

BUKU AJAR

PENGOLAHAN CITRA



Penyusun:
Nana Ramadijanti,S.Kom.,M.Kom.
NIP. 19711109.1998.02.2.001

Drs. Achmad Basuki,M.Kom.,Ph.D.
NIP. 19690112.1994.03.1.002

Fadilah Fahrul H., S.ST.
NIP. 2000000061

Program Studi Teknik Informatika
Departemen Teknik Informatika dan Komputer
Politeknik Elektronika Negeri Surabaya
2014

HALAMAN PENGESAHAN

Dengan ini saya yang bertanda tangan di bawah ini :

Nama : Arif Basofi, S.Kom, MT.
NIP : 197609212003121002
Pangkat/Golongan : Penata/IIIC

Menyetujui Judul dan Isi dari Buku Ajar/~~Laporan Môdul Praktikum*~~ dengan:

Judul : Pengolahan Citra

Yang diusulkan oleh:

Nama : Nana Ramadijanti, S.Kom.,M.Kom.
NIP : 197111091998022001
Pangkat/Golongan : Pembina Tk 1/ IV b

Telah sesuai dengan kebutuhan proses belajar mengajar pada mata kuliah/~~praktikum*~~ Pengolahan Citra pada program studi D3 Teknik Informatika Politeknik Elektronika Negeri Surabaya.

Surabaya, 13 Nopember 2014

Mengetahui,
Ketua UPPM,

Menyetujui
Ketua Program Studi D3 Teknik Informatika

Eko Henfri Binugroho, SST, MT
NIP 19791223.200312.1.002

Arif Basofi, S.Kom.,M
NIP: 197609212003121002

KATA PENGANTAR

Alhamdulillah, Segala puji hanya untuk Allah SWT, Tuhan Yang Maha Berilmu. atas petunjukNya-lah kami dapat menyelesaikan buku ajar mahasiswa program studi D3 dan D4 Teknik Informatika untuk matakuliah Pengolahan Citra. Diharapkan dengan adanya buku ini, mahasiswa mendapatkan referensi dalam mengikuti perkuliahan Teori dan Praktikum Pengolahan Citra.

Materi yang disusun pada buku ini sesuai dengan silabus matakuliah Pengolahan Citra di program studi D3 dan D4 Teknik Informatika. Jumlah bab pada buku ini ada 9 (sembilan) bab. Isi buku ajar setiap bab disusun mulai dari materi ajar, ringkasan materi, dan soal-soal latihan. Ucapan terima kasih tak lupa kami sampaikan kepada beberapa pihak yang telah memberikan kontribusi dalam penyelesaian buku ajar Pengolahan Citra ini, yaitu:

- Dr. Zainal Arief,ST.,MT.,selaku Direktur PENS
- Ir. Anang Budikarso,MT.,selaku Pembantu Direktur II PENS
- Eko Henfri Binugroho, SST, MSc,selaku Ketua Unit Penelitian PENS
- Arna Fariza,S.Kom.,M.Kom.,selaku Ketua Departemen Teknik Informatika dan Komputer PENS
- Arif Basofi,S.Kom.,M.T.,selaku Ketua Program Studi D3 Teknik Informatika PENS
- Isbat Uzzin Nadhori,S.Kom.,M.T.,selaku Ketua Program Studi D4 Teknik Informatika PENS
- Dosen-dosen dan karyawan PENS,khususnya rekan-rakan di program studi D3 dan D4 Teknik Informatika PENS

Sekalipun buku ini telah selesai dengan proses yang cukup panjang, akan tetapi masih tidak menutup adanya kekurangan padanya. Segala masukan, kritik dan review sangat kami harapkan untuk semakin menyempurnakannya pada kesempatan mendatang.

Surabaya, Oktober 2014

PENYUSUN

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	iv
DAFTAR GAMBAR	viii
DAFTAR TABEL	xii
Bab 1.....	1
Pengantar Pengolahan Citra.....	1
1.1.Pengantar Pengolahan Citra	1
1.2.Dasar Dasar Pengolahan Citra	3
1.2.1.Teknik-Teknik Pengambilan Citra Digital.....	4
1.2.2.Model Citra Digital	12
1.2.3.Teknik-Teknik Pengolahan Citra Digital.....	14
1.3.Ringkasan	19
1.4.Latihan.....	20
Bab 2.....	21
Dasar Citra Digital	21
2.1. Sistem Penglihatan Manusia Dan Komputer.....	22
2.1.1. Sistem Visual Manusia.....	23
2.1.2. Adaptasi Dan Diskriminasi Brightness	25
2.1.3. Sensitivitas Kontras.....	26
2.1.4. Rasio Weber.....	26
2.1.5. Brightness Sebagai Fungsi Intensitas.....	27
2.1.6. Pola Mach Band.....	27
2.1.7. Simultan Kontras	28
2.2. Sinyal.....	28
2.2.1. Sinyal Waktu.....	28
2.2.2. Sinyal Spatial.....	29
2.2.3. Sinyal Spatialtemporal.....	29
2.2.4. Tipe sinyal	29
2.2.5.Analog Dan Digital.....	29
2.2.6.Sampling.....	29
2.2.7.Kuantisasi	30
2.3.Representasi Citra Digital	30
2.3.1.Fungsi Intensitas Cahaya	31
2.3.2.Gray Level	32
2.3.3.Jumlah Bit	32
2.3.4.Resolusi.....	32
2.3.5.Resolusi Spatial.....	32
2.3.6.Resolusi Kecemerlangan	33
2.3.7.Efek Checkerboard	35
2.3.8.False Counturing	35
2.3.9.Sampling Non Uniform	36
2.3.10.Kuantisasi Non Uniform	36
2.4.Relasi Dasar Piksel Pada Citra B/W	37
2.4.1.Tetangga Piksel	37
2.4.2.Konektivitas	37
2.4.3.Pelabelan Komponen Terhubung.....	38
2.4.4.Pengukuran Jarak	40
2.4.5.Operasi Aritmatika Dan Logika.....	42
2.5.Ringkasan	44
2.6.Latihan.....	44

Bab 3.....	45
Peningkatan Mutu Citra Domain Spatial	45
3.1. Jenis Teknik Peningkatan Mutu Citra.....	45
3.1.1.Citra yang Baik.....	46
3.2.Domain Spatial	46
3.3.Mask/Filter	46
3.4.Pemrosesan Piksel	47
3.5.Kontras Stretching	47
3.6.Thresholding	49
3.7.Transformasi Tingkat Keabuan.....	51
3.7.1.Fungsi Identitas	52
3.7.2.Citra Negatif	52
3.7.3.Transformasi Logaritma	52
3.7.4.Transformasi Inverse Logaritma.....	53
3.7.5.Transformasi Algoritma Power Law	53
3.7.5.1.Koreksi Gamma	54
3.7.5.2.Citra MRI.....	55
3.7.5.3.Efek Penurunan Gamma.....	55
3.8.Tranformasi Fungsi Linier Piecewise.....	56
3.8.1.Gray Level Slicing	56
3.8.2.Bit Plane Slicing	57
3.9.Pemrosesan Dengan Histogram	58
3.10.Histogram Normalisasi	59
3.11.Histogram Equalisasi	60
3.12.Histogram Spesifik	65
3.13.Peningkatan Mutu Citra Secara Lokal	66
3.14.Peningkatan Mutu Citra dengan Operasi Logika dan Aritmatika.....	67
3.14.1. Operasi Logika	67
3.14.2. Operasi Aritmatika Pengurangan Citra	68
3.14.3. Operasi Aritmatika Averaging Citra.....	69
3.15.Ringkasan	70
3.16.Latihan.....	70
Bab 4.....	72
Tranformasi Fourier	72
4.1. Dasar-dasar Transformasi Fourier.....	72
4.1.1. Transformasi Fourier 1D	73
4.1.2. Transformasi Fourier 2D	74
4.2. Transformasi Fourier Diskrit	76
4.2.1. DFT 1D	76
4.2.2. Transformasi Fourier Diskrit 2D	79
4.3. Fast Fourier Transform	81
4.3.1. FFT 1D	81
4.3.2. FFT 2D	83
4.5. Transformasi Cosinus Diskrit.....	86
4.5.2. DCT 2D	87
4.6.Ringkasan	88
4.7.Latihan.....	90
Bab 5.....	91
Filtering.....	91
5.1.Prinsip-Prinsip Filtering.....	91
5.2. Konvolusi.....	96
5.3. Low Pass Filter (Reduksi Noise).....	100
5.3.1. Membangkitkan Noise Uniform	101
5.3.2. Membangkitkan Noise Gaussian	101
5.3.3. Membangkitkan Noise Salt & Pepper.....	102

<i>5.3.4. Membangkitkan Noise Speckle</i>	102
<i>5.3.5. Reduksi Noise Menggunakan Filter Rata-Rata</i>	102
<i>5.3.6. Reduksi Noise Menggunakan Filter Gaussian</i>	105
<i>5.4. High Pass Filter (Deteksi Tepi)</i>	107
<i>5.4.1. Metode Robert</i>	109
<i>5.4.2. Metode Prewitt</i>	110
<i>5.4.3. Metode Sobel</i>	110
<i>5.5. Ringkasan</i>	111
<i>5.6. Latihan</i>	111
Bab 6.....	114
Citra Berwarna.....	114
<i>6.1. Konsep Warna</i>	114
<i>6.2. Ruang Warna</i>	115
<i>6.2.1. RGB</i>	115
<i>6.2.2. CMY(K)</i>	116
<i>6.2.3. HSV</i>	118
<i>6.2.4. CIE XYZ</i>	118
<i>6.2.4. Lab - Photoshop</i>	119
<i>6.3. Color Gamut</i>	120
<i>6.4. Konversi Warna</i>	121
<i>6.5. Ringkasan</i>	123
<i>6.6. Latihan</i>	124
Bab 7.....	125
Fitur Pada Citra	125
<i>7.1. Fitur Warna Citra</i>	125
<i>7.2. Histogram Warna</i>	126
<i>7.2.1. Histogram RGB</i>	127
<i>7.2.2. Histogram HSV</i>	130
<i>7.3. Fitur Bentuk</i>	131
<i>7.3.1. Deteksi Tepi</i>	131
<i>7.3.2. Histogram Proyeksi</i>	132
<i>7.4. Ringkasan</i>	133
<i>7.5. Latihan</i>	134
Bab 8.....	135
Morfologi.....	135
<i>8.1. Morfologi</i>	135
<i>8.2. Struktur Elemen</i>	136
<i>8.3. Dilasi</i>	136
<i>8.4. Erosi</i>	138
<i>8.5. Opening</i>	140
<i>8.6. Closing</i>	141
<i>8.7. Hit-or-Miss Transform</i>	143
<i>8.8. Thinning</i>	143
<i>8.9. Ringkasan</i>	147
<i>8.10. Latihan</i>	147
Bab 9.....	149
Aplikasi Pengolahan Citra	149
<i>9.1. Pengenalan Kematangan Buah Tomat</i>	149
<i>9.1.1. Data Referensi</i>	150
<i>9.1.2. Blok Diagram Sistem</i>	151
<i>9.1.3. Ekstraksi Fitur</i>	152
<i>9.1.4. Proses Matching</i>	155
<i>9.1.4. Hasil Percobaan</i>	155
<i>9.2. Pengenalan Nilai Nominal Uang Kertas Rupiah</i>	157
<i>9.2.1. Blok Diagram</i>	158

9.2.2.	<i>Data Training</i>	158
9.2.3.	<i>Fitur Warna</i>	158
9.2.4.	<i>Preprocessing</i>	159
9.2.5.	<i>Proses Template Matching</i>	161
9.2.6.	<i>Hasil Percobaan</i>	161
9.3.	<i>Pengenalan Bunga indonesia</i>	163
9.3.1.	<i>HSV (Hue, Saturation, Value) Color Space</i>	164
9.3.2.	<i>CCD (Centroid Contour Distance)</i>	164
9.3.3.	<i>Sobel Edge Detection</i>	165
9.3.4.	<i>Pengumpulan Data</i>	165
9.3.5.	<i>Blok Diagram</i>	166
9.3.6.	<i>Ekstraksi Fitur</i>	168
9.3.7.	<i>Perhitungan Jarak Kemiripan Gambar</i>	172
9.3.8.	<i>Pengujian Pengukuran Kemiripan Gambar</i>	174
Daftar Pustaka		176

DAFTAR GAMBAR

Gambar 1.1. Contoh Citra Analog dan Citra Digital	4
Gambar 1.2. Proses Pengolahan Citra	4
Gambar 1.3. Kamera dan Frame-Grabber	5
Gambar 1.4. Gamma Ray Imaging	6
Gambar 1.5. X-Ray Imaging	6
Gambar 1.6. Image Ultraviolet	7
Gambar 1.7. Image Infrared	7
Gambar 1.8. Citra Satelit LANDSAT	8
Gambar 1.9. Multispectral Badai Andrew	9
Gambar 1.10. Visual Spektrum	9
Gambar 1.11. Visual Spektrum	10
Gambar 1.12. Citra dengan Band Gelombang Mikro	10
Gambar 1.13. Citra dengan Band Radio	11
Gambar 1.14. Penampang Gambaran Sebuah Model Seismik	11
Gambar 1.15. Image Ultrasound	12
Gambar 1.16. Citra yang Dibangkitkan oleh Komputer	12
Gambar 1.17. Citra sebagai Matrik	13
Gambar 1.18. Representasi Kurva $f(x,y)$ dari Citra	13
Gambar 1.19. Tiga Layer Layer RGB pada Citra Berwarna	14
Gambar 1.20. Contoh Teknik Pengolahan Citra Low Level	15
Gambar 1.21. Contoh Teknik Pengolahan Citra Mid Level	15
Gambar 1.22. Proses Dasar Pengolahan Citra	16
Gambar 1.23. Contoh Citra Hasil Enhancement	17
Gambar 1.24. Contoh Citra Hasil Restorasi	17
Gambar 1.25. Citra Asal dan Citra Hasil Proses Morfologi	17
Gambar 1.26. Contoh Citra Hasil Segmentasi	18
Gambar 2.1. Uji Citra	22
Gambar 2.2. Penampang Mata Manusia	24
Gambar 2.3. Cara Kerja Mata Manusia	25
Gambar 2.4. Adaptasi dan Diskriminasi Brightness	25
Gambar 2.5. Sensitifitas Kontras	26
Gambar 2.6. Brightness Discrimination dengan Rasio Weber	26
Gambar 2.7. Pola Intensitas Brightness	27
Gambar 2.8. Brightnes yang Diterima dengan Pencahayaan Aktual	27
Gambar 2.9. Ilustrasi Pola Mach Band	28
Gambar 2.10. Contoh Simultan Kontras	28
Gambar 2.11. Proses Sampling	30
Gambar 2.12. Proses Kuantisasi	30
Gambar 2.13. Citra Digital	30
Gambar 2.14. Representasi Citra Digital	31
Gambar 2.15. (a) Citra kontinu diproyeksikan ke sensor array	31
Gambar 2.15. (b) Citra hasil sampling dan kuantisasi	31
Gambar 2.16. Proses Resolusi Spatial	33
Gambar 2.17. Kwantisasi Hitam Putih	34
Gambar 2.18. Kwantisasi Warna	34
Gambar 2.19. Efek Checkerboard	35
Gambar 2.19. Efek False Counturing	35
Gambar 2.20. Contoh Sampling Non Uniform	36
Gambar 2.21. Contoh Konektivitas Piksel dengan N_8 dan N_m	38
Gambar 2.22. Contoh Subset S1 dan S2	39
Gambar 2.23. Cara Kerja Operasi Masking	43
Gambar 3.1. Domain Spatial	46
Gambar 3.2. Mask/Filter	47

