, rgb2gray, rgb2ind, and mat2gray.

4.2.1 rgb2gray

Fungsi ini digunakan untuk mengubah citra berwarna (RGB) menjadi citra gray scale. Rgb2gray digunakan untuk menghilangkan komponen Hue dan Saturation dan tetap menyimpan komponen Luminance.

Contoh penggunaan command ini adalah :

```
I = imread('monas.jpg');..... membaca citra yang bernama monas.jpg  
J = rgb2gray(I); ..... Mengguba citra berwarna monas ke citra grayscale  
figure, imshow(I), figure, imshow(J); ..... menampilkan citra asli dan citra hasil
```

4.2.2 im2bw

Mengubah citra berwarna ke citra binary image. im2bw menghasilkan citra biner dari citra indeks atau citra RGB, command ini mengubah citra input ke dalam format grayscale (jika belum berupa citra intensitas) dan menggunakan *thresholding* untuk mengubah citra grayscale ke citra biner. Output citra biner BW mempunyai nilai 1 (putih) dan 0 (hitam) untuk semua piksel. Contoh penggunaan command ini dapat ditulis dibawah ini.

```
load pohon ..... memanggil citra yang bernama pohon  
BW = im2bw(X,map,0.4); ..... mengubah citra intensitas menjadi citra hitam putih  
imshow(X,map), figure, imshow(BW)..... menampilkan citra asal dan citra hasil
```

4.2.3 gray2ind

Convert grayscale or binary image to indexed image.

`[X,map] = gray2ind(I,n)` converts the grayscale image I to an indexed image X. n specifies the size of the colormap, gray(n). n must be an integer between 1 and 65536. If n is omitted, it defaults to 64.

`[X,map] = gray2ind(BW,n)` converts the binary image BW to an indexed image X. n specifies the size of the colormap, gray(n). If n is omitted, it defaults to 2.

gray2ind scales and then rounds the intensity image to produce an equivalent indexed image

Convert a grayscale image into an indexed image and then view the result.

```
I = imread('cameraman.tif');  
[X, map] = gray2ind(I, 16);  
imshow(X, map);
```


4.2.4 gray2ind, im2bw, ind2gray, rgb2gray, rgb2ind, and mat2gray

ind2gray

Convert an indexed image to an intensity image

`I = ind2gray(X,map)` converts the image `X` with colormap `map` to an intensity image `I`. `ind2gray` removes the hue and saturation information from the input image while retaining the luminance.

load trees

```
I = ind2gray(X,map);
```

```
imshow(X,map)
```

```
figure,imshow(I)
```

4.2.5 rgb2ind

Convert an RGB image to an indexed image

`rgb2ind` converts RGB images to indexed images using one of three different methods: Uniform quantization Minimum variance quantization Colormap approximation

For all these methods, `rgb2ind` also dithers the image unless you specify 'nodither' for `dither_option`.

`[X,map] = rgb2ind(RGB,n)` converts the RGB image to an indexed image `X` using minimum variance quantization. `map` contains at most `n` colors. `n` must be less than or equal to 65536.

`X = rgb2ind(RGB,map)` converts the RGB image to an indexed image `X` with colormap `map` by matching colors in RGB with the nearest color in the colormap `map`. `size(map,1)` must be less than or equal to 65536.

`[X,map] = rgb2ind(RGB,tol)` converts the RGB image to an indexed image `X` using uniform quantization. `map` contains at most $(\text{floor}(1/\text{tol})+1)^3$ colors. `tol` must be between 0.0 and 1.0.

`[...] = rgb2ind(...,dither_option)` enables or disables dithering. `dither_option` is a string that can have one of these values: 'dither' (default) dithers, if necessary, to achieve better color resolution at the expense of spatial resolution. 'nodither' maps each color in the original image to the closest color in the new map. No dithering is performed.

```
RGB = imread('peppers.png');
```

```
[X,map] = rgb2ind(RGB,128);
```

```
imshow(X,map)
```

4.2.6 mat2gray

convert a matrix to a grayscale intensity image

`I = mat2gray(A,[amin amax])` converts the matrix `A` to the intensity image `I`. The returned matrix `I` contains values in the range 0.0 (black) to 1.0 (full intensity or white). `amin` and `amax` are the values in `A` that correspond to 0.0 and 1.0 in `I`.

`I = mat2gray(A)` sets the values of `amin` and `amax` to the minimum and maximum values in `A`.

```
I = imread('rice.png');
```

```
J = filter2(fspecial('sobel'),I);
```

```
K = mat2gray(J);
```

```
imshow(I), figure, imshow(K)
```

BAB V

FITUR CITRA DIGITAL

Tujuan intruksional umum : Memahami dan mengerti fitur-fitur yang dimiliki oleh suatu citra digital

Tujuan instruksional khusus : Mengetahui fitur – fitur warna, bentuk, dan tekstur

5.1 Tekstur

Fitur tekstur merupakan suatu karakteristik atau sifat yang dimiliki oleh citra, tekstur juga dianggap sebagai sifat kontekstual sehingga untuk mendefinisikannya mencakup nilai-nilai gray level dengan nilai piksel tetangganya. Ukuran atau size dari piksel tetangga tergantung pada tipe teksturnya, sehingga tekstur dapat didefinisikan sebagai:

- Tekstur merupakan distribusi spasial dari nilai gray level, sehingga merupakan histogram 2D atau dua matriks yang dapat dianalisis dengan tekstur analysis tool.
- Tekstur dalam suatu citra dapat sebagai perbedaan dua level resolusi suatu citra

Fitur statistik dapat digunakan sebagai metode untuk mengklasifikasikan tekstur berdasarkan gray level, GLCM atau Grey level co-occurrence matrices dapat digunakan untuk mengekstrak second order statistic dari suatu citra. GLCM dapat didefinisikan sebagai suatu matriks frekuensi dari dua piksel yang dipisahkan oleh suatu vektor yang ada dalam citra. Dimana distribusi frekuensi dalam matriks tergantung pada derajat dan jarak antara dua piksel tersebut.

Tekstur dapat juga didefinisikan sebagai susunan dari warna atau intensitas dalam suatu citra digital atau daerah tertentu dalam citra. Tekstur terdiri dari beberapa elemen (Texel). Beberapa masalah dalam analisis tekstur antara lain:

- Segmentasi dari texels sulit atau bahkan tidak mungkin dalam praktek sehari-hari.

- Perhitungan statistics yang menjelaskan texture dapat dihitung dari nilai grey atau warna. Metode ini kurang akurat tapi algoritma cuup efisien. Perhitungan ini dapat digunakan untuk klasifikasi citra dan segmentasi citra.

Beberapa perhitungan Statistik sederhana dari teksture:

- *Perhtungan Edge Density and Direction*
- Menggunakan edge detector sebagai langkah pertama dalam analisa tekstur.
- Jumlah piksel pinggir dalam suatu region yang menggambarkan betapa kompleksnya suatu region citra.
- Arah edges dapat digunakan untuk menentukan karakteristik dari tekstur.
- *Two Edge-based Texture Measures*

1. Penentuan pinggir(Edgeness)setiap region

$$\text{Fedgeness} = |\{ p \mid \text{gradient_magnitude}(p) \geq \text{threshold} \}| / N$$

Dimana N adalah ukura dari unit area

2. edge magnitude and direction histograms

$$F_{\text{magdir}} = (H_{\text{magnitude}}, H_{\text{direction}})$$

Dimana F_{magdir} adalah histograms yang telah dinormalisasiakn dari gradient magnitudes and gradient directions .

Co-occurrence Matrix Features

Suatu Matrik co-occurrence matrix merupakan array 2D array C , dimana:

- Kedua kolom dan rais mewakili nilai – nilai citra
 - $C(i,j)$ menunjukan berapa kali nilai i terjadi dengan j dalam suatu nilai a tertentu yang berkaitan dengan d.
 - Hubungan spatial dapat ditulis sebagai suatau vector $d = (d_r, d_c)$.

<table border="1"> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>2</td><td>2</td></tr> <tr><td>0</td><td>0</td><td>2</td><td>2</td></tr> <tr><td>0</td><td>0</td><td>2</td><td>2</td></tr> <tr><td>0</td><td>0</td><td>2</td><td>2</td></tr> </table>	1	1	0	0	1	1	0	0	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2		<table border="1"> <tr><td>0</td><td>1</td><td>0</td><td>3</td></tr> <tr><td>1</td><td>2</td><td>0</td><td>2</td></tr> <tr><td>2</td><td>0</td><td>0</td><td>1</td></tr> </table> C_d	0	1	0	3	1	2	0	2	2	0	0	1
1	1	0	0																																			
1	1	0	0																																			
0	0	2	2																																			
0	0	2	2																																			
0	0	2	2																																			
0	0	2	2																																			
0	1	0	3																																			
1	2	0	2																																			
2	0	0	1																																			
Citra grey scale		Matrik Co-occurrence																																				

Dari C dapat dihitung N , Normalisasi co-occurrence matrix, dimana setiap nilai dibagi dengan jumlah semua nilai.

Fitur Co-occurrence

$$Energy = \sum_i \sum_j N_d^2(i, j)$$

$$Entropy = -\sum_i \sum_j N_d(i, j) \log_2 N_d(i, j)$$

$$Contras = \sum_i \sum_j (i - j)^2 N_d(i, j)$$

$$Homogeneity = \sum_i \sum_j \frac{N_d(i, j)}{1 + |i - j|}$$

$$Correlation = \frac{\sum_i \sum_j (i - \mu_i)(j - \mu_j) N_d(i, j)}{\sigma_i \sigma_j}$$

Dimana μ_i, μ_j adalah mean dan σ_i, σ_j adalah standar deviasi dari baris dan kolom. Energy digunakan untuk menghitung homogenitas dari matrik yang telah dinormalisasi.

5.2 Tepi (edge)

Gambar 5.1 adalah contoh operasi pendeteksian tepi pada citra *Camera*. Operasi ini menghasilkan semua tepi (*edge*) di dalam citra.



Gambar 5.1. (a) *Citra camera*, (b) *citra hasil pendeteksian seluruh tepi*

BAB VI

JPEG dan JPEG 2000

6.1 JPEG

Joint Photographic Experts Group (JPEG) merupakan skema kompresi file bitmap. Awalnya, file yang menyimpan hasil foto digital memiliki ukuran yang besar sehingga tidak praktis. Dengan format baru ini, hasil foto yang semula besar dapat dimampatkan menjadi kecil.

Komite JPEG telah membuat banyak standar sejak tahun 1986. ISO sebenarnya telah bekerja sejak tahun 1983 untuk mencoba metoda bagaimana menambah kualitas grafik menjadi teks dari waktu ke waktu. Nama resmi dari standar Citra digital ini disebut 'JPEG' dan merupakan standar rekomendasi oleh ISO/IEC 109 18-1-ITU-T T.81 yang dipublikasikan oleh standar nasional ISO dan CCITT yang disebut ITU-T.