Gambar 3.3. Kontras Streching	48
Gambar 3.4. a) Citra asal b) Citra Hasil Kontras Stretching.....	48
Gambar 3.5. Fungsi $s=T(r)$ dengan Thresholding m	49
Gambar 3.6. Contoh Citra Biner dengan Thresholding	49
Gambar 3.7. Contoh Thresholding Pada Kuantisasi Citra	50
Gambar 3.8. Histogram Thresholding	51
Gambar 3.9. Grafik Perubahan Tingkat Keabuan pada Transformasi Citra	51
Gambar 3.10. (a) Citra Mamogram Asal (b) Citra Negatif.....	52
Gambar 3.11. Contoh Citra Hasil Transformasi Logaritma	53
Gambar 3.12. Kurva Mapping Transformasi Power Law ($\gamma = 0.01$ s/d $\gamma = 25.$)	54
Gambar 3.13. Koreksi Gamma dan Hasil CRT di Monitor	54
Gambar 3.14. Citra Transformasi Power Law Pada MRI.....	55
Gambar 3.15. Citra Transformasi Power Law dengan Efek Penurunan Gamma	56
Gambar 3.16. Gray Level Slicing	57
Gambar 3.17. Bit Plane Slicing	57
Gambar 3.18. Citra Fraktal 8 Bit	58
Gambar 3.19. 8 Bit Planes	58
Gambar 3.20. Histogram Citra	59
Gambar 3.21. Contoh histogram.....	60
Gambar 3.22. Histogram Kumulatif	61
Gambar 3.23. Histogram ekualisasi.....	62
Gambar 3.24. Citra (Asal dan Hist. Ekualisasi) dan Histogram (Asal dan Ekualisasi)	62
Gambar 3.25. Citra Asal dan Histogram Citra Asal	63
Gambar 3.26. Citra Output dan Histogram Ekualisasi	64
Gambar 3.27. Perbandingan Citra Output dan Histogram Ekualisasi.....	66
Gambar 3.28. Contoh Perbandingan Citra Hasil Peningkatan Mutu (Global dan Lokal)	66
Gambar 3.29. Contoh Operasi AND.....	67
Gambar 3.30. Contoh Operasi OR.....	67
Gambar 3.31. Contoh Pengurangan Citra Fraktal.....	68
Gambar 3.32. Contoh Pengurangan Citra dengan Mask Radiografi.....	69
Gambar 3.33. Contoh Averaging Citra.....	70
Gambar 3.34. Citra Input Gray Level 4x4 dan Fungsi Transformasi	70
Gambar 3.35. Citra Input Gray Level 8x8 dan Matrik Citra Gray Levelnya.....	71
Gambar 4.1. Transformasi Fourier	72
Gambar 4.2. Contoh hasil transformasi fourier	73
Gambar 4.3. Contoh hasil transformasi fourier 2D.....	75
Gambar 4.4. Hasil transformasi fourier dalam surface	75
Gambar 4.5. Contoh DFT real dan imaginer	78
Gambar 4.6. Contoh DFT real dan imaginer	78
Gambar 4.7. Contoh citra dalam $f(x,y)$	80
Gambar 4.8. Contoh hasil DFT 2D.....	80
Gambar 4.9. Contoh hasil DFT 2D dalam magnitude dan phase	81
Gambar 4.10. Gambar fungsi cosinus 1 periode.....	81
Gambar 4.11. Gambar fungsi sinus 1 periode	82
Gambar 4.12. Format koordinat frekwensi	85
Gambar 4.13. Contoh transformasi fourier	85
Gambar 4.14. Contoh hasil DCT	87
Gambar 4.15. Contoh citra dalam $f(x,y)$	88
Gambar 4.16. Contoh hasil DCT 2D	88
Gambar 4.17. Contoh gambar untuk DFT dan DCT	90
Gambar 5.1. Format koordinat frekwensi pada citra	92
Gambar 5.2. Contoh transformasi fourier citra bergradiasi tinggi	92
Gambar 5.3. Contoh transformasi fourier citra bergradiasi rendah	93
Gambar 5.4. transformasi fourier untuk fungsi positif integer	94
Gambar 5.6. Transformasi fourier untuk fungsi simetri nol	95
Gambar 5.6. Perhitungan konvolusi secara grafis	97
Gambar 5.7. Macam-macam noise (a) gaussian (b) speckle dan (c) salt & pepper.....	100
Gambar 5.8 Model Fungsi Gaussian dalam ruang.....	106
Gambar 5.9. Model contour dari filter gaussian.	106

Gambar 5.10. Proses Deteksi Tepi	107
Gambar 5.11. Hasil beberapa deteksi tepi	108
Gambar 6.1 Spektrum Gelombang Elektromagnetik	114
Gambar 6.2 Elemen Warna Dasar RGB	116
Gambar 6.3 RGB Cube.....	116
Gambar 6.4 Elemen Dasar CMY(K)	117
Gambar 6.5 CMY(K) Color Wheel	117
Gambar 6.6 Koordinat Polar HSV	118
Gambar 6.7 CIE Coor Space Chromaticity Diagram.....	119
Gambar 6.8 Lab - Photoshop.....	119
Gambar 6.9 Color Gamut	120
Gambar 7.1 Deteksi Kematangan Buah Tomat	126
Gambar 7.2 Histogram RGB	127
Gambar 7.3 Perbedaan Histogram RGB dari Beberapa Gambar.....	128
Gambar 7.4 Histogram Indeks RGB	128
Gambar 7.5 Perbedaan Histogram Indeks RGB Pada Beberapa Gambar	129
Gambar 7.6 Referensi warna paint/36 warna.....	130
Gambar 7.7 Histogram Referensi 12 Warna.....	130
Gambar 7.8 Perbedaan Histogram Referensi dari Beberapa Gambar	130
Gambar 7.10 Bentuk Dasar	131
Gambar 7.11 Mendapatkan Fitur Object Persegi dengan Deteksi Tepi.....	132
Gambar 7.12 Mendapatkan Fitur Object Segi Tiga dengan Deteksi Tepi	132
Gambar 7.13 Histogram proyeksi object persegi.....	133
Gambar 8.1 Contoh Structuring Element	136
Gambar 8.2 Proses dilasi objek S dan A menjadi D	137
Gambar 8.3 Proses Erosi S terhadap A menjadi E	138
Gambar 8.4 Capture Proses Erosi	139
Gambar 8.5 Contoh Erosi	140
Gambar 8.6 Proses Opening	140
Gambar 8.7 Contoh Opening.....	141
Gambar 8.8. Contoh Proses Closing	142
Gambar 8.9 Perbedaan hasil proses opening dan closing	142
Gambar 8.10 Hit-orMiss Transform	143
Gambar 8.11 Contoh Thinning	145
Gambar 8.12 Proses Thinning	146
Gambar 9.1. Data gambar referensi untuk buah tomat mentah (hijau)	150
Gambar 9.2. Data gambar referensi untuk buah tomat setengah matang/campur (jingga)	150
Gambar 9.3. Data gambar referensi untuk buah tomat matang (merah)	151
Gambar 9.4. Blok diagram sistem pengenalan kematangan buah tomat	151
Gambar 9.5. Data test	152
Gambar 9.6. Proses mendapatkan histogram dari hue-index.....	152
Gambar 9.7. Histogram hue index dari tiga buah tomat dengan kematangan berbeda.....	153
Gambar 9.8. Fitur umum	154
Gambar 9.9. Hasil fitur khusus	155
Gambar 9.10. Hasil percobaan	156
Gambar 9.11. Salah satu hasil percobaan dengan data uji yang berbeda.....	156
Gambar 9.12. Contoh uang kertas rupiah	157
Gambar 9.13. Blok diagram aplikasi mengenali nominal uang kertas	158
Gambar 9.14. Histogram dari semua data training	159
Gambar 9.15. Hasil dari auto level	160
Gambar 9.15. Perubahan nilai histogram sebelum dan sesudah auto level	161
Gambar 9.16. Hasil percobaan tanpa autolevel	162
Gambar 9.17. Hasil percobaan dengan auto level	163
Gambar 9.18 Model warna HSV	164
Gambar 9.19. Pendekatan Centroid-Contour Distance.....	165
Gambar 9.20 Diagram alir aplikasi.....	168
Gambar 9.21. Diagram Blok Ekstraksi Fitur Warna.....	169
Gambar 9.22. Diagram Blok Ekstraksi Fitur Bentuk.....	170
Gambar 9.23. Hasil deteksi tepi metode sobel.....	170

Gambar 9.24. Ilustrasi penerapan rumus CCD pada gambar bunga	171
Gambar 9.25 Hasil pencarian 5 gambar peringkat atas	174

DAFTAR TABEL

Tabel 1.1. Bidang Komputer Dilihat dari Input dan Outputnya	3
Tabel 1.2. Band Tematik pada Visual dan Inframerah	8
Tabel 3.1. Proses Perhitungan Distribusi Kumulatif	60
Tabel 3.2. Histogram Ekualisasi	61
Tabel 3.3. Proses Histogram Ekualisasi	64
Tabel 9.1. Contoh data gambar bunga di Indonesia.....	166
Tabel 9.2. Contoh perhitungan CCD	171
Tabel 9.3. Hasil pengujian pengukuran kemiripan gambar	175

BAB 1

PENGANTAR PENGOLAHAN CITRA

POKOK BAHASAN:

- ✓ Pengantar Pengolahan Citra
- ✓ Dasar-Dasar Pengolahan Citra Digital
- ✓ Teknik-Teknik Pengambilan Citra Digital
- ✓ Model Citra Digital
- ✓ Teknik-Teknik Pengolahan Citra Digital

TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat:

- ✓ Menjelaskan dasar-dasar pengolahan citra digital
- ✓ Menjelaskan macam-macam teknik-teknik pengambilan citra digital
- ✓ Menyebutkan contoh-contoh manfaat pengolahan citra digital
- ✓ Menyebutkan macam-macam model citra digital
- ✓ Menjelaskan metodologi pengolahan citra digital

1.1. Pengantar Pengolahan Citra

Pengolahan citra (image processing) merupakan suatu sistem dimana proses dilakukan dengan masukan berupa citra (image) dan hasilnya juga berupa citra (image). Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputasi yang memungkinkan manusia dapat mengambil informasi dari suatu citra maka image processing tidak dapat dilepaskan dengan bidang computer vision.

Sesuai dengan perkembangan computer vision itu sendiri, pengolahan citra mempunyai dua tujuan utama, yaitu:

- 1) Memperbaiki kualitas citra, dimana citra yang dihasilkan dapat menampilkan informasi secara jelas atau dengan kata lain manusia dapat melihat informasi yang diharapkan dengan menginterpretasikan citra yang ada. Dalam hal ini interpretasi terhadap informasi yang ada tetap dilakukan oleh manusia (human perception).

- 2) Mengekstraksi informasi ciri yang menonjol pada suatu citra, dimana hasilnya adalah informasi citra dimana manusia mendapatkan informasi ciri dari citra secara numerik atau dengan kata lain komputer (mesin) melakukan interpretasi terhadap informasi yang ada pada citra melalui besaran-besaran data yang dapat dibedakan secara jelas (besaran-besaran ini berupa besaran numerik).

Dalam perkembangan lebih lanjut image processing dan computer vision digunakan sebagai pengganti mata manusia, dengan perangkat input image capture seperti kamera dan scanner dijadikan sebagai mata dan mesin komputer (dengan program komputasinya) dijadikan sebagai otak yang mengolah informasi. Sehingga muncul beberapa pecahan bidang yang menjadi penting dalam computer vision antara lain: pattern recognition (pengenalan pola), biometric (pengenalan identifikasi manusia berdasarkan ciri-ciri biologis yang tampak pada badan manusia), content based image and video retrieval (mendapatkan kembali citra atau video dengan informasi tertentu), video editing dan lain-lain.

Salah satu bidang dengan menggunakan pengolahan citra yang saat ini banyak dikembangkan orang adalah Biometric, yaitu bidang yang mempelajari bagaimana dapat mengidentifikasi seseorang dengan ciri yang unik yang ada dalam tubuh manusia. Salah satunya adalah identifikasi sidik jari, yang merupakan ciri unik yang dapat membedakan orang yang satu dengan yang lainnya. Untuk melakukan identifikasi sidik jari diperlukan pengolahan citra untuk melakukan capture (penangkapan citra sidik jari), sampai pada ekstraksi ciri yaitu mengekstrak besaran-besaran numerik yang dapat dijadikan suatu ciri sidik jari seperti core (pusat sidik jari) dan minusi (percabangan yang ada pada sidik jari), yang pada akhirnya dilakukan proses pembelajaran agar komputer dapat secara tepat mengidentifikasi sidik jari.

Dalam perkembangan lebih lanjut dari ilmu komputasi yang memanfaatkan pengolahan citra, ternyata untuk meng-identifikasi seseorang tidak hanya dengan sidik jari, tetapi dapat juga dilakukan dengan pengenalan wajah (face recognition) atau pengenalan iris pada mata (iris recognition). Dalam model pengenalan wajah dan pengenalan iris, proses pengolahan citra yang dilakukan menjadi tidak sederhana, baik dari sisi capture atau pengambilan citra, sampai pada ekstraksi cirinya. Pada pengenalan wajah proses capture ini sangat menentukan tingkat kesulitan dalam komputasinya, salah satunya bahwa dalam setiap proses capture ternyata cahaya, warna, posisi, skala dan kemiringan menjadi suatu masalah yang perlu diperhatikan.

Hubungan image processing dengan pembagian bidang dalam komputer yang melibatkan input dan output tertentu dapat dijelaskan dengan tabel 1.1 berikut ini.

Tabel 1.1. Bidang Komputer Dilihat dari Input dan Outputnya

		Output	
Input	Image	Deskripsi	
	Image	Image Processing	Pattern Recognition, Computer Vision
	Deskripsi	Computer Graphics	Data Processing lainnya

Dalam tabel 1.1. di atas terlihat bahwa pengolahan citra (image processing) merupakan suatu bidang pengetahuan dimana inputnya berupa citra dan hasilnya juga berupa citra dengan proses yang berupa perbaikan baik kualitas citra atau penyajian informasi citra. Agar hasilnya berupa data numerik atau teks yang menyatakan informasi yang ada dalam citra diperlukan pengetahuan yang dipelajari dalam pattern recognition dan computer vision.

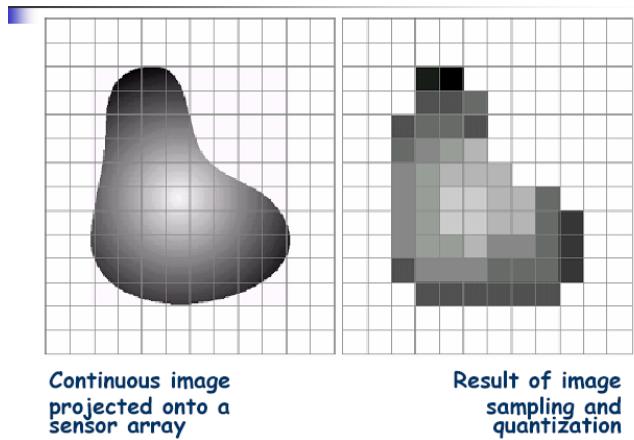
Beberapa contoh manfaat pengolahan citra :

1. Pengolahan citra dalam Bidang Kedokteran
 - memperjelas hasil x-ray organ tubuh manusia
2. Pengolahan citra dalam bidang teknologi dan komunikasi
 - membersihkan noise data gambar dari satelit
3. Pengolahan citra dalam dunia game
 - menciptakan efek-efek bayangan di atas permukaan air, tampilan yang kabur, transparansi, pencahayaan dll
4. Pengolahan citra dalam fotografi dan perfilman
 - gambar baru yang muncul perlahan-lahan di atas gambar yang lama, menghaluskan gambar, membuat gambar menjadi kasar, menerangkan gambar, menggelapkan gambar dll.

1.2. Dasar-Dasar Pengolahan Citra

Ada beberapa hal yang penting di dalam pengolahan citra digital antara lain: teknik-teknik pengambilan citra, model citra digital, sampling dan kuantisasi, threshold, histogram, proses filtering, perbaikan citra sampai pada pengolahan citra digital yang lebih lanjut seperti segmentasi, image clustering dan ekstraksi ciri.

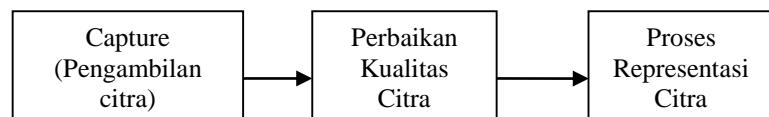
Ada perbedaan yang sangat mendasar mengenai citra digital dan citra analog, seperti yang terlihat seperti gambar 1.1. berikut ini.



Gambar 1.1. Contoh Citra Analog dan Citra Digital

Citra digital merupakan representatif dari citra yang diambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom atau dengan kata lain sampling pada citra menyatakan besar kecilnya ukuran piksel (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahan yang dinyatakan dalam nilai tingkat keabuan (gray scale) sesuai dengan jumlah bit biner yang digunakan oleh mesin dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra.