6.2. Lossy Compression

Dalam lossy compression Coding citra digital dikerjakan dengan beberapa langkah, yaitu:

- i. dibagi-bagi dalam sub-citra atau blok 8 x 8 dan diproses dari kiri ke kanan, atas ke bawah
- ii. Level shifting, yaitu setiap sub-citra atau blok dikurangi dengan 2^{n-1} dimana $2n$ adalah jumlah gray scale yang digunakan
- iii. Hitung koefisien DCT untuk setiap sub-citra
- iv. Kuantisasi setiap DCT
 - a. Bagi dengan matrik normal
 - b. Hasilnya dibulatkan dengan rumus berikut:

$$T(u,v) = DCT\{f(x,y)\}; \overset{v}{T}(u,v) = \text{round} \left\| \frac{T(u,v)}{Z(u,v)} \right\|$$

- v. Buat array berdimensi satu dari koefisien-koefisien yang telah dikuantisasi, dengan menggunakan model zig-zag
- vi. Koding array ini dengan menggunakan variabel run length dan length coding
 - a. Koefisien DC : selisih antara koefisien DC dari sub-citra dengan koefisien DC sebelumnya.

- b. Koefisien AC : Merupakan nilai yang bukan nol dari koefisien AC dan jumlah koefisien yang bernilai nol yang mendahului nilai bukan nol AC yang dihitung. EOB dan 16 kode yang bernilai nol yang mengikuti mempunyai kode-kode sub-citra (=blok)

Contoh JPEG coding

52	55	61	66	70	61	64	73	-76	-73	-67	-62	-58	-67	-64	-55
63	59	66	90	109	85	69	72	-65	-69	-62	-38	-19	-43	-59	-56
62	59	68	113	144	104	66	73	-66	-69	-60	-15	16	-24	-62	-55
63	58	71	122	154	106	70	69	-65	-70	-57	-6	26	-22	-58	-59
67	61	68	104	126	88	68	70	-61	-67	-60	-24	-2	-40	-60	-58
79	65	60	70	77	68	58	75	-49	-63	-68	-58	-51	-65	-70	-53
85	71	64	59	55	61	65	83	-43	-57	-64	-69	-73	-67	-63	-45
87	79	69	68	65	76	78	94	-41	-49	-59	-60	-63	-52	-50	-34

CITRA ASLI

LEVEL SHIFT

-415	-29	-62	25	55	-20	-1	3	-26	-3	-6	2	2	0	0	0
7	-21	-62	9	11	-7	-6	6	1	-2	-4	0	0	0	0	0
-46	8	77	-25	-30	10	7	-5	-3	1	5	-1	-1	0	0	0
-50	13	35	-15	-9	6	0	3	-4	1	2	-1	0	0	0	0
11	-8	-13	-2	-1	1	-4	1	1	0	0	0	0	0	0	0
-10	1	3	-3	-1	0	2	-1	0	0	0	0	0	0	0	0
-4	-1	2	-1	2	-3	1	-2	0	0	0	0	0	0	0	0
-1	-1	-1	-2	-1	-1	0	-1	0	0	0	0	0	0	0	0

DCT

KUANTISASI

ZIG-ZAG

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

CODING

1010110 0100 001 0100 0101 100001 0110 100011 001 100011
001 001 100101 11100110 110110 0110 11110100 000 1010

Sementara itu Decoding Lossy cokmpression citra digital dikerjakan dengan beberapa langkah , yaitu:

- i. Decoding array dan gunakan model zig-zag ke T(u,v)
- ii. Kalikan dengan matrik normal: 'T=(u,v) = "T(u,v)
- iii. Hitung 2D DICT : 'f(x,y) = DICT{T(u,v) Z(u,v)
- iv. Level Shit Back : Tambah 2n-1 untuk setiap sub-citra
- v. Letakan sub-sub citra kembali ke tempat yang benar

Decode array menurutb order zig-zag kembali ke sub-citra 8 x 8

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Mengalikan dengan matrik normal

-416	-33	-60	32	48	0	0	0
12	-24	-56	0	0	0	0	0
-42	13	80	-24	-40	0	0	0
-56	17	44	-29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

DICT

-70	-64	-61	-64	-69	-66	-58	-50	58
-72	-73	-61	-39	-30	-40	-54	-59	56
-68	-78	-58	-9	13	-12	-48	-64	60
-59	-77	-57	0	22	-13	-51	-60	69
-54	-75	-64	-23	-13	-44	-63	-56	74
-52	-71	-72	-54	-54	-71	-71	-54	76
-45	-59	-70	-68	-67	-67	-61	-50	83
-35	-47	-61	-66	-60	-48	-44	-44	93

Level shift back

64	67	64	59	62	70	78
55	67	89	98	88	74	69
60	50	70	119	141	116	80
69	51	71	128	149	115	77
74	53	64	105	115	84	65
76	57	56	74	75	57	57
83	69	59	60	61	61	67
93	81	67	62	69	80	84

Letakan sub-citra pada tempat yang sesuai

$$C_R = \frac{512}{92} \rightarrow R_D = 1 - \frac{1}{C_R} = 1 - \frac{92}{512} = 82\%$$