Proses pengolahan citra secara diagram proses dimulai dari pengambilan citra, perbaikan kualitas citra, sampai dengan pernyataan representatif citra dicitrakan dengan gambar 1.2. berikut ini.

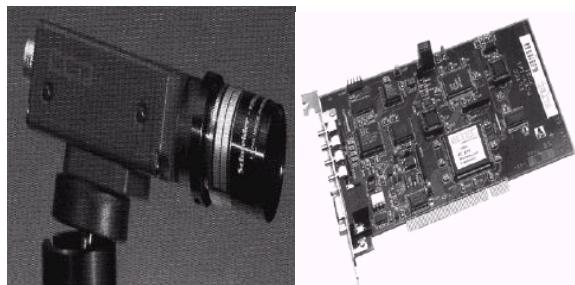


Gambar 1.2. Proses Pengolahan Citra

1.2.1. Teknik-Teknik Pengambilan Citra Digital

Ada beberapa teknik pengambilan digital yang bisa dilakukan antara lain dengan menggunakan kamera digital (atau boleh juga web-cam) atau menggunakan scanner (bisa scanner umum, finger-print scanner atau bar-code scanner). Hasil dari citra dengan menggunakan teknik kamera atau scanner berupa citra raster (atau citra dengan model matrik). Teknik pengambilan citra selain membutuhkan peralatan input, juga dibutuhkan

suatu card yang disebut dengan frame grabber yang berupa rangkaian untuk mengolah citra secara hardware.



Gambar 1.3. Kamera dan Frame-Grabber

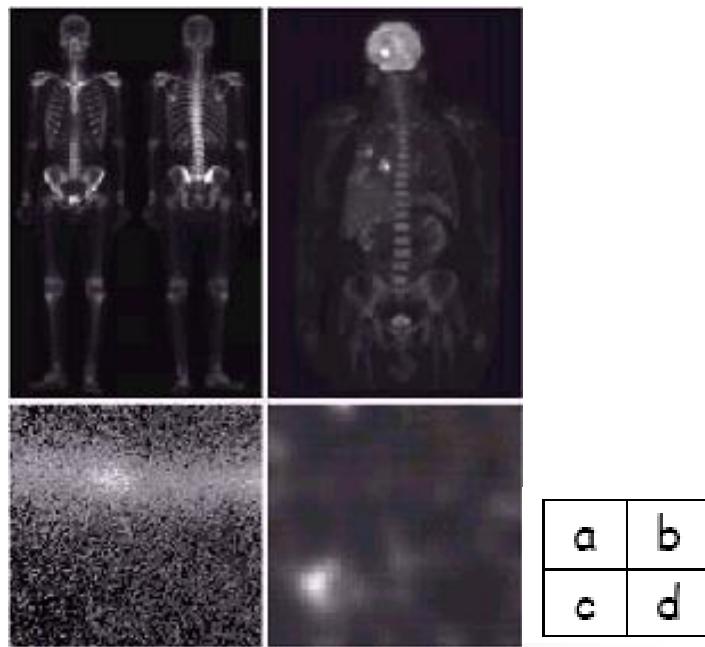
Selain itu pengambilan citra dapat dilakukan dengan menggunakan teknik grafika komputer yaitu dengan membentuk obyek citra komputer sesuai dengan teknik mengambil citra. Teknik ini banyak digunakan untuk keperluan entertainment seperti pembuatan animasi atau untuk keperluan GIS (Geographics Information System). Hasil dari citra dengan menggunakan teknik grafika ini bisa berupa citra raster atau bisa berupa citra vektor.

Baik citra raster atau citra vektor mempunyai kelebihan dan kekurangan masing-masing sehingga untuk memutuskan menggunakan model citra raster atau citra vektor diperlukan pengetahuan yang mendalam mengenai plan yang sedang dikerjakan. Dalam pembahasan pengolahan citra digital lebih banyak dibahas mengenai model citra raster, baru pada pengolah yang advance (pengolahan lebih lanjut) model citra vektor diperlukan.

Teknik pengambilan gambar akan membedakan proses citra yang akan digunakan di dalamnya. Misalkan scanner dan camera akan menghasilkan citra dalam format gambar tunggal, camera video dan webcam akan menghasilkan citra dalam format video. Demikian pula dengan resolusi dan format warna yang juga akan berbeda.

Berikut ini beberapa contoh citra selain gambar atau photo yang sering dilihat baik dengan scanner atau camera, yang dibedakan berdasarkan teknik pengambilan sinyalnya.

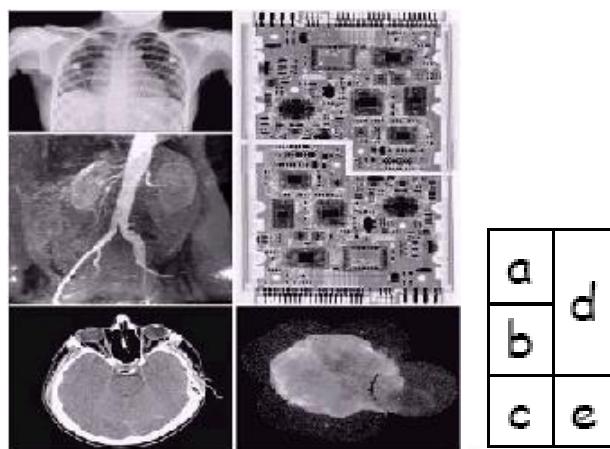
- 1) Gamma Ray Imaging, citra ini biasanya digunakan untuk citra-citra nuklir dan proses nuklir, juga digunakan untuk keperluan astronomi.



Gambar 1.4. Gamma Ray Imaging

- (a) Scan tulang, (b) Gambar PET (Positron emission tomography)
- (c) Cygnus Loop, (d) Radiasi Gamma dari reaktor katup

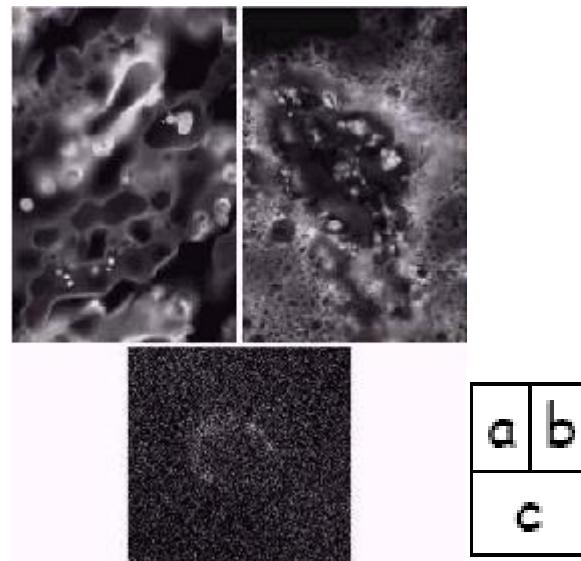
- 2) X-Ray Imaging, citra ini banyak ditemui untuk keperluan medis, rangkaian elektronik dan astronomi



Gambar 1.5. X-Ray Imaging

- (a) Sinar-X dada (familiar), (b) aorta angiogram, (c) CT Kepala
- (d) Circuit board, (e) Cygnus Loop

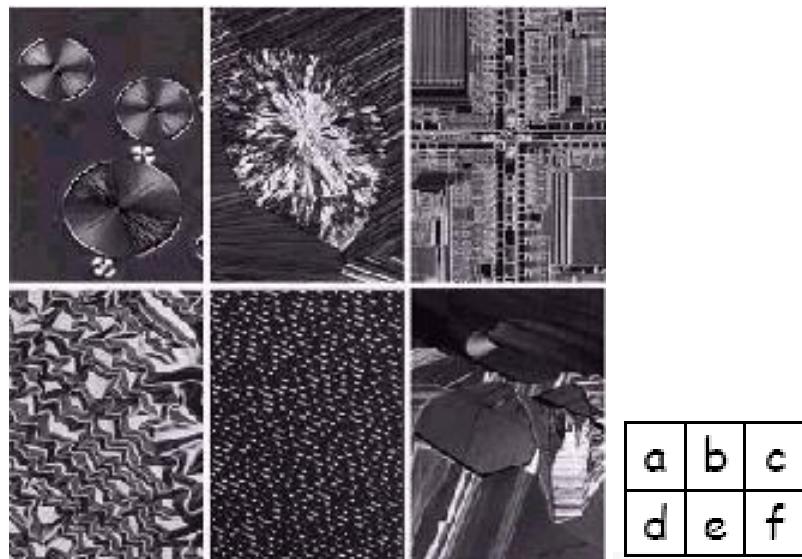
- 3) Image Ultraviolet, citra ini banyak ditemui untuk keperluan citra biologi, microscopy, lithography, laser dan astronomi



Gambar 1.6. Image Ultraviolet

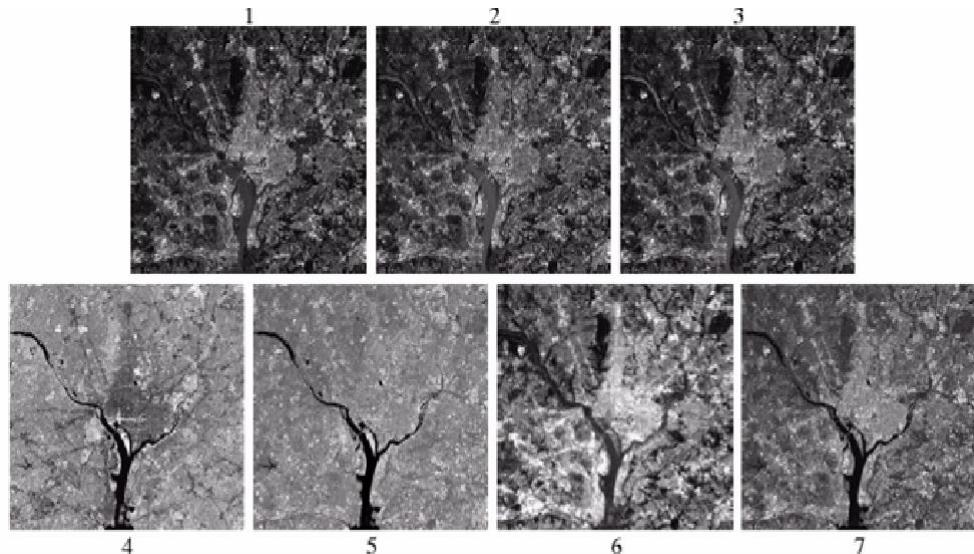
(a) Jagung Normal, (b) Jagung corengan, (c) Cygnus Loop

- 4) Image Infrared, citra ini banyak ditemui untuk keperluan farmasi, mikrobiotik, dan mrico material.



Gambar 1.7. Image Infrared

- (a). taxol (agen anti kanker), (b). kolesterol, (c). Mikroprosesor
 (d). Film tipis nickel oksida , (e). Permukaan CD audio, (f). Organic superconductor
- 5) Remote Sensing : foto pada band visual dan infra merah, contoh untuk memantau kondisi lingkungan pada planet : NASA's Landsat Washington DC



Gambar 1.8. Citra Satelit LANDSAT

Washington DC (angka merujuk pada tabel 1.2.)

Tabel 1.2. Band Tematik pada Visual dan Inframerah

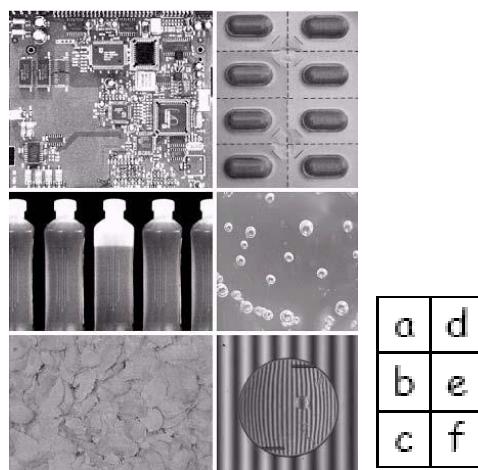
Band No	Name	Wavelength (μm)	Characteristics and Uses
1	Visible blue	0.45-0.52	Maximum water penetration
2	Visible green	0.52-0.60	Good for measuring plant vigor
3	Visible red	0.63-0.69	Vegetation discrimination
4	Near infrared	0.76-0.90	Biomass and shoreline mapping
5	Middle infrared	1.55-1.75	Moisture content of soil and vegetation
6	Thermal infrared	10.4-12.5	Soil moisture, thermal mapping
7	Middle infrared	2.08-2.35	Mineral mapping

Remote Sensing : pengamatan dan prediksi cuaca. Dengan sensor visual dan inframerah.



Gambar 1.9. Multispectral Badai Andrew

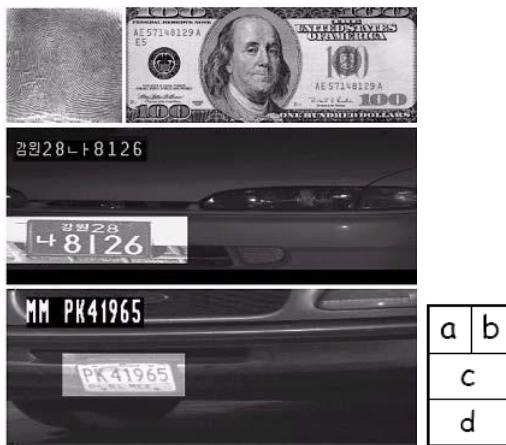
6) Visual Spektrum



Gambar 1.10. Visual Spektrum

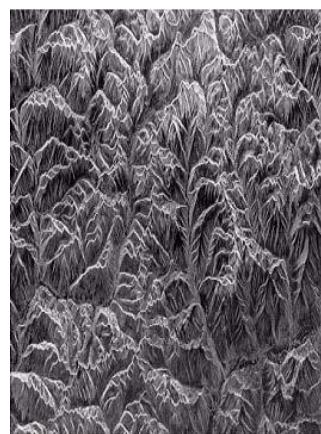
(inspeksi visual otomatis dari barang-barang manufaktur)

- (a). Circuit board : memeriksa bagian-bagian yang hilang
- (b). Tempat pil : mencari pil-pil yang tidak ada
- (c). Botol: mencari botol yang tidak terisi penuh pada level yang dapat diterima
- (d). Gelembung di dalam produk plastik: mendeteksi jumlah kantong udara yang tidak dapat diterima
- (e). Sereal: inspeksi untuk warna dan adanya anomali seperti serpihan yang dapat terbakar.
- (f). Image pengganti lensa pada mata manusia: inspeksi terhadap kerusakan atau produksi implant yang tidak tepat.



Gambar 1.11. Visual Spektrum

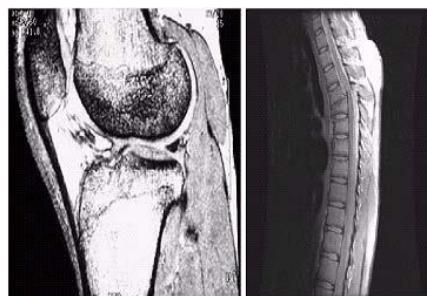
- (a). Cap jempol: otomatis mencari pada database yang sesuai
- (b). Uang kertas : otomatis menghitung / membaca nomor seri untuk melacak dan mengidentifikasi tagihan (c) dan (d) Otomatis membaca nomor plat nomer
- 7) Band Gelombang Mikro



Gambar 1.12. Citra dengan Band Gelombang Mikro

- Pencitraan radar : satu-satunya cara untuk menjelajahi daerah yang tidak dapat diakses dari permukaan bumi
 - Citra radar pegunungan di tenggara Tibet
- Catatan kejelasan dan detail gambar, tidak terbebani oleh awan atau kondisi lain atmosfer yang biasanya mengganggu dengan gambar dalam band visual.

8) Band Radio



Gambar 1.13. Citra dengan Band Radio

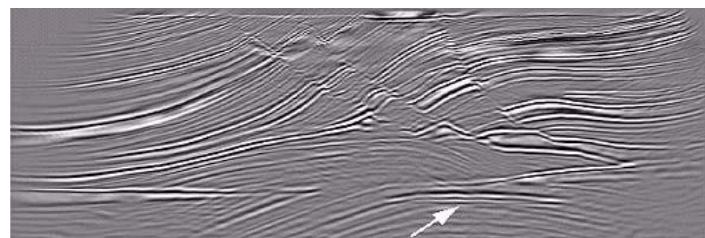
- Obat

Magnetic resonance image (MRI) : gambar 2D bagian pasien(semua bidang)

- (a) lutut
- (b) tulang punggung

- Astronomi

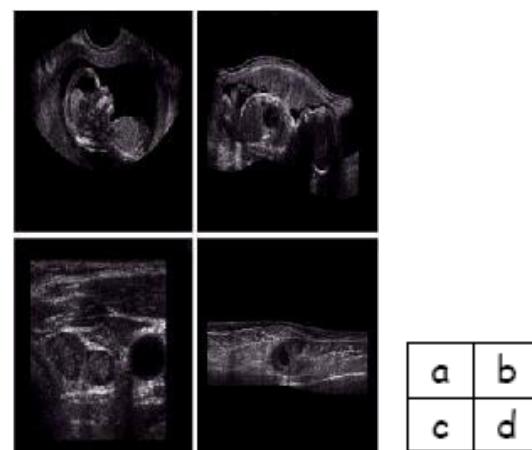
9) Akustik : Aplikasi Geologi : menggunakan suara di titik rendah spektrum suara(seratus Hz), Mineral dan eksplorasi minyak



Gambar 1.14. Penampang Gambaran Sebuah Model Seismik

Panah menunjuk ke sebuah hidrokarbon (minyak dan / atau gas) perangkap (titik cerah)

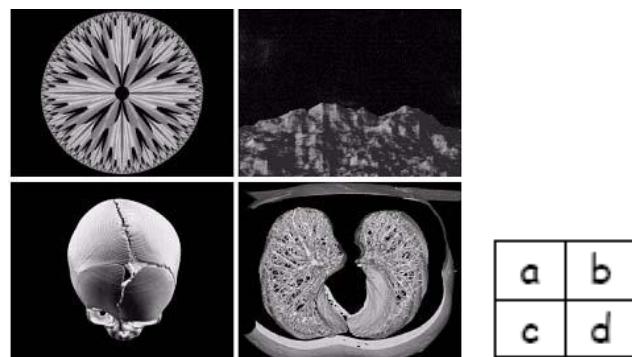
10) Citra Ultrasound, citra ini banyak ditemui untuk keperluan medis seperti USG, diagnosa thyroid.



Gambar 1.15. Image Ultrasound

- (a) Bayi, (b) Bayi dengan sisi lain, (c) Thyroid, (d) Menampilkan lapisan otot lesi

11) Citra yang dibangkitkan oleh komputer



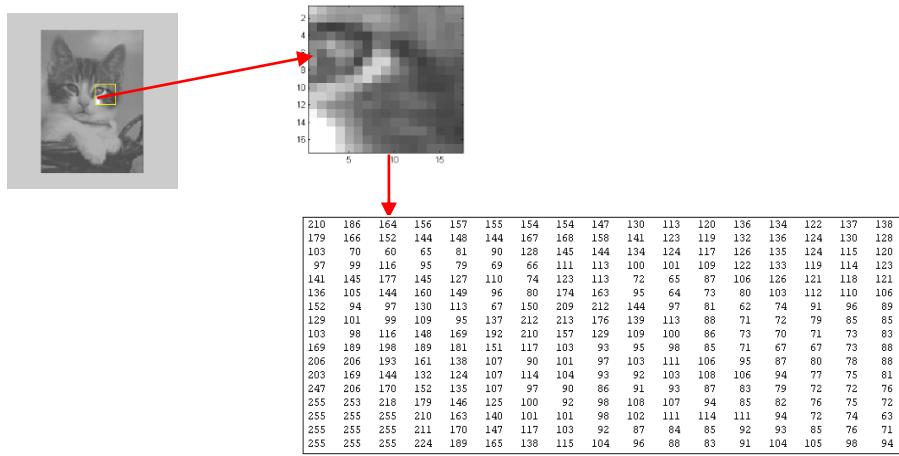
Gambar 1.16. Citra yang Dibangkitkan oleh Komputer

- (a) dan (b) Fraktal : reproduksi berulang-ulang pada pola dasar mengikuti beberapa aturan matematika
(c) dan (d) Pemodelan 3-D Komputer

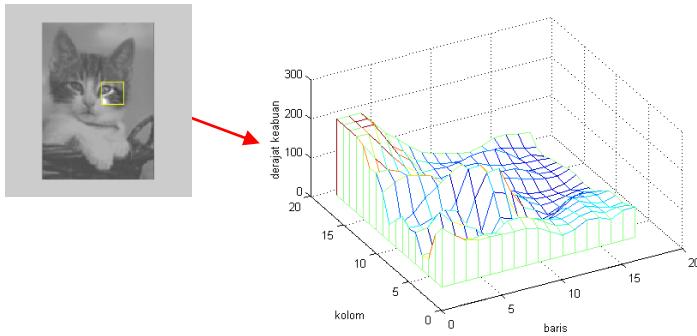
1.2.2. Model Citra Digital

Seperti telah dijelaskan, citra digital adalah citra yang diambil berdasarkan sampling dan kuantisasi tertentu sehingga citra digital ini terbentuk dari piksel-piksel yang besarnya tergantung pada besar kecilnya sampling dan nilainya (besarnya derajat keabuan) tergantung pada kuantisasi. Berdasarkan pengertian ini maka model citra digital dinyatakan dalam bentuk matrik yang nilainya berupa nilai derajat keabuan seperti terlihat pada citra 1.17. berikut. Model ini menyatakan model dari citra gray-scale yaitu citra yang terdiri dari derajat keabuan tertentu.

Dengan kata lain dapat dinyatakan bahwa citra digital adalah citra yang didefinisikan sebagai fungsi $f(x,y)$ dimana x menyatakan nomor baris, y menyatakan nilai kolom, dan f menyatakan nilai derajat keabuan dari citra. Sehingga (x,y) adalah posisi dari piksel dan f adalah nilai derajat keabuan pada titik (x,y) seperti terlihat pada gambar 1.18.

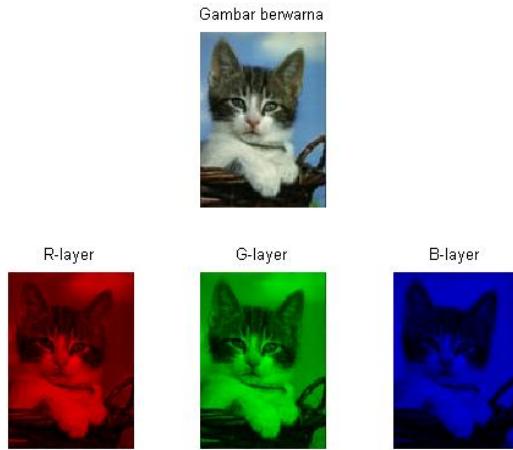


Gambar 1.17. Citra sebagai Matrik



Gambar 1.18. Representasi Kurva $f(x,y)$ dari Citra

Representasi model data di atas adalah representasi dari citra gray-scale (menggunakan derajat keabuan). Untuk citra berwarna, maka digunakan model RGB (Red-Green-Blue) dimana satu citra berwarna dinyatakan sebagai tiga buah matrik gray-scale yang berupa matrik untuk Red (R-layer), matrik untuk Green (G-layer) dan matrik untuk Blue (B-layer).



Gambar 1.19. Tiga Layer Layer RGB pada Citra Berwarna

R-layer adalah matrik yang menyatakan derajat kecerahan untuk warna merah (misalkan untuk skala keabuan 0-255, maka nilai 0 menyatakan gelap (hitam) dan 255 menyatakan warna merah. G-layer adalah matrik yang menyatakan derajat kecerahan untuk warna hijau, dan B-layer adalah matrik yang menyatakan derajat kecerahan untuk warna biru. Citra 1.5 menyatakan model RGB dari citra berwarna yang dipisah menjadi R-layer, G-layer dan B-layer.

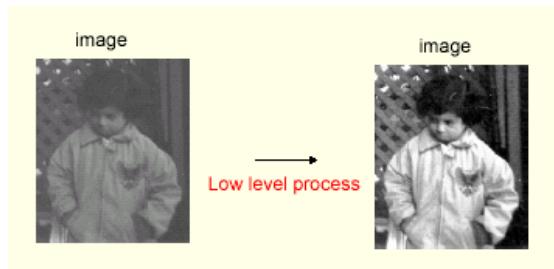
Model matrik pada citra digital memungkinkan dilakukannya operasional-operasional matrik, seperti aritmatika matrik (seperti penjumlahan, pengurangan dan perkalian) atau proses seperti invers dan transpose. Hal inilah yang menjadi dasar dari pengolahan citra. Pada bab-bab berikutnya proses pengolahan matrik ini menjadi acuan dalam mengolah citra.

1.2.3. Teknik-Teknik Pengolahan Citra Digital

Secara umum teknik pengolahan citra digital dibagi menjadi tiga tingkat pengolahan yaitu:

- Tahap 1 yang dinamakan dengan Low-Level Processing (pengolahan tingkat rendah), pengolahan ini operasional-operasional dasar dalam pengolahan citra, seperti pengurangan noise (noise reduction), perbaikan citra (image enhancement) dan restorasi citra (image restoration).

Contoh teknik pengolahan Low-Level Processing dapat diperhatikan gambar 1.20.

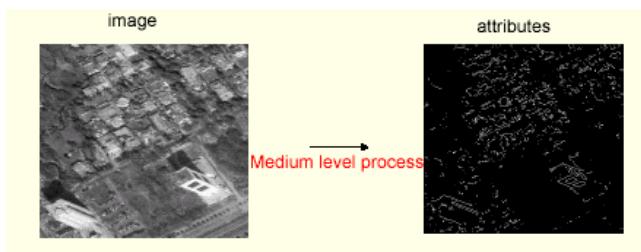


Gambar 1.20. Contoh Teknik Pengolahan Citra Low Level

Low proses citra diantaranya : mengurangi noise, menambah kontras, mempertajam citra

- Tahap 2 yang dinamakan dengan Mid-Level Processing (pengolahan tingkat menengah) yang meliputi segmentasi pada citra, deskripsi obyek dan klasifikasi obyek secara terpisah.

Contoh teknik pengolahan Low-Level Processing dapat diperhatikan gambar 1.21.



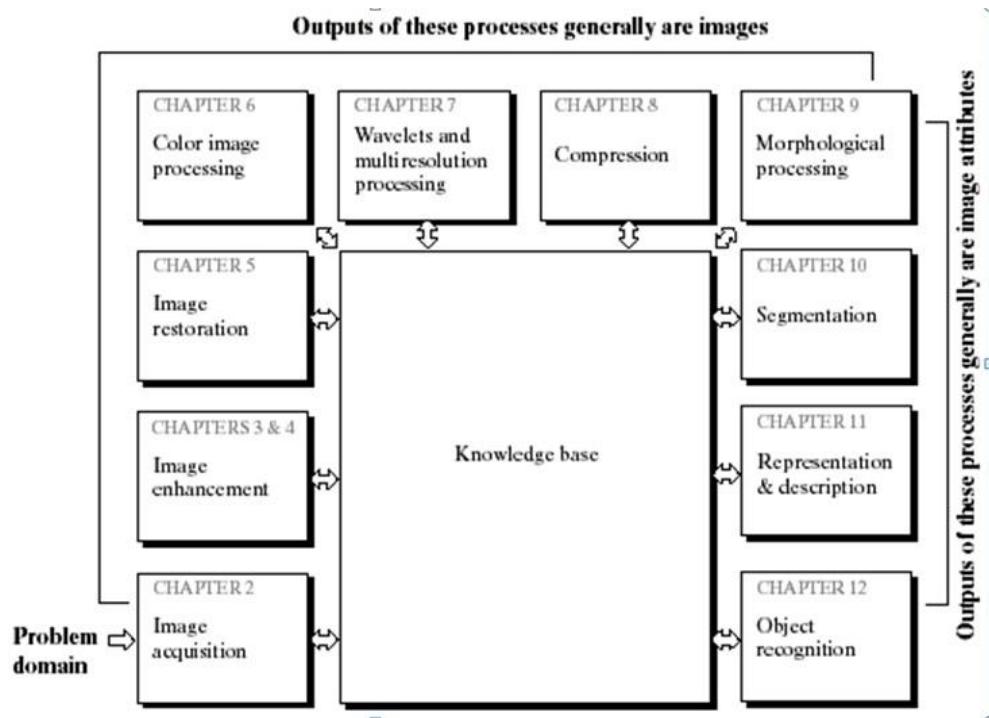
Gambar 1.21. Contoh Teknik Pengolahan Citra Mid Level

Mid level proses di pengolahan citra diantaranya untuk membagi citra dalam daerah atau obyek sesuai untuk proses komputer. Proses detailnya bagaimana mendapatkan attributes, tepian, kontur, dan mengenali obyek individu

- Tahap 3 yang dinamakan dengan High-Level Processing (pengolahan tingkat tinggi), yang meliputi analisa citra.

High level proses di pengolahan citra diantaranya bertujuan untuk agar komputer bisa merasakan, pengenalan obyek, analisa citra, dan kemampuan fungsi pengamatan normal digabung dengan vision

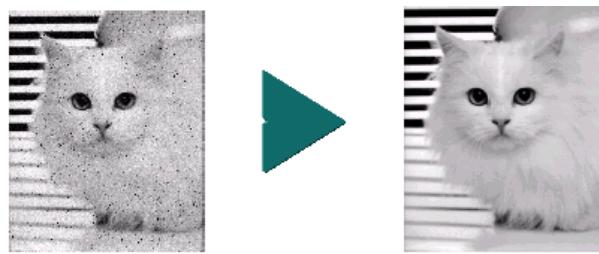
Dari ketiga tahap pengolahan citra digital di atas, dapat dinyatakan suatu gambaran mengenai teknik-teknik pengolahan citra digital dan macam-macamnya menurut buku Digital Image Processing, Gonzales ditunjukkan pada gambar 1.22.



Gambar 1.22. Proses Dasar Pengolahan Citra

Penjelasan dari beberapa proses dasar pengolahan citra sebagai berikut :

- 1) Pembentukan Citra (Image Acquisition) : Proses dimana sebuah gambar diambil dari sebuah sensor (seperti kamera TV monokrom atau warna) dan dijadikan digital. Jika output dari kamera atau sensor belum dalam bentuk digital, maka converter analog-to-digital akan mendigitalkannya. Kamera terdiri dari dua bagian : pertama sebuah lensa yang mengumpulkan pancaran radiasi dari objek yang diinginkan dan yang membentuk citra dari objek nyata, kedua peralatan semikonduktor atau *Charge Coupled Device* (CCD) yang mengkonversi radiasi dari bidang gambar menjadi sinyal listrik. Pada image acquisition juga terdapat frame grabber yang memerlukan jalan arus listrik untuk mendigitalkan sinyal listrik dari sensor pencitraan untuk menyimpan citra dalam memori komputer (RAM), gambar frame grabber.
- 2) Image Enhancement, berupa proses perbaikan citra dengan meningkatkan kualitas citra baik secara kontras maupun secara kecerahan.



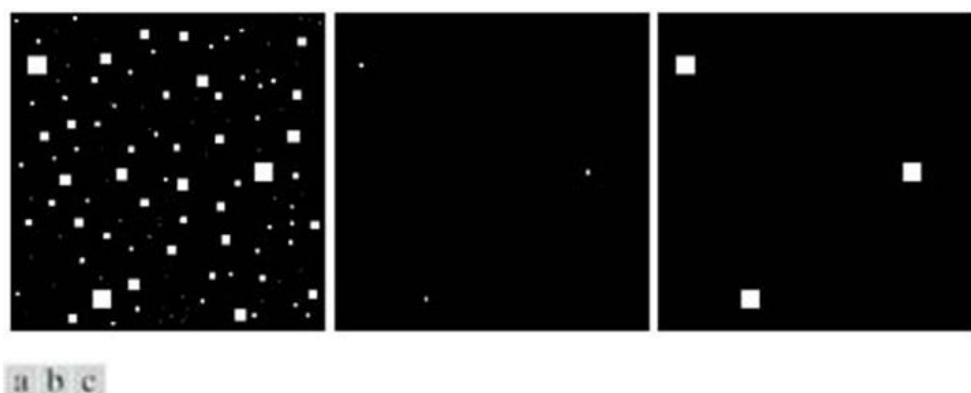
Gambar 1.23. Contoh Citra Hasil Enhancement

- 3) Image Restoration, proses memperbaiki model citra, biasanya berhubungan dengan bentuk citra yang sesuai.