6.2 Discrete Cosine Transform

DCT dikerjakan dengan membagi citra dalam blok-blok 8x8, dimana x dan y adalah indeks suatu citra sementara u dan v menunjukkan indeks dari koefisien DCT. Sebelum DCT dikerjakan koefisien di-scale dengan menggunakan titik tengah dari range yang sudah ada.

$$F(u, v) = \frac{1}{4} C_u C_v \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

$$C_u = \frac{1}{\sqrt{2}}$$

Untuk $u = 0$, maka $C_u = 1$ sebaliknya $C_u = 0$ jika $u \neq 0$. Sedangkan 64 operasi untuk setiap koefisien DCT dapat dihitung dengan:

$$C_v = \frac{1}{\sqrt{2}}$$

Untuk $v = 0$, $C_v = 1$ dan $C_v = 0$ jika $v \neq 0$

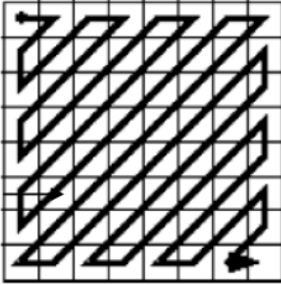
6.3 DCT Analysis

Analisa DCT dapat dikerjakan dengan mengatur spatial information ke dalam frekwensi-frekwensi sebagai berikut:

- ♦ Koefisien kiri atas (koefisien pertama) yang disebut koefisien DC dan sisanya disebut koefisien AC. Mata manusia kurang sensitif terhadap koefisien AC (frekwensi tinggi). Sehingga semakin tinggi frekwensi AC maka memerlukan kuantisasi lebih cepat karena kehilangan informasi sulit diketahui pada saat normalisasi.
- ♦ Kuantisasi
- ♦ Setelah DCT maka 64 koefisien, koefisien DC yang ada di kiri atas dan koefisien lainnya disebut koefisien 63 sebanyak 63. Kuantisasi dikerjakan dengan menggunakan tabel kuantisasi yang dibuat berdasarkan Sifat dan karakteristik mata manusia. Warna dan frekwensi-frekwensi yang lebih tinggi kurang penting dibandingkan dengan Luminance dan frekwensi rendah dapat disimpan dalam table. Berbagai jenis tabel dan nilai dapat digunakan berdasarkan ratio atau kualitas yang diinginkan.

6.4 ZigZag Scanning

Zig zag scanning adalah proses pengaturan kemabli matrik 8x8 yang telah dikuantisasi kedalam aturan zig-zag.



Modified Huffman Coding

Codewords dipakai untuk to descriptors, dimana setiap {Zero Run Length/Magnitude Category} diberi codeword.

Run/Category	Code Length	Codeword
0/1	2	00
2/3	10	1111110111
4/1	6	111011
6/2	12	111111110110

Penambahan bits yang ada menunjukkan nilai tertentu dalam kategori magnitude yang digunakan. Predictive coding juga digunakan pada koefisien DC dari blok satu ke blok yang lain. Hal ini dapat secara dinamis dan statis, Huffman codes bekerja dengan menggunakan code-kode yang lebih pendek. Jika table code table dapat berubah maka tabel-tabel dapat lebig digunakan secara optimal setiap citra.

Format file JPEG adalah yang paling canggih yang tercatat di awal. Ini mencapai rasio kompresi yang sangat baik bagi sebagian besar jenis citra, biasanya 30:1 pada saat true color. Teknik kompresi yang digunakan oleh JPEG dimulai dengan warna subsampling psychovisual dan kuantisasi. DCT kompresi ini kemudian diterapkan ke satu 8 x 8 blok pada waktu dan komponen frekuensi tidak penting domain dieliminasi. Akhirnya sisa komponen dikodekan secara optimal menggunakan Huffman coding. Sayangnya metode kompresi beroperasi perlahan dan juga sangat sulit untuk dilaksanakan. Namun, chip VLSI yang

tersedia yang dapat melakukan coding JPEG sangat cepat di hardware. Kelompok JPEG kerja juga telah menempatkan encoding dan decoding perangkat lunak dalam domain publik.

Bahkan JPEG mendukung berbagai pilihan kualitas / kompresi, mulai dari kompresi sedang hampir lossless untuk kompresi lossy tapi sangat efisien. Tingkat kualitas dapat ditentukan, biasanya dinyatakan sebagai persentase. Efisiensi kompresi yang diberikan sebelumnya adalah untuk pengaturan kualitas 75%. Pengaturan Tinggi memberikan kualitas yang lebih tinggi tetapi kurang dan pengaturan kompresi lebih rendah memberikan kualitas yang lebih rendah tapi kompresi lebih besar. Tabel di bawah menunjukkan efisiensi kompresi untuk citra bumi dengan berbagai pengaturan kualitas. Dua angka yang ditampilkan untuk setiap pengaturan, efisiensi relatif terhadap citra terkompresi palet warna (8 bit per piksel) dan efisiensi data true color (24 bit per piksel).

Kompresi sampai dengan 75%, biasanya masih menghasilkan efisiensi kompresi yang baik tanpa pengurangan yang signifikan dalam kualitas citra asli. Pengaturan kualitas rendah dapat mencapai pengurangan besar dalam ukuran citra, namun citra yang dihasilkan mungkin akan terlihat sangat jelek ketika decompressed. Tingkat kualitas terendah yang diterima akan tergantung pada citra asli.