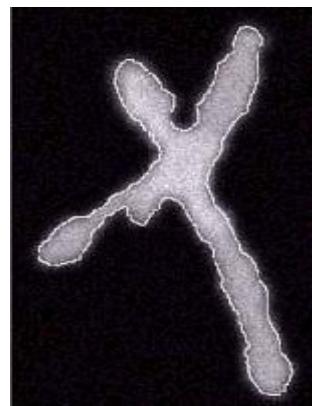
Gambar 1.24. Contoh Citra Hasil Restorasi

- 4) Color Image Processing, suatu proses yang melibatkan citra berwarna, baik berupa image enhancement, image restoration atau yang lainnya.
- 5) Wavelet dan Multiresolution Processing, merupakan suatu proses yang menyatakan citra dalam beberapa resolusi.
- 6) Image Compression, merupakan proses yang digunakan untuk mengubah ukuran data pada citra. Proses ini saat ini terus dikembangkan. JPEG-4 merupakan salah satu bentuk image compression yang baik dan saat ini banyak digunakan. Saat ini para peneliti sedang mencanangkan JPEG-7.
- 7) Morphological Processing, proses untuk memperoleh informasi yang menyatakan deskripsi dari suatu bentuk pada citra.



Gambar 1.25. Citra Asal dan Citra Hasil Proses Morfologi

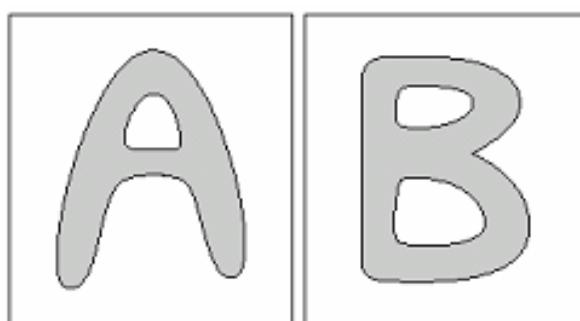
- a. Citra persegi ukuran 1,3,5,7,9,15, b. Citra hasil erosi dengan struktur elemen 1 dengan hasil yang tampak adalah citra persegi ukuran 13, c. Citra hasil dilasi pada citra b dengan struktur elemen sama
- 8) Segmentation, proses untuk membedakan atau memisahkan obyek-obyek yang ada dalam suatu citra, seperti memisahkan obyek dengan latar belakangnya.



Gambar 1.26. Contoh Citra Hasil Segmentasi

- 9) Object Recognition, suatu proses yang dilakukan untuk mengenali obyek-obyek apa saja yang ada dalam suatu citra.
- 10) Representasi adalah membuat keputusan apakah data harus direpresentasikan sebagai batas (*Boundary*) atau sebagai wilayah yang lengkap.
 - Representasi boundary adalah fokus pada karakteristik bentuk eksternal, seperti sudut dan infleksi.
 - Representasi region adalah fokus pada sifat-sifat internal, seperti tekstur atau bentuk kerangka.

Ilustrasi representasi dan deskripsi

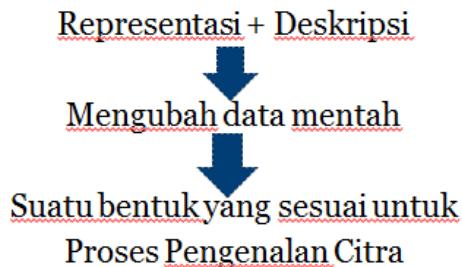


Gambar 1.27. Representasi dan Deskripsi Huruf A dan Huruf B

1 Komponen Terhubung dan 1 Lubang

Huruf B : 1 Komponen Terhubung dan 2 Lubang

Sedangkan Representasi dan Deskripsi hubungannya dengan mendapatkan bentuk yang cocok untuk proses pengenalan citra dapat dijelaskan berikut :



11) Recognition dan Interpretation

Recognition adalah proses yang memberikan label pada sebuah objek berdasarkan informasi yang diberikan oleh deskriptor. Interpretation bertugas mengartikan rangkaian objek – objek hasil recognition.

12) Basis Pengetahuan (Knowledbase)

Sebuah permasalahan utama mengetahui rincian daerah dari suatu gambar dimana interest informasi telah diketahui, maka basis pengetahuan membantu untuk membatasi pencarian.

Dalam buku ini, sebelas macam teknik pengolahan citra di atas tidak dapat dijelaskan semua, karena buku ini hanya menjelaskan operasional-operasional dasar dalam pengolahan citra. Masing-masing bagian dapat menjadi suatu topik bahasan yang panjang dan sangat menarik. Akan lebih menarik lagi bila pengolahan citra ini secara bersama-sama digunakan dengan soft-computing (komputasi cerdas), sehingga akan didapatkan suatu hasil yang saat ini kami (para peneliti bidang citra) harapkan yaitu bahwa “komputer dapat melihat seperti manusia”.

1.3. Ringkasan

1. Pengolahan citra (image processing) merupakan suatu sistem dimana proses dilakukan dengan masukan berupa citra (image) dan hasilnya juga berupa citra (image).
2. Pengolahan citra mempunyai dua tujuan utama, yaitu memperbaiki kualitas citra dan mengekstraksi informasi ciri yang menonjol pada suatu citra.
3. Citra digital adalah citra yang diambil berdasarkan sampling dan kuantisasi tertentu sehingga citra digital ini terbentuk dari piksel-piksel yang besarnya

tergantung pada besar kecilnya sampling dan nilainya (besarnya derajat keabuan) tergantung pada kuantisasi.

4. Secara umum teknik pengolahan citra digital dibagi menjadi 3 tingkat pengolahan yaitu:
 - a) Low-Level processing (pengolahan tingkat rendah), melakukan operasional dasar dalam pengolahan citra seperti pengurangan noise (noise reduction), perbaikan citra (image enhancement) dan restorasi citra (image restoration).
 - b) Mid-Level Processing (pengolahan tingkat menengah) yang meliputi segmentasi pada citra, deskripsi obyek dan klasifikasi obyek secara terpisah.
 - c) High-Level Processing (pengolahan tingkat tinggi), meliputi analisa citra.

1.4. Latihan

1. Sebutkan perbedaan input dan output dari proses-proses pengolahan citra digital, komputer grafik, pattern recognition dan data processing
2. Apa pengertian sampling dan kuantisasi dalam citra ?
3. Bila citra gary-scale dengan ukuran 320x240 piksel dan derajat keabuan 256, berapakah sampling dan kuantisasi pada citra tersebut ?
4. Apa pengertian dari capture pada pengolahan citra digital, dan bagaimana caranya?
5. Sebutkan alat-alat yang digunakan untuk melakukan capture dalam pengolahan citra digital, dan jelaskan cara kerjanya masing-masing !
6. Apa perbedaan citra raster dan citra vektor ? Jelaskan bagaimana anda dapat menentukan bentuk citra raster ataukah citra vektor yang lebih baik digunakan ?
7. Berdasarkan teknik pengambilannya, sebutkan jenis-jenis citra yang saudara kenal ?
8. Dalam citra berwarna terdapat layer RGB, bila masing-masing layer RGB tersebut bernilai 28, berapa bitkah warna yang diperlukan untuk satu piksel ?
9. Jelaskan tiga tahap dalam pengolahan citra, dan apa contoh dari masing-masing tahap tersebut ?
10. Apa perbedaan image enhancement dan image restoration ?

BAB 2

DASAR CITRA DIGITAL

POKOK BAHASAN:

- ✓ Sistem Penglihatan Manusia dan komputer
- ✓ Struktur Mata Manusia
- ✓ Adaptasi dan Diskriminasi Brightness
- ✓ Sensitivitas Kontras
- ✓ Rasio Weber
- ✓ Brightness Sebagai Fungsi Intensitas
- ✓ Pola Mach Band
- ✓ Simultan Kontras
- ✓ Sinyal
- ✓ Sinyal Waktu
- ✓ Sinyal Spatial
- ✓ Sinyal Spatialtemporal
- ✓ Analog Dan Digital
- ✓ Sampling
- ✓ Kuantisasi
- ✓ Representasi Citra Digital
- ✓ Fungsi Intensitas Cahaya
- ✓ Gray Level
- ✓ Jumlah Bit
- ✓ Resolusi
- ✓ Efek Checkerboard
- ✓ False Counturing
- ✓ Sampling Non Uniform
- ✓ Kuantisasi Non Uniform
- ✓ Relasi Dasar Piksel B/W
- ✓ Tetangga Piksel
- ✓ Pelabelan Komponen Terhubung
- ✓ Pengukuran Jarak
- ✓ Operasi Aritmatika Dan Logika

TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat:

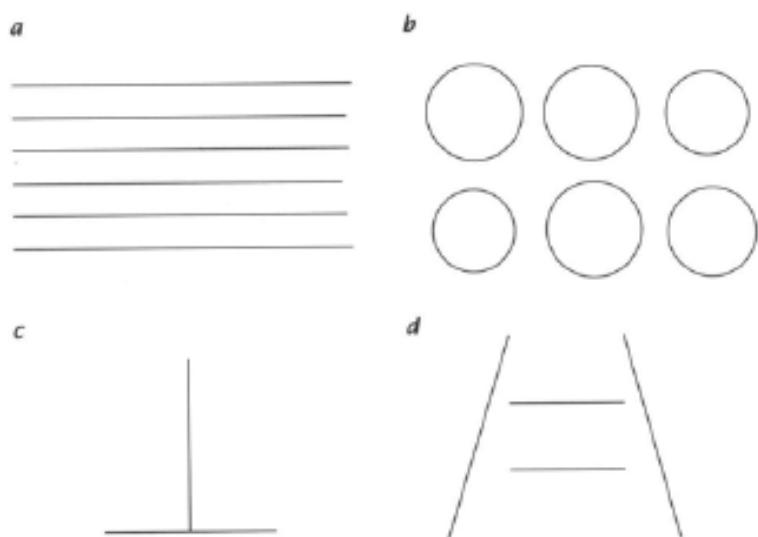
- ✓ Menjelaskan bagaimana pembentukan citra oleh mata
- ✓ Menjelaskan pengertian citra digital
- ✓ Menjelaskan resolusi sampling dan kuantisasi pada citra

- ✓ Menjelaskan pengertian piksel dan ketetanggaan
- ✓ Melakukan pemrosesan citra menggunakan relasi dasar piksel citra B/W
- ✓ Melakukan pemrosesan citra dengan mengukur jarak piksel dan tetangganya

2.1. Sistem Penglihatan Manusia Dan Komputer

Mempelajari pengolahan citra tidak bisa dilakukan tanpa memperhatikan sistem penglihatan manusia (human vision). Sejumlah uji coba dan evaluasi diberikan pada citra yang akan diproses oleh sistem visual yang kita miliki. Tanpa memperhatikan sistem human vision kita mungkin banyak salah dalam menginterpretasikan citra. Beberapa pertanyaan sederhana tentang human vision kita yang erat hubungannya dengan pengolahan citra :

- Perbedaan intensitas apa yang dapat kita bedakan ?
- Apa yang dimaksud dengan resolusi spasial mata kita?
- Bagaimana memperkirakan dan membandingkan jarak dan luasan secara akurat ?
- Bagaimana kita merasakan warna?
- Dengan fitur apa kita dapat mendekripsi dan membedakan obyek?



Gambar 2.1. Uji Citra

Hasil pengujian citra oleh human vision kita untuk memperkirakan jarak dan luasan/area pada gambar 2.1. sebagai berikut :

- a. Garis sejajar dengan perbedaan panjang sampai dengan 5%.
- b. Lingkaran dengan perbedaan radius hingga 10%.

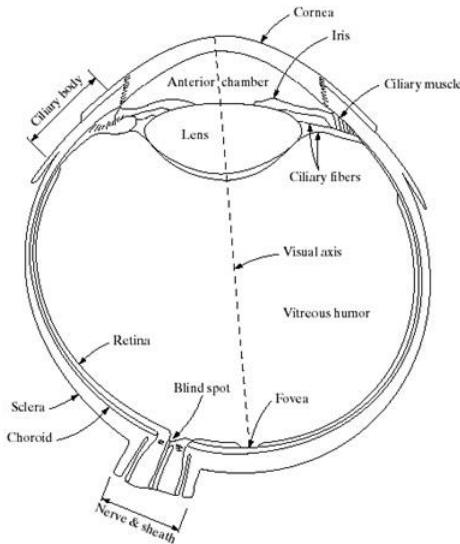
- c. Garis vertikal kelihatan lebih panjang daripada garis horisontal padahal sebenarnya memiliki panjang yang sama dengan garis horizontal.
- d. Tipuan perspektif : garis bagian atas kelihatan lebih panjang daripada garis di bawahnya padahal sebenarnya memiliki panjang yang sama.

2.1.1. Sistem Visual Manusia

Struktur mata manusia perhatikan gambar 2.2., adalah :

- 1. Bentuknya hampir menyerupai sebuah bola.
- 2. Rata-rata diameter 20 mm.
- 3. Mempunyai 3 membran :
 - o Cornea dan Sclera – bagian luar
 - o Choroid
 - o Retina – membungkus mata
- Cornea : membran bagian luar yang keras, jaringan transparan, meliputi permukaan mata bagian depan.
- Sclera : membrane bagian luar yang opaque, membungkus remainder dunia optik.
- Choroid : terletak di bawah sclera, jaringan berisi pembuluh darah yang berfungsi sebagai sumber utama gizi ke mata. Lapisan choroid memiliki pigmen sehingga sangat membantu untuk mengurangi jumlah cahaya ekstra memasuki mata dan kembali berpencar di dalam dunia optik.
- Lensa : baik inframerah dan sinar ultraviolet diserap oleh protein dalam struktur lensa. Dalam jumlah yang berlebihan, dapat menyebabkan kerusakan pada mata.
- Retina
 - o Membran mata terdalam yang berjajar dalam seluruh dinding bagian posterior.
 - o Pada saat mata benar-benar fokus, cahaya dari suatu obyek di luar mata tergambar pada retina.
- *Fovea* di bagian *retina* terdiri dari dua jenis *receptor*:
 - o Sejumlah *cone receptor*, sensitif terhadap warna, visi *cone* disebut *photocopic vision* atau *bright light vision*

- Sejumlah *rod receptor*, memberikan gambar keseluruhan pandangan dan sensitif terhadap iluminasi tingkat rendah, visi *rod* disebut *scotopic vision* atau *dim-light vision*
- *Blind Spot*
 - adalah bagian retina yang tidak mengandung *receptor* sehingga tidak dapat menerima dan menginterpretasi informasi.

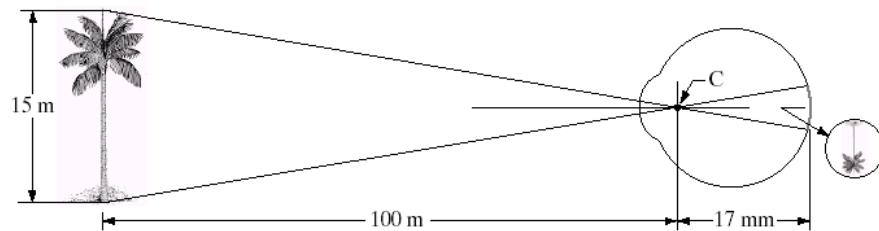


Gambar 2.2. Penampang Mata Manusia

Pembentukan citra oleh sensor mata, perhatikan gambar 2.2.yaitu :

1. Mata kita menangkap sebuah gambar saat ada cahaya yang tertangkap oleh mata.
2. Intensitas cahaya ditangkap oleh diagram iris dan diteruskan ke bagian retina mata.
3. Bayangan obyek pada retina mata dibentuk dengan mengikuti konsep sistem optik dimana fokus lensa terletak antara retina dan lensa mata.
4. Mata dan syaraf otak dapat menginterpretasi bayangan yang merupakan obyek pada posisi terbalik.

Gambar 2.3. merepresentasikan cara kerja mata manusia dalam menangkap sebuah gambar.



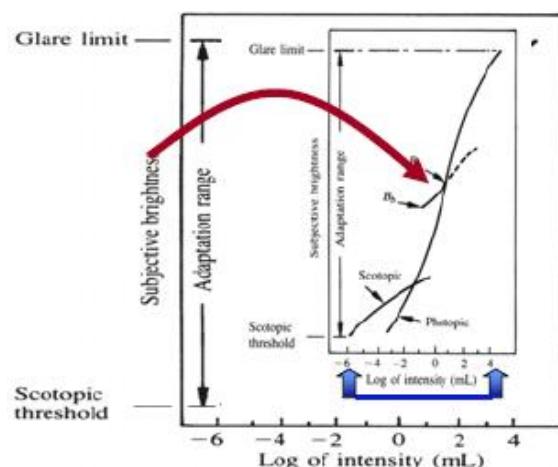
Gambar 2.3. Cara Kerja Mata Manusia

2.1.2. Adaptasi Dan Diskriminasi Brightness

Merupakan fenomena penyesuaian mata manusia dalam membedakan gradasi tingkat kecemerlangan ;

- Batas daerah tingkat kecemerlangan yang mampu dibedakan secara sekaligus oleh mata manusia lebih kecil dibandingkan dengan daerah tingkat kecemerlangan sebenarnya.
- Contoh tentang adaptasi brightness pada mata manusia adalah perubahan pupil ketika dari ruang gelap ke ruang terang atau sebaliknya. Pada kamera, perubahan titik fokus saat pengambilan gambar, jika titik fokusnya gelap maka gambar akan menjadi lebih terang dan sebaliknya.

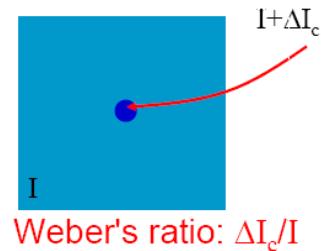
Sehubungan dengan adaptasi dan diskriminasi brightness diilustrasikan pada gambar 2.4.



Gambar 2.4. Adaptasi dan Diskriminasi Brightness

2.1.3. Sensitivitas Kontras

Kemampuan mata untuk membedakan antar perubahan dalam kecerahan pada setiap tingkat adaptasi spesifik yang cukup menarik. I adalah pencahayaan merata pada daerah yang datar yang cukup besar untuk memenuhi seluruh bidang pandang. ΔI_c adalah perubahan kecerahan objek yang hanya diperlukan untuk membedakan obyek dari background. Sensitivitas kontras kemudian dihubungkan dengan gambar 2.6.

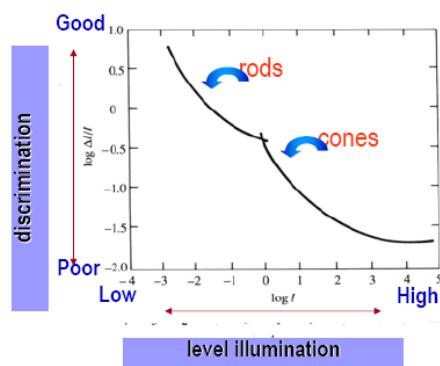


Gambar 2.5. Sensitifitas Kontras

Brightness discrimination bagus maka $\Delta I_c/I$ kecil, kebalikannya brightness discrimination jelek maka $\Delta I_c/I$ besar.

2.1.4. Rasio Weber

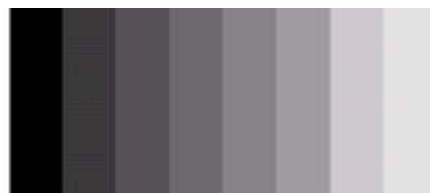
Brightness discrimination kecil (ratio weber besar) di tingkat pencahayaan yang rendah dan meningkat secara signifikan (ratio berkurang) karena peningkatan pencahayaan pada background. Sulit untuk membedakan *discrimination* ketika daerah cerah dan akan lebih mudah membedakan jika di daerah gelap. Tentang rasio weber yang berbanding terbalik dengan discrimination brightness dapat dilihat pada gambar 2.6.



Gambar 2.6. Brightness Discrimination dengan Rasio Weber

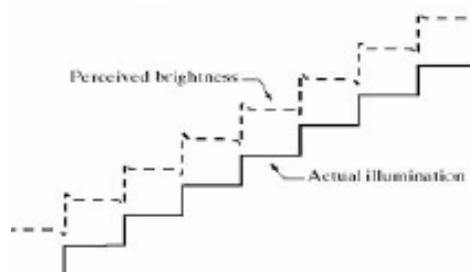
2.1.5. Brightness Sebagai Fungsi Intensitas

Brightness bukanlah fungsi sederhana dari intensitas. Sistem visual cenderung untuk lebih rendah atau lebih tinggi di sekitar daerah batas dengan intensitas yang berbeda.



Gambar 2.7. Pola Intensitas Brightness

Perhatikan gambar 2.7. intensitas dari garis adalah konstan tetapi kita bisa melihat suatu pola kecerahan yang lebih tinggi di dekat perbatasan garis, gafik gambar 2.8.



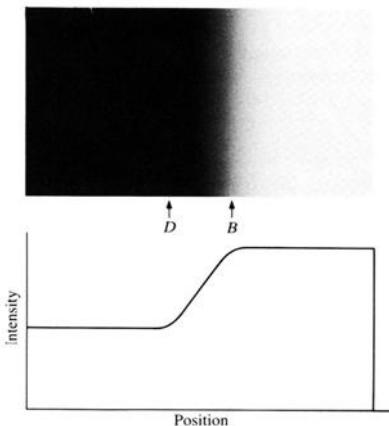
Gambar 2.8. Brightnes yang Diterima dengan Pencahayaan Aktual

2.1.6. Pola Mach Band

Pola Mach Band ditemukan oleh Ernst Mach 1865.

Mengacu pada gambar 2.9. muncul pertanyaan apakah level kegelapan daerah b dan d sama ?

Pola brightness yang diterima tampak lebih gelap di daerah d dan lebih terang dibandingkan daerah b dimana sebenarnya daerah dari d ke b mempunyai intensitas yang sama



Gambar 2.9. Ilustrasi Pola Mach Band

2.1.7. Simultan Kontras

Dengan memperhatikan gambar 2.10. dapat dilihat kotak kecil disebelah kiri kelihatan lebih gelap dari kotak kecil disebelah kanan, padahal intensitasnya sama tetapi intensitas latar belakang berbeda. Hal yang sama terjadi bila kertas putih di meja kelihatan lebih putih daripada kertas sama diarahkan ke sinar matahari.



Gambar 2.10. Contoh Simultan Kontras

2.2. Sinyal

Sinyal adalah sebuah fungsi yang membawa informasi. Pada umumnya berisi sinyal perubahan atas beberapa himpunan dimensi *spatiotemporal*. Dari makna *spatiotemporal* adalah yang ada dalam ruang dan waktu, memiliki ekstensi spasial dan durasi temporal.

2.2.1. Sinyal Waktu

Beberapa sinyal bervariasi dari waktu ke waktu: $f(t)$

contoh: sinyal audio

Mungkin pada satu tingkat sebagai kumpulan berbagai nada yang berbeda terdengar frekuensi yang bervariasi dari waktu ke waktu.

2.2.2. Sinyal Spatial

Sinyal dapat bervariasi setiap ruang juga. Sebuah gambar yang dapat dianggap sebagai suatu fungsi dari 2 dimensi: $f(x, y)$ untuk gambar monokrom, nilai fungsi adalah jumlah cahaya pada titik tersebut. CAT medis dan MRI scanner menghasilkan gambar yang merupakan fungsi 3 dimensi: $f(x,y,z)$

2.2.3. Sinyal Spatialtemporal

Fungsi sinyal spatialtemporal : $f(x, y, t)$

x dan y adalah dimensi spasial, t adalah waktu.

Contoh : sinyal video, animasi, atau gambar bergerak lainnya.

2.2.4. Tipe sinyal

Kebanyakan sinyal yang terjadi secara alamiah adalah fungsi pada domain kontinyu. Namun, sinyal di komputer memiliki sampel diskrit dari domain kontinyu, dengan kata lain, sinyal dimanipulasi oleh komputer mempunyai domain diskrit.

2.2.5. Analog Dan Digital

Sinyal yang terjadi secara alamiah juga memiliki rentang bernilai real di mana nilai-nilai terjadi dengan presisi yang tak terbatas (infinite).

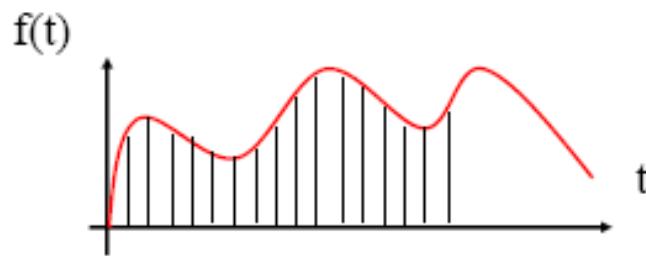
Untuk menyimpan dan memanipulasi sinyal dengan komputer kita perlu menyimpan nilai-nilai dengan ketepatan yang terbatas (finite) dengan demikian, sinyal-sinyal ini memiliki rentang diskrit.

Sinyal dengan range dan domain kontinyu disebut analog.

Sinyal dengan range dan domain distrit disebut digital.

2.2.6. Sampling

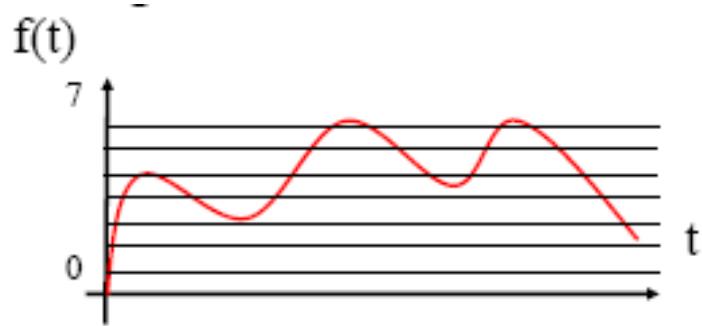
- Sampling adalah jarak dari nilai-nilai diskrit dalam domain sinyal.
- Sampling-rate adalah berapa banyak sampel yang diambil per unit dari masing-masing dimensi. misal, sampel per detik, frame per detik, dll



Gambar 2.11. Proses Sampling

2.2.7. Kuantisasi

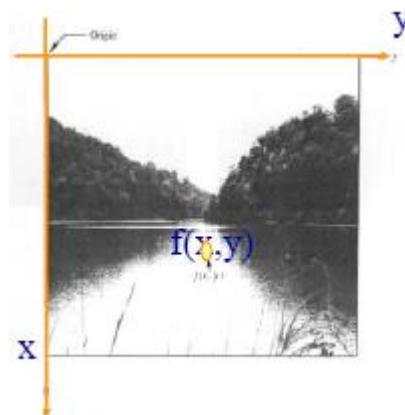
Kuantisasi adalah jarak nilai-nilai diskrit dalam range sinyal. Biasanya dianggap sebagai jumlah bit per sampel dari sinyal. misalnya, 1 bit per piksel (citra hitam/putih), 16-bit audio, 24-bit citra berwarna. Delapan level sama dengan 2^3 maka menggunakan 3 bit untuk menyatakan nilai fungsi.



Gambar 2.12. Proses Kuantisasi

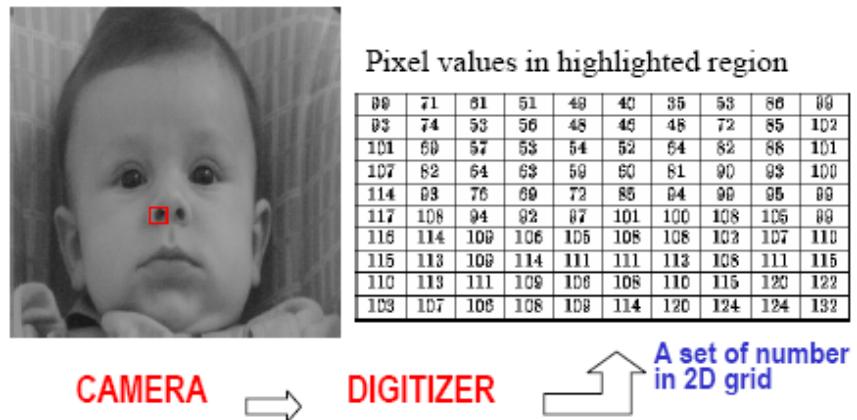
2.3. Representasi Citra Digital

Citra digital adalah sebuah citra $f(x, y)$ yang telah didigitasi baik dalam koordinat spatial dan brightness. Nilai f pada setiap titik (x, y) sebanding dengan kecerahan (atau tingkat abu-abu) dari citra pada titik tersebut.

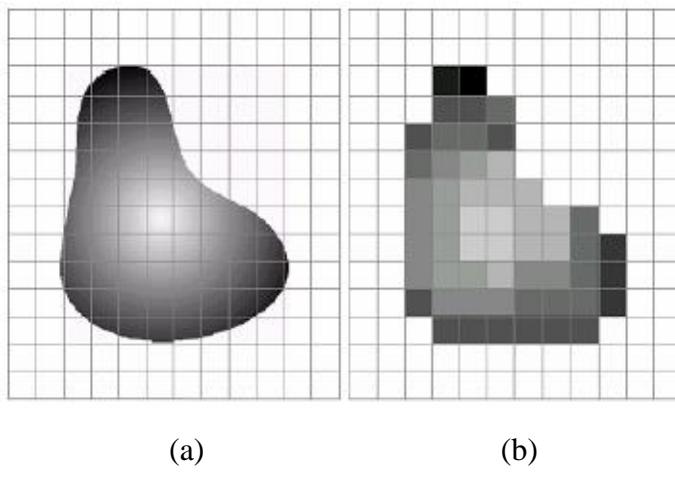


Gambar 2.13. Citra Digital

Citra digital dapat dinyatakan sebagai matrik dimana indeks baris dan kolom mengidentifikasi piksel dalam citra dan nilai elemen matrik yang sesuai mengidentifikasi tingkat keabuan pada piksel tersebut.



Gambar 2.14. Representasi Citra Digital



Gambar 2.15. (a) Citra kontinu diproyeksikan ke sensor array

Gambar 2.15. (b) Citra hasil sampling dan kuantisasi

2.3.1. Fungsi Intensitas Cahaya

Citra 2D mengacu pada fungsi intensitas cahaya $f(x, y)$. Amplitudo dari f pada koordinat spasial (x, y) memberikan intensitas (brightness) dari gambar pada piksel tersebut. Cahaya adalah suatu bentuk energi sehingga $f(x, y)$ harus tidak sama dengan nol dan terbatas.

$$0 < f(x, y) < \infty$$

2.3.2. Gray Level

Kita sebut intensitas dari citra monokrom f pada koordinat (x, y) tingkat keabuan (l) dari citra pada suatu piksel, sehingga, l berada dalam range $l_{min} \leq l \leq l_{max}$, nilai l_{min} positif dan l_{max} terbatas. Range gray scale $[l_{min}, l_{max}]$ pada prakteknya, menggeser interval untuk $[0, l]$, dimana $0 =$ hitam, $l =$ putih

2.3.3. Jumlah Bit

Jumlah dari derajat keabuan pada umumnya merupakan bilangan pangkat dari dua.

$$L = 2^k$$

Jumlah bit diperlukan untuk menyimpan citra digitasi, $k=8$ untuk citra gray scale.

$$b = m \times n \times k$$

2.3.4. Resolusi

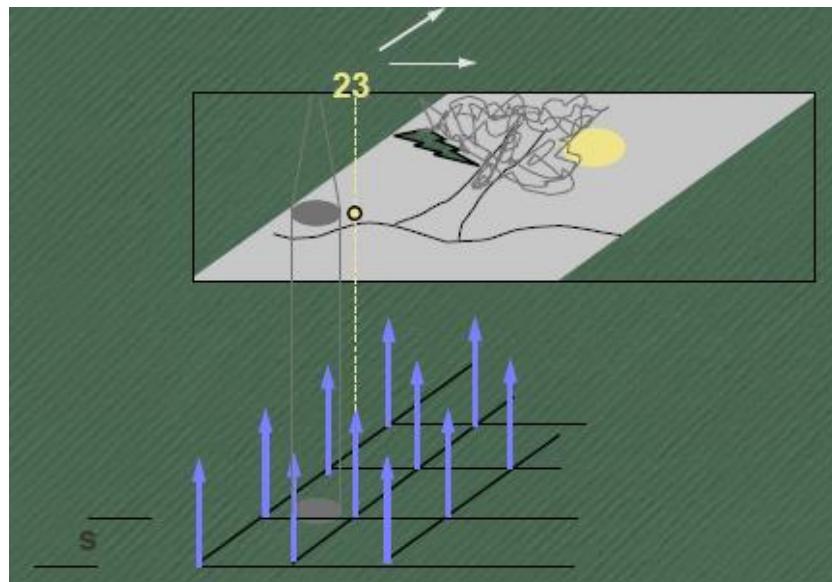
Resolusi citra dikenal dua macam : resolusi spasial dan resolusi kecemerlangan, berpengaruh pada besarnya informasi citra yang hilang.