Huffman Coding

Huffman Coding adalah teknik kompresi lossless yang mengeksploitasi redundansi statistik. Huffman coding mengurangi besar dalam ukuran untuk setiap jenis citra dari encoding panjang berjalan namun membutuhkan waktu eksekusi lebih signifikan. Rasio kompresi biasanya 1:01-8:01 tergantung pada isi citra. Huffman Coding menggantikan setiap nilai piksel dengan pola bit variabel panjang dan paket ke dalam aliran bit berkesinambungan. Nilai piksel paling sering diwakili oleh pola bit pendek dan nilai-nilai yang sering kurang adalah dikodekan sebagai pola lama. (Tentu saja pola-pola ini sedikit harus dirancang sehingga panjang mereka dapat diidentifikasi ketika bit stream diekstrak.

Untuk menyederhanakan kita akan mengcitrakan Huffman Coding untuk citra 1024 x 768 x terdiri dari hanya 8 tingkat intensitas. Jika setiap piksel disimpan sebagai integer 3-bit maka citra terkompresi akan menempati Kbits 2304 (288 Kbytes). Misalkan intensitas piksel didistribusikan seperti terlihat pada tabel berikut. Frekuensi dinyatakan sebagai persentase dari jumlah total untuk kemudahan (misalnya 40% dari semua piksel memiliki intensitas dari 2).

Tabel . Huffman coding

Intensitas	Frekwensi (%)	Kode Baku	Kode Piksel Huffman
0	0	000	-
1	10	001	011
2	40	010	1
3	6	011	01010
4	10	100	0100
5	4	101	01011
6	30	110	00
7	0	111	-

Dua yang terakhir kolom dari tabel di atas menunjukkan kode biner 3-bit dari nilai piksel asli dan variabel panjang kode Huffman yang akan menggantikan mereka dalam bit stream dikemas. Intensitas yang paling sering dikompresi dari 3 bit untuk 1 bit sedangkan yang paling sering diperluas untuk 5 bit (nilai-nilai yang tidak pernah terjadi diabaikan). Kita bisa bekerja di luar rata-rata jumlah bit yang diperlukan untuk mewakili setiap piksel dari frekuensi berhubungan dengan setiap panjang kode Huffman. 40% dari piksel yang disimpan dalam 1 bit, 30% dalam 2 bit, 10% dalam 3 bit, 10% pada 4 bit dan 10% dalam 5 bit. Hal ini memberikan rata-rata 2,2 bit / piksel yang setara dengan penurunan 27% atau 77 Kbytes untuk seluruh citra. (Perhatikan bahwa sejumlah kecil memori juga diperlukan untuk mencatat tugas kode Huffman.) Huffman Coding bekerja lebih baik untuk citra yang mengandung tingkat intensitas lebih.

Lempel-Ziv-Welch (LZW) coding

Lempel-Ziv-Welch (LZW) coding adalah metode statistik yang digunakan dalam kompresi lossless. Metode ini memberikan rasio kompresi untuk Huffman coding, sebesar 1:01. Namun, LZW coding cenderung lebih cepat karena citra dipindai hanya sekali. LZW mungkin adalah teknik kompresi file grafik yang paling banyak digunakan saat ini. Compuserve's Graphics Interchange Format (GIF).

LZW coding bekerja dengan mencari urutan umum dari nilai dalam data. Sebuah tabel nilai setiap piksel. Ketika urutan baru ditemukan itu diberi nilai kode dan ditambahkan ke tabel.

Ketika sebuah urutan yang dikenal ditemui lagi maka hanya nilai kode yang perlu dikelurakan. Pendekatan ini bekerja baik pada kebanyakan citra.

Arithmetic coding

Arithmetic coding adalah teknik kompresi lossless yang paling efisien. Seperti Huffman Coding, arithmetic coding juga menggunakan metode statistik berdasarkan frekuensi yang masing-masing dengan nilai piksel yang ada. Namun, proses encoding cukup berbeda dengan Huffman. Arithmetic decoding dikerjakan dengan mengganti setiap nilai piksel dengan pola bit tertentu, seluruh citra dikodekan sebagai angka multi-presisi tunggal. Berturut-turut nilai piksel digabungkan deret hitung dengan cara yang dapat memerlukan waktu kurang dari satu bit untuk mewakili nilai-nilai yang paling sering muncul. Hal ini memungkinkan untuk mencapai rasio kompresi hingga 100:1. Sayangnya pengkodean aritmatika sangat sulit untuk dilaksanakan. Metode ini telah dipatenkan oleh IBM, sehingga penggunaannya juga harus ada izin dari IBM.