1. Resolusi spasial: halus / kasarnya pembagian kisi-kisi baris dan kolom. transformasi citra kontinue ke citra digital disebut dijitisasi (sampling). hasil dijitisasi dengan jumlah baris 256 dan jumlah kolom 256 - resolusi spasial 256 x 256.
2. Resolusi kecemerlangan (intensitas / brightness): halus / kasarnya pembagian tingkat kecemerlangan. transformasi data analog yang bersifat kontinue ke daerah intensitas diskrit disebut kwantisasi. bila intensitas piksel berkisar antara 0 dan 255 - resolusi kecemerlangan citra adalah 256.

2.3.5. Resolusi Spatial

Resolusi spatial adalah halus / kasarnya pembagian kisi-kisi baris dan kolom. Transformasi citra kontinue ke citra digital disebut dijitisasi (**sampling**). Hasil digitasi dengan jumlah baris 256 dan jumlah kolom 256 - resolusi spasial 256 x 256.

Proses capture pada kamera melakukan penangkapan besaran intensitas cahaya pada sejumlah titik yang ditentukan oleh besar kecilnya kemampuan resolusi sebuah kamera. Proses pengambilan titik-titik ini dinamakan dengan sampling.



Gambar 2.16. Proses Resolusi Spatial

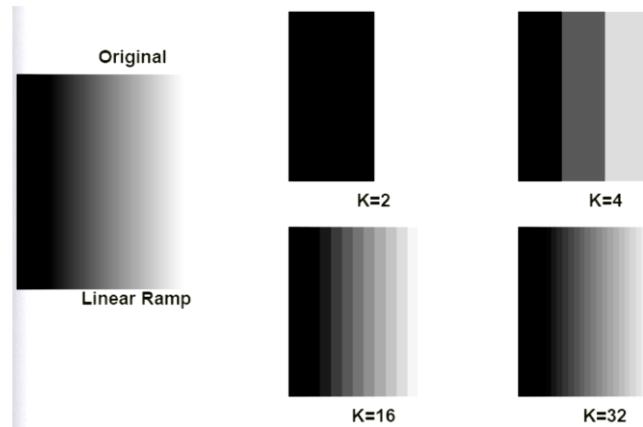
- Sampling Uniform dan Non-Uniform

Sampling Uniform mempunyai spasi (interval) baris dan kolom yang sama pada seluruh area sebuah citra.

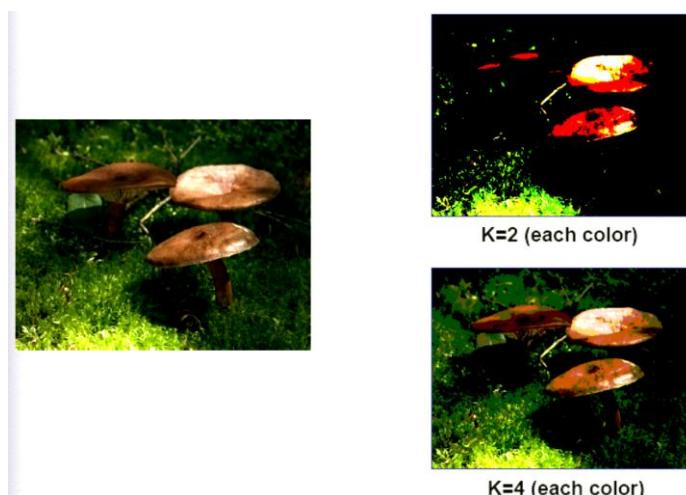
- Sampling Non-uniform bersifat adaptif tergantung karakteristik citra dan bertujuan untuk menghindari adanya informasi yang hilang. Daerah citra yang mengandung detil yang tinggi di-sampling secara lebih halus, sedangkan daerah yang homogen dapat di-sampling lebih kasar. Kerugian sistem sampling Non-uniform adalah diperlukannya data ukuran spasi atau tanda batas akhir suatu spasi.

2.3.6. Resolusi Kecemerlangan

Resolusi kecemerlangan (intensitas / brightness) adalah halus / kasarnya pembagian tingkat kecemerlangan. Transformasi data analog yang bersifat kontinue ke daerah intensitas diskrit disebut **kwantisasi**. Bila intensitas piksel berkisar antara 0 dan 255 - resolusi kecemerlangan citra adalah 256.



Gambar 2.17. Kwantisasi Hitam Putih



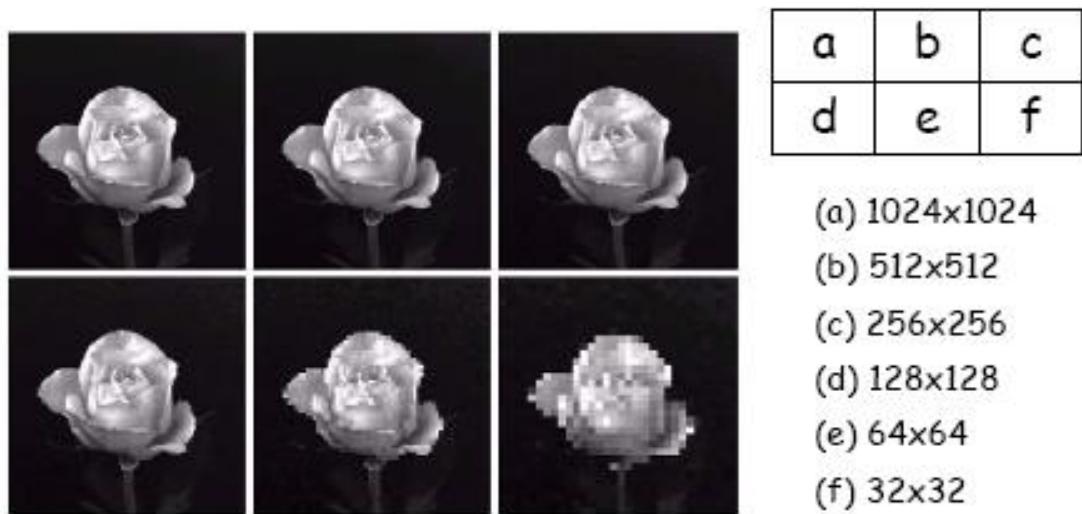
Gambar 2.18 Kwantisasi Warna

Kwantisasi Unifor, Non-uniform dan Tapered :

- Kwantisasi Uniform mempunyai interval pengelompokan tingkat keabuan yang sama (misal: intensitas 1 s/d 10 diberi nilai 1, intensitas 11 s/d 20 diberi nilai 2, dstnya).
- Kwantisasi Non-uniform: Kwantisasi yang lebih halus diperlukan terutama pada bagian citra yang meng-gambarkan detil atau tekstur atau batas suatu wilayah obyek, dan kwantisasi yang lebih kasar diberlakukan pada wilayah yang sama pada bagian obyek.
- Kwantisasi Tapered: bila ada daerah tingkat keabuan yang sering muncul sebaiknya di-kwantisasi secara lebih halus dan diluar batas daerah tersebut dapat di-kwantisasi secara lebih kasar (local stretching).

2.3.7. Efek Checkerboard

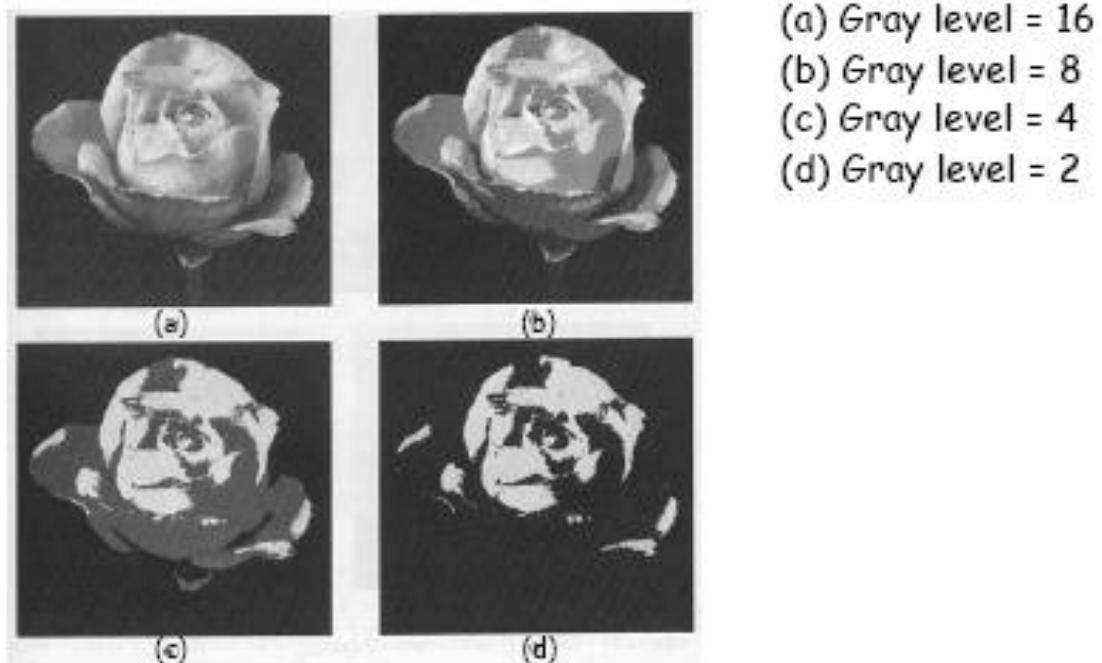
Jika penurunan resolusi terlalu banyak, efek checkerboard dapat terjadi.



Gambar 2.19. Efek Checkerboard

2.3.8. False Counturing

Jika derajat keabuan tidak cukup, daerah yang halus akan terpengaruh. False contouring dapat terjadi di daerah halus pada derajat keabuan yang cukup.



Gambar 2.19. Efek False Counturing

2.3.9. Sampling Non Uniform

Untuk nilai tetap resolusi spasial, tampilan gambar dapat ditingkatkan dengan menggunakan adaptif sampling.

- Sampling halus → sampling yang diperlukan dalam transisi derajat keabuan yang tajam.
- Sampling kasar → sampling yang digunakan di daerah yang relatif halus.



a) b) c)

Gambar 2.20. Contoh Sampling Non Uniform

- a) Gambar dengan level detail yang rendah
- b) Gambar dengan level detail medium
- c) Gambar dengan level detail yang relatif banyak jumlahnya

- Citra dengan wajah terlapisi pada latar belakang yang seragam.
 - Latar belakang → detail informasi sedikit
→ sampling kasar
 - Wajah → lebih detail → disampling halus
 - Jika kita dapat menggunakan sampling adaptif, kualitas citra dapat ditingkatkan.
- Selain itu, kita harus lebih memperhatikan daerah sekitar batas objek → transmisi ketajaman derajat keabuan dari objek ke background.

2.3.10. Kuantisasi Non Uniform

- Ketidaksamaan level space dalam proses kuantisasi mempengaruhi menurunnya jumlah derajat keabuan.

- Gunakan sedikit derajat keabuan di sekitar tepi. Mengapa? Karena mata relatif lemah untuk memperkirakan warna keabuan di dekat perubahan level yang mendadak.
- Gunakan lebih banyak derajat keabuan di daerah yang halus untuk menghindari “false contouring”.

2.4. Relasi Dasar Piksel Pada Citra B/W

Relasi dasar piksel pada citra B/W terdiri dari :

1. Ketetanggaan piksel
2. Konektivitas
3. Penamaan dari komponen yang terhubung
4. Pengukuran jarak
5. Operasi aritmatika/logika

2.4.1. Tetangga Piksel

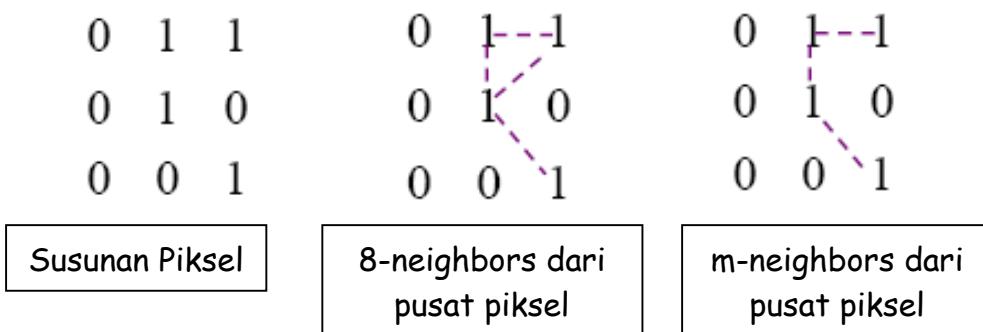
Sebuah piksel p pada koordinat (x,y) dapat mempunyai salah satu model tetangga piksel sbb $N_4(p)$ atau $N_D(p)$ atau $N_8(p)$:

- $N_4(p)$: 4-neighbors of p
 $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$
- $N_D(p)$: 4-diagonal neighbors of p
 $(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$
- $N_8(p)$: 8-neighbors of p :
a combination of $N_4(p)$ and $N_D(p)$

2.4.2. Konektivitas

- V adalah himpunan nilai-nilai derajat keabuan yang digunakan untuk konektivitas yang didefinisikan
 - 4-connectivity:
 - 2 piksel p dan q dengan nilai-nilai dari v adalah 4-connectivity jika q dalam himpunan $n_4(p)$
 - 8-connectivity:

- 2 piksel p dan q dengan nilai-nilai dari v adalah 8-connectivity jika q dalam himpunan $n_8(p)$
- m-connectivity (mixed connectivity):
 - 2 piksel p dan q dengan nilai-nilai dari v adalah m-connectivity jika :
 - q adalah dalam himpunan $n_4(p)$ atau
 - q adalah dalam himpunan $n_d(p)$ dan himpunan $n_4(p) \cap n_4(q)$ adalah kosong.
 - (himpunan yang piksel 4-connectivity dari kedua p dan q yang nilai-nilai dari v)



Gambar 2.21. Contoh Konektivitas Piksel dengan N_8 dan N_m

- M-connectivity menghilangkan koneksi jalur ganda yang muncul dalam 8-connectivity.

2.4.3. Pelabelan Komponen Terhubung

- Piksel p adjacent dengan piksel q jika mereka terhubung. Dua bidang gambar subset s1 dan s2 adalah adjacent jika beberapa piksel pada s1 berbatasan dengan beberapa piksel s2.
- Jalur dari piksel p dengan koordinat (x, y) untuk piksel q dengan koordinat (s, t) adalah sebuah urutan piksel yang berbeda dengan koordinat

$$(x_0, y_0), (x_1, y_1), \dots (x_n, y_n)$$

dimana :

$$(x_0, y_0) = (x, y), (x_n, y_n) = (s, t) \text{ dan}$$

$$(x_i, y_i) \text{ adalah adjacent dengan } (x_{i-1}, y_{i-1})$$

- n adalah panjang jalur
- Kita dapat mendefinisikan 4 -, 8 -, atau m-jalur tergantung pada jenis adjacency yang ditentukan.

0	0	0	0	0	0	0	1	1	0
1	0	0	1	0	0	1	0	0	1
1	0	0	1	0	1	1	0	0	0
0	0	1	1	1	0	0	0	0	0
0	0	1	1	1	0	0	1	1	1

Gambar 2.22. Contoh Subset S1 dan S2

- perhatikan gambar dua subset s1 dan s2
- untuk v = (1), tentukan apakah s1 dan s2 adalah
 - 4-connected
 - 8-connected
 - m-connected
- Scan gambar dari kiri ke kanan
- p menyatakan piksel pada setiap langkah dalam proses scanning
- r menyatakan neighbor atas p
- t menunjukkan tangan kiri neighbor dari p, berurutan.
- ketika kita mendapat p, poin r dan t telah dihitung dan diberi label jika mereka 1's.

r

t *p*

- Jika hasil dari p = 0, pindah
- Jika hasil dari p = 1, periksa r dan t.