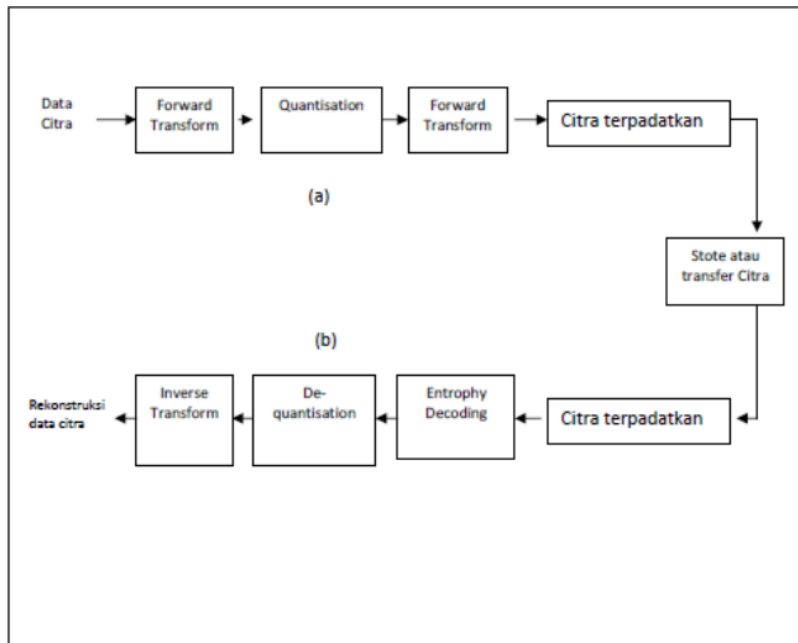
JPEG 2000

JPEG 2000 merupakan standar kompresi citra dan sistem koding, JPEG 2000 diperkenalkan oleh Joint Photographic Experts Group pada tahun 2000 yang merupakan pengembangan dari JPEG yang menggunakan teknologi DCT yang dibuat pada tahun 1992, sementara JPEG 2000 menggunakan teknologi Wavelet transform. Format JPEG 2000 menggunakan ekstension .jp2 yang digunakan di ISO/IEC 15444-1 dan .jpx menurut ISO/IEC 15444-2.

JPEG 2000 mempunyai performance yang lebih baik dibandingkan dengan JPEG terutama lebih fleksible dalam codestream-nya. Namun sebagai sekuensinya JPEG 2000 memerlukan encoder dan decoder yang cukup kompleks. Perbedaan lain dibandingkan JPEG, JPEG 2000 menghasilkan apa yang disebut ringing artifact, yang merupakan paten blur sebagai lingkaran tepi sementara JPEG menghasilkan ringing artifact dan blocking artifact karena adanya blok 8 x 8.

JPEG 2000 diperlukan untuk mengurangi kekurangan atau kelemahan yang ada pada JPEG, JPEG 2000 diperlukan mengatasi antara lain:

- **Kompresi Low bit-rate dibawah 0.25 bpp , dimana JPEG tidak menyediakan**
- **Kompresi Lossless and lossy compression dalam satu codestream.**
- **Kompresi citra yang berukuran lebih dari 64 x 64 tetap menggunakan tiling.**



Gambar . Diagram Blok JPEG2000 (a) encoding, (b) decoding

Skema encoder-decoder

Tiling

Setelah transformasi warna, citra dipisah ke dalam bagian – bagian yang disebut *tiles*, tile merupakan daerah segi empat dari suatu citra yang ditransforasikan dan di- encode secara terpisah. Tiles dapat

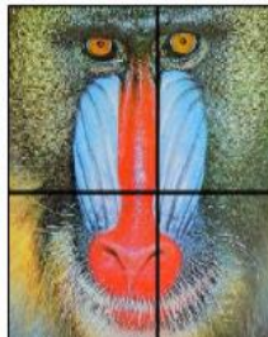
bermacam-macam ukuran yang kadang juga dianggap sebagai sebagai satu citra dengan satu tile. Sekali sebuah ukuran dipilih maka semua tile akan mempunyai ukuran yang sama kecuali tile yang ada terletak pada sebelah kanan bawah . Membagi citra ke dalam tiling mempunyai keuntungan anantara lain dalam proses decoder memerlukan momory yang tidak terlalu besar untuk men-decode citra dan hanya tile-tile tertentu yang di-decode sesuai dengan keperluan terutama untuk partial decoding suatu citra. Namun kelemahan dari metode ini adalah kualitas citra mengalami penurunan karena rendahnya rasio signal-to-noise. Dengan menggunakan banyak tile maka akan di-generate suatu blok yang pengaruhnya hampir sama dengan JPEG standar.

1. Proses tiling citra:

- Citra yang cukup besar ukuran memory-nya yang bisa di decode
- Citra kemudian dibagi menjadi bagian-bagian yang berbentuk epp (tidak overlap) dalam blok-blok atau tile akan dipadatkan sendiri-sendiri.

2. DC-level shifting:

-
- Codec mengharapkan input data mempunyai kisaran dinamis antara (0 -- 255 -> -128 -- 128)
- Jika nilai input sample merupakan bilangan unsigned, maka nilai nominal dinamis disesuaikan dengan mengurangi dengan nilai (2^{P-1} , dimana P adalah komponen presisi).
- Semua operasi , termasuk komponen mixing, wavelet transform, quantization and entropy coding dikerjakan secara terpisah pada setiap tile yang ada di dalam citra.
- Proses tiling mempengaruhi kualitas citra

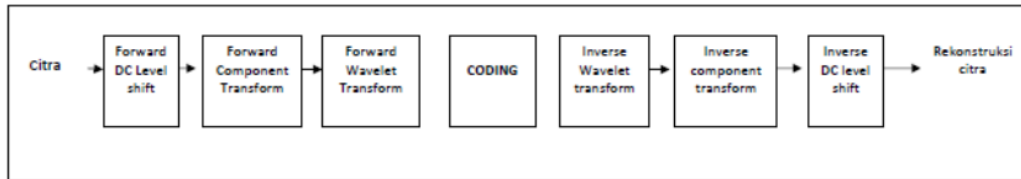


Gambar. Contoh tiling dari citra 8 bit

Pengaruh Tiling terhadap Kualitas citra			
Tiling	Tidak di-Tiling	Tiling denga ukuran 128 x 128	Tile dengan ukuran 64 x 64
0,125	24,75	23,42	20,07
0,25	26,49	25,69	23,95
0,5	28,27	27,29	26,80

. Components transformation:

- Petakan data dari RGB to YCrCb (Y, Cr, Cb), perlu untuk mengurangi korelasi antara komponen-komponen, meningkatkan efisiensi coding. Terdapat transform reversible and irreversible.



Gambar . Penempatan DC level shifting dengan komponen transform

Komponen transform Forward reversible

$$Y_0(x, y) = \left\lfloor \frac{I_0(x, y) + 2I_1(x, y) + I_2(x, y)}{4} \right\rfloor$$

$$Y_1(x, y) = I_2(x, y) - I_1(x, y)$$

$$Y_2(x, y) = I_0(x, y) - I_1(x, y)$$

Komponen transform Inverse reversible

$$I_1(x, y) = Y_0(x, y) - \left\lfloor \frac{Y_2(x, y) + Y_1(x, y)}{4} \right\rfloor$$

$$I_0(x, y) = Y_2(x, y) + I_1(x, y)$$

$$I_2(x, y) = Y_1(x, y) + I_1(x, y)$$

BAB VII

CONTENT BASED IMAGE RETRIEVAL

- Tujuan intruksional umum* : *Memahami bagaimana algoritma matching dua citra digital berdasarkan fitur warna, bentuk, dan tekstur*
- Tujuan instruksional khusus* :
 - *Mengerti teknik-teknik CBIR*
 - *Memahami bagaimana matching citra query dengan citra dalam database*

Content based image retrieval dapat diartikan sebagai pencarian secara otomatis (automatic Retrieval) dari suatu citra database dengan menggunakan fitur warna dan bentuk. Fitur yang digunakan dapat berdasarkan fitur primitive maupun semantik. Satu hal penting dalam semua pengolahan citra adalah ekstraksi informasi dari suatu citra (seperti pengenalan bentuk atau tekstur) sebelum analisa citra lebih lanjut. Database citra berbeda dengan database teks yang berupa huruf/kata yang disimpan sebagai karakter string ASCII. CBIR merupakan bagian dari contoh pengolahan citra, CBIR menekankan pada retrieval citra yang memiliki karakteristik dari suatu koleksi citra. Sementara itu pengolahan citra lebih luas cakupannya seperti image enhancement, compression, transmisi, dan interpretasi, objek recognition).

Beberapa hal perlu diperhatikan dalam CBIR diantaranya adalah:

- Pemahaman user terhadap kebutuhan dan information-seeking behavior.
- Identifikasi teknik yang cocok bagaimana menerjemahkan content dari citra
- Ekstraksi fitur citra awal (raw image)
- Bagaiman menyediakan storage untuk database citra yang biasanya cukup besar
- Matching citra query dan citra yang ada dalam database
- Efisiensi

7.2 Database citra digital

Terdapat banyak contoh database citra digital yaitu Art Collections, medical, scientific, earth sciences, general image collections for licencing, dan WWW. Beberapa contoh lain adalah :

- Art Collections : Koleksi seni pada museum-museum
- Medical Image Databases : CT, MRI, Ultrasound, The Visible Human
- Scientific Databases: Earth Sciences

- General Image Collections for Licensing : Corbis, Getty Images
- The World Wide Web

Dalam CBIRW diperlukan image query, query citra dapat berupa citra (image) disebut sebagai Query by Example (QBE), sketsa, dan symbol. Dari macam query tersebut maka yang paling sering digunakan query by Example

7.3 Fitur citra

Dalam CBIR matching citra digital didasarkan fitur-fitur citra yang, fitur tersebut antara lain: warna (histograms, gridded layout, wavelets), tekstur (Laws, Gabor filters, local binary partition), bentuk (first segment the image, then use statistical or structural shape similarity measures), dan objek dan hubungan diantara objek tersebut

7.4 Query-by-Example

Seperti diketahui bahwa suatu citra mempunyai beberapa karakteristik atau atribut yang dapat digunakan untuk keperluan retrieval, atribut tersebut antara lain:

- warna, tekstur, dan bentuk
- Susunan atau adanya tipe objek tertentu (misalnya meja dikelilingi kursi)
- Hadirnya atau adanya kejadian-kejadian tertentu seperti pertandingan sepak bola
- Kehadiran seseorang, lokasi atau peristiwa-peristiwa tertentu seperti Preside sedang berpidato
- Foto emosi seseorang seperti senang, susah, dan stress.
- Metadata seperti siapa yang mengambil foto, dimana, dan kapan.

Query dapat dibagi menjadi beberapa level, dimana semakin tinggi level semakin meningkat kompleksitasnya

- **Level 1** merupakan tipe retrieval dengan menggunakan fitur *primitive seperti* warna, texture, shape atau spatial location dari elemen citra. Contoh query ini adalah “cari foto atau objek yang tipis dan gelap yang ada pada kira atas., “Cari citra bitang-bintang yang berwarna kuning yang berbeuk lingkaran” secara umum level ini “ Cari gambar yang mirip dengan gambar ini”.

- **Level 2** merupakan query dengan menggunakan fitur *derived* (biasanya dikenal sebagai fitur *logical*). Dalam level ini melibatkan beberapa logical inference tentang identitas about the identity of the objects yang diwakili dalam citra. Level ini dapat dibedakan menjadi dua kategori:

- a) Retrieval objek menurut jenis misalnya, Cari gambar bus way.
- b) Retrieval objek atau orang misalnya. Cari gambar Monas.

Level 3 merupakan retrieval dengan menggunakan atribut *abstract*, atribut abstrak mencakup significant amount of high-level reasoning tentang arti dan tujuan objects yang ada. Level ini dapat dibagi menjadi :

- a) Retrieval kejadian atau aktifitas contoh “Cari gambar orang sedang menari”
- b) Retrieval gambar wajah dengan ekspresi emosi (“Cari gambar wajah stress”).

7.5 Matching image query

Fitur Warna

Fitur warna dikerjakan dengan menggunakan jarak histogram warna yang mirip, perhitungan jarak histogram warna dapat dihitung dengan menggunakan rumus:

$$d_{hist}(I, Q) = (h(I) - h(Q))A(h(I) - h(Q))$$

Dimana $h(I)$ adalah histogram database citra dengan K -bin, sedangkan $h(Q)$ adalah histogram citra query. Sementara A merupakan matrik $K \times K$.

Menghitung fitur tekstur

Jarak fitur tekstur dikerjakan tahapan:

- Retrieve citra yang ada dalam region yang mempunyai tekstur yang sama atau mirip dengan region disekitarnya.
- Kemudian Gridded (just like gridded color, but use texture).
- Histogram-based (e.g. compare the LBP histograms).

Menghitung Jarak Fitur Bentuk

Fitur bentuk jauh lebih kompleks dibanding warna dan tekstur, bentuk memerlukan identifikasi dari region untuk membandingkan citra. Terdapat beberapa cara untuk mengukur kemiripan dari *pattern recognition* yang dapat digunakan untuk mengukur distance untuk, antara lain:

- Global Shape Properties Projection Matching,
- In projection matching, yaitu proyeksi horizontal dan vertical dari suatu histogram.
- Boundary Matching dengan menggunakan Fourier Descriptors DAN sudut dan sisi).

Sedangkan jarak antara query bentuk dan bentuk citra mempunyai dua komponen:

- Energi yang diperlukan untuk energi untuk mengubah query bentuk kedalam satu dari bentuk citra yang paling cocok.
- Pengukuran bagaimana deformed query yang cocok dengan citra.

Beberapa langkah untuk matching citra, yaitu:

- Segmentasi citra kedalam beberapa region
- Cari properties mereka dan hubungan diantara mereka
- Buat graph yang merepresentasikan region dan tepi dan hubungan diantara keduanya
- Gunakan matching untuk membandingkan dengan citra Tigaer sebagai suatu graph z

7.6 CBIR dalam compressed domain

Berdasarkan standar kompresi JPEG, proses dekompresi yang lengkap dapat diringkas sebagai: *i.* decoding entropi melalui tabel Huffman, *ii.* dequantization untuk mendapatkan koefisien DCT, dan *iii.* IDCT untuk merekonstruksi blok piksel. Untuk setiap sistem CBIR yang bekerja dalam domain piksel, ini adalah syarat operasi atau operasi pendahuluan sebelum ekstraksi kunci dan matching distance yang akan

7.7 Aplikasi CBIR dalam kehidupan sehari-hari

7.6.1 Pencegah tindak Kejahatan (Crime prevention)

Penerapan CBIR dibidang ini contohnya seperti foto wajah pelaku tidak kejahatan, sidik jari,

Law enforcement agencies typically maintain large archives of visual evidence, including past suspects' facial photographs (generally known as mugshots), fingerprints, retina mata, palm print. Face recognition (pengenalan wajah)

Daftar Pustaka

- Irianto, Suhedro. 2013. Segmentation for Image Indexing and Retrieval on Discrete Cosines Domain. TELKOMNIKA Vol. 11, No. 1, 119 - 126 ISSN: 1693-6930 Accredited by DGHE (DIKTI), Decree No: 51/Dikti/Kep/2010.
- Irianto, Suhendro. 2008. Image Retrieval In The Compressed Domain: Integration of low level features, image keywords and descriptions, segmentation techniques and image ontology for image retrieval in the compressed domain. Ph.D Thesis, University Of Bradford, United Kingdom.
- Saphiro, G. Linda dan Gorge C. Stockman. Computer Vision. 2001. Prentice Hall, Upper Saddle, New Jersey, U.S.A.
- Smeulders , Arnold. W.M., Amarnath Gupta, Marcel Worring, dan Simone Santini. 2000. Content-Based Image Retrieval At The End Of The Early Years. IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 22, No. 12, 123-142
- Thompson, Clay dan Lauren Shure. Image Processing ToolBox. 2012. User's Guide. The Math Works, Inc. 3 Apple Hill Drive Natick, Massachusetts, U.S.A



PCD Buku

ORIGINALITY REPORT

18%

SIMILARITY INDEX

17%

INTERNET SOURCES

4%

PUBLICATIONS

3%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

1%

★ elib.unikom.ac.id

Internet Source

Exclude quotes Off

Exclude bibliography Off

Exclude matches Off

PCD Buku

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

Instructor

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60

PAGE 61

PAGE 62

PAGE 63

PAGE 64

PAGE 65

PAGE 66

PAGE 67

PAGE 68

PAGE 69

PAGE 70

PAGE 71

PAGE 72

PAGE 73

PAGE 74

PAGE 75

PAGE 76

PAGE 77

PAGE 78

PAGE 79

PAGE 80

PAGE 81

PAGE 82

PAGE 83

PAGE 84

PAGE 85

PAGE 86

PAGE 87

PAGE 88

PAGE 89

PAGE 90

PAGE 91