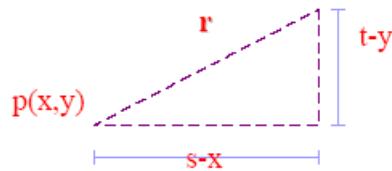
- jika keduanya 0, assign label baru ke p.
- jika salah satunya bernilai 1, assign label tersebut ke p.
- jika keduanya adalah 1
 - jika memiliki tabel yang sama, assign tabel tersebut ke p.
 - jika tidak, assign salah satu tabel ke p dan buat sebuah catatan bahwa kedua label sama. (r dan t terhubung ke p)
- Pada akhir scan, semua point yang bernilai 1 telah diberi label.
- Lakukan scan lagi, assign label baru untuk setiap label yang sama.
Apa yang harus kita lakukan dengan 8-connected komponen?
- lakukan hal yang sama namun periksa juga tetangga diagonal atas dari p.
 - jika p adalah 0, pindah.
 - jika p adalah 1
 - jika 4 tetangga semuanya adalah 0, assign label baru ke p.
 - jika hanya 1 tetangga yang bernilai 1, assign label tersebut ke p.
 - jika 2 atau lebih tetangga bernilai 1, assign salah satu label ke p dan buat catatan bahwa kelasnya sama.
 - setelah menyelesaikan scan, lakukan putaran kedua dan mengenalkan tabel yang unik kepada setiap kelas yang sama

2.4.4. Pengukuran Jarak

- Untuk piksel p, q, dan z dengan koordinat (x,y), (s,t) dan (u,v) berurutan.
- d adalah sebuah fungsi jarak atau matrik jika :
 - (a) $d(p,q) \geq 0$; $d(p,q) = 0$ iff $p=q$
 - (b) $d(p,q) = d(q,p)$
 - (c) $d(p,z) \leq d(p,q) + d(q,z)$

Jarak euclidean antara p dan q

$$D_e(p, q) = \sqrt{(x-s)^2 + (y-t)^2}$$



radius (r) terpusat pada (x,y)

Jarak city-block : jarak d4

$$D_4(p, q) = |x - s| + |y - t|$$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	2		
2				

- Terpusat (x,y)
- $d_4 = 1$ adalah 4-neighbors dari (x,y)

Jarak chessboard : jarak d8

$$D_8(p, q) = \max(|x - s| + |y - t|)$$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Terpusat pada (x,y)

- Jarak d_4 dan jarak d_8 antara piksel p dan q = panjang 4- dan 8-path antara kedua poin, berurutan.
- Kita dapat membandingkan baik jarak d_4 dan jarak d_8 antara p dan q tanpa memperhatikan apakah jalan yang terhubung ada di antara mereka karena definisi jarak ini hanya melibatkan koordinat.
- Jarak dari m-connectivity pada jalur antara 2 piksel tergantung pada nilai piksel sepanjang jalur.

- Misalnya, jika konektivitas dari piksel bernilai 1 maka dilewatkan. temukan m-distance antara p dan p_4

p_3	p_4	0	1	1	1	1	1
p_1	p_2	0	1	0	1	1	1
p		1		1		1	

distance = 2 distance = 3 distance = 4

2.4.5. Operasi Aritmatika Dan Logika

Digunakan secara ekstensif di sebagian besar cabang pemrosesan citra.

- operasi aritmatika 2 antara piksel p dan q:
- penambahan: $p + q$ digunakan dalam gambar rata-rata untuk mengurangi noise.
- pengurangan: $p - q$ alat dasar dalam pencitraan medis.
- perkalian: $p \times q$
- untuk memperbaiki bayangan derajat keabuan hasil dari ketidakseragaman pada pencahayaan atau dalam sensor digunakan untuk memperoleh gambar.
- pembagian: $p \div q$
- operasi aritmatika seluruh gambar dilakukan piksel dengan piksel.
- and : p and q
 $(p \cdot q)$
- or : p or q
 $(p + q)$
- complement : not q (\bar{q})
- operasi logika hanya untuk gambar biner.
- operasi aritmatika dipakai untuk multivalue piksel.
- operasi logika digunakan untuk pekerjaan seperti masking, deteksi fitur, dan analisa bentuk.
- operasi logika dilakukan piksel demi piksel

Operasi Mask

- disamping pemrosesan piksel demi piksel pada gambar, operasi aritmatika dan logika digunakan dalam operasi berorientasi tetangga.

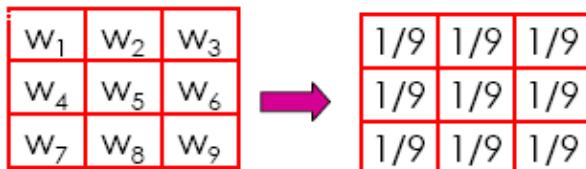
		:	
	Z_1	Z_2	Z_3
...	Z_4	Z_5	Z_6
	Z_7	Z_8	Z_9
		:	

- Nilai yang ditetapkan terhadap piksel merupakan fungsi dari derajat keabuan dan derajat keabuan dari tetangga-tetangganya.
- misalnya, ganti nilai keabuan piksel Z_5 dengan rata-rata nilai keabuan terhadap sekitarnya dalam mask 3x3.

$$Z = \frac{1}{9}(Z_1 + Z_2 + Z_3 + \dots + Z_9)$$

Dalam kondisi umum :

$$\begin{aligned} Z &= \frac{1}{9}Z_1 + \frac{1}{9}Z_2 + \frac{1}{9}Z_3 + \dots + \frac{1}{9}Z_9 \\ &= w_1Z_1 + w_2Z_2 + w_3Z_3 + \dots + w_9Z_9 \\ &= \sum_{i=1}^9 w_iZ_i \end{aligned}$$



Gambar 2.23. Cara Kerja Operasi Masking

Koefisien Mask

- Pemilihan yang tepat dan penerapan koefisien mask pada setiap posisi piksel dalam gambar memungkinkan berbagai operasi gambar yang berguna
 - noise reduction
 - region thinning
 - edge detection
- Menerapkan mask di lokasi masing-masing piksel dalam gambar adalah tugas komputasi mahal.

2.5. Ringkasan

1. Subjective brightness adalah tingkat kecemerlangan yang dapat ditangkap sistem visual manusia yang merupakan fungsi logaritmik dari intensitas cahaya yang masuk ke mata manusia. Mempunyai daerah intensitas yang bergerak dari ambang scotopic (redup) ke ambang photopic (terang)
2. Brightness adaptation merupakan fenomena penyesuaian mata manusia dalam membedakan gradasi tingkat kecemerlangan
3. Citra digital merupakan fungsi intensitas cahaya $f(x,y)$, dimana harga x dan y merupakan koordinat spasial dan harga fungsi tersebut pada setiap titik (x,y) merupakan tingkat kecemerlangan citra pada titik tersebut.
4. Dalam citra digital dikenal istilah resolusi spatial dan resolusi kecemerlangan
5. Resolusi spatial adalah halus / kasarnya pembagian kisi-kisi baris dan kolom.
6. Resolusi kecemerlangan adalah adalah halus / kasarnya pembagian tingkat kecemerlangan.
7. Relasi dasar piksel pada citra B/W terdiri dari : ketetanggaan piksel, konektivitas, penamaan dari komponen yang terhubung, pengukuran jarak, Operasi aritmatika/logika

2.6. Latihan

1. Berikan penjelasan singkat mengenai sistem visual manusia!
2. Apa yang dimaksud dengan subjective brightness?
3. Berikan penjelasan mengenai fenomena match band dan simultaneus contrast!
4. Apa yang dimaksud dengan sampling atau digitasi citra?
5. Apa yang mempengaruhi resolusi spatial pada sebuah Citra digital?
6. Resolusi kecemerlangan dipengaruhi oleh kwantisasi sebuah citra. Jelaskan apa yang dimaksud dengan resolusi kecemerlangan dan apa hubungannya dengan kwantisasi!
7. Jelaskan perbedaan antara kwantisasi uniform dan kwantisasi non-uniform!
8. Apa yang dimaksud dengan efek checkerboard dan false contouring ?
9. Gambarkan posisi piksel pada ketetanggaan piksel N4(p), ND(p) dan N8(p)!

BAB 3

PENINGKATAN MUTU CITRA DOMAIN SPATIAL

POKOK BAHASAN:

- ✓ Jenis Teknik Peningkatan Mutu Citra
- ✓ Pemrosesan Piksel
- ✓ Citra Negatif
- ✓ Kontras Stretching
- ✓ Thresholding
- ✓ Transformasi Tingkat Keabuan
- ✓ Gray Level Slicing
- ✓ Bit Plane Slicing
- ✓ Histogram Equalisasi
- ✓ Histogram Equalisasi Spesifik
- ✓ Peningkatan Mutu Citra dengan Operasi Logika dan Aritmatika

TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat:

- ✓ Mengerti macam-macam metode peningkatan mutu citra di domain spatial
- ✓ Melakukan pemrosesan terhadap citra agar hasilnya mempunyai kualitas relatif lebih *baik* dari citra awal untuk aplikasi tertentu

3.1. Jenis Teknik Peningkatan Mutu Citra

Teknik peningkatan mutu citra dapat dibagi menjadi dua:

1. Peningkatan mutu citra pada domain spasial : teknik yang didasarkan pada manipulasi langsung piksel pada citra
 - Point Processing
 - Mask Processing
2. Peningkatan mutu citra pada domain frekuensi : teknik yang didasarkan pada modifikasi transformasi fourier pada citra
3. Terdapat beberapa teknik peningkatan mutu citra berdasarkan pada kombinasi dari domain spasial dan domain frekuensi

3.1.1. Citra yang Baik

Definisi citra yang baik sebagai berikut :

1. Dari sisi visual manusia

- Evaluasi visual tentang kualitas gambar adalah proses yang sangat subjektif
- Sulit untuk membuat standard definisi citra yang baik

2. Dari sisi mesin / computer

- Tugas Evaluasi lebih mudah.
- Citra yang baik adalah citra yang mampu dikenali dengan baik oleh mesin / komputer.

3. Sejumlah trial and error biasanya diperlukan sebelum pendekatan peningkatan mutu citra tertentu dipilih.

3.2. Domain Spatial

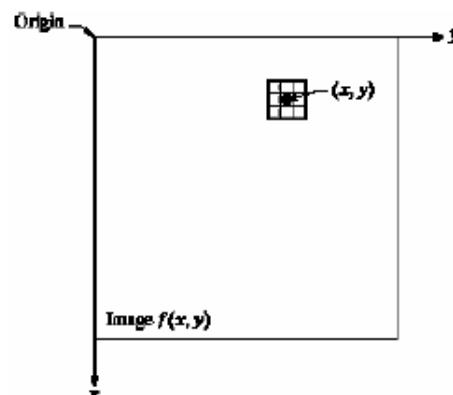
Domain spatial adalah prosedur yang beroperasi langsung pada piksel.

$$g(x, y) = T[f(x, y)]$$

di mana :

- $f(x, y)$ adalah gambar input
- $g(x, y)$ adalah diproses gambar
- T adalah operator pada f didefinisikan melalui beberapa lingkungan (x, y)

Perhatikan ilustrasi Domain Spatial pada gambar 3.1.

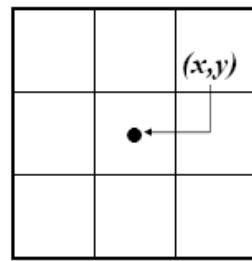


Gambar 3.1. Domain Spatial

3.3. Mask/Filter

Pemrosesan di domain spatial selain melibatkan 1 piksel juga melibatkan ketetanggaan dari piksel (x, y) dapat didefinisikan dengan menggunakan persegi atau lingkaran daerah sub citra berpusat di (x, y) . Bagian tengah sub citra dipindahkan dari

piksel ke piksel dimulai pada bagian ujung kiri atas citra. Piksel (x,y) di center dengan 8 ketetanggaan pada gambar 3.2.



Gambar 3.2. Mask/Filter

3.4. Pemrosesan Piksel

Cara paling mudah untuk melakukan peningkatan mutu pada domain spasial adalah dengan melakukan pemrosesan yang hanya melibatkan satu piksel saja (tidak menggunakan jendela ketetanggaan). Pengolahan menggunakan histogram juga termasuk dalam bagian pemrosesan piksel. Perhatikan contoh pemrosesan piksel berikut :

- Ketetanggaan = 1×1 pixel
- g hanya tergantung hanya pada nilai f pada (x, y)
- T adalah tingkat keabuan (atau intensitas atau pemetaan) fungsi transformasi

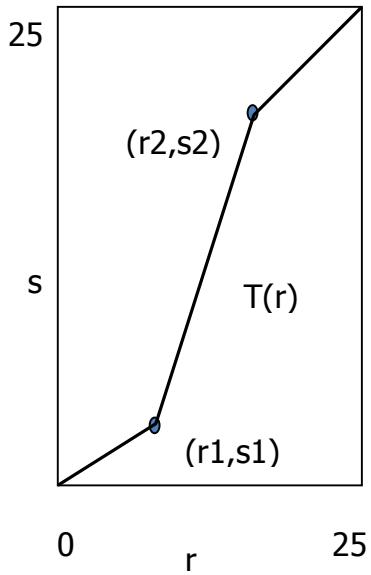
$$s = T(r)$$

Dimana :

- r = tingkat keabuan dari $f(x, y)$
- s = tingkat keabuan dari $g(x, y)$

3.5. Kontras Stretching

Kontras stretching adalah merubah kontras dari suatu citra dengan cara merubah greylevel piksel-piksel pada citra menurut fungsi $s = T(r)$ tertentu.

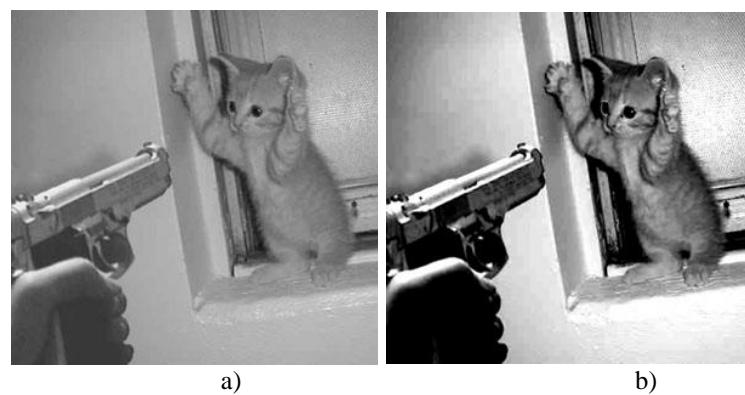


Gambar 3.3. Kontras Stretching

Perhatikan gambar 3.1. yang mengilustrasikan proses kontras stretching sebagai berikut :

- $r_1 \leq r_2, s_1 \leq s_2$
- $r_1 = r_2, s_1 = s_2 \rightarrow$ tidak ada perubahan
- $r_1 = r_2, s_1 = 0, s_2 = 255 \rightarrow$ tresholding menjadi citra biner dengan ambang r_1

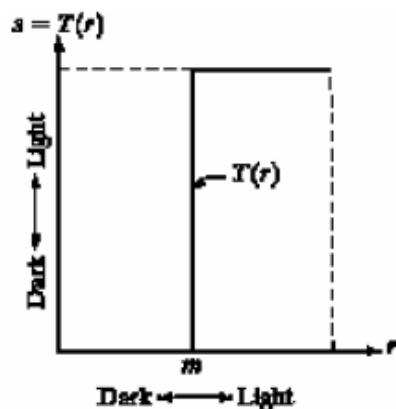
Contoh hasil kontras stretching perhatikan gambar 3.4. :



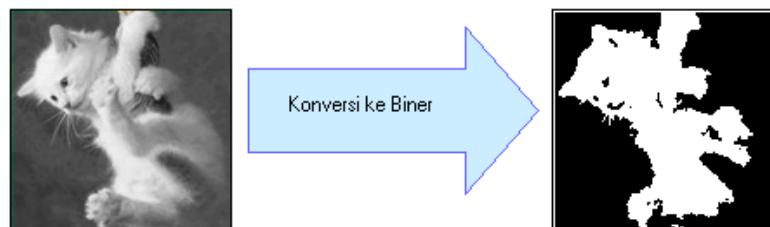
Gambar 3.4. a) Citra asal b) Citra Hasil Kontras Stretching

3.6. Thresholding

Pemakaian thresholding dapat menghasilkan citra biner (dua level warna), dengan fungsi $T(r) = m$ sebagai nilai thresholding. Piksel dengan level warna $r \leq m$ akan berubah menjadi warna hitam atau $s=0$, sedangkan piksel dengan level warna $r \geq m$ berubah menjadi warna putih atau $s=255$, perhatikan gambar 3.3. fungsi transformasi untuk citra biner dengan thresholding m .



Gambar 3.5. Fungsi $s=T(r)$ dengan Thresholding m



Gambar 3.6. Contoh Citra Biner dengan Thresholding

Persamaan thresholding citra biner dengan thresholding 128 sebagai berikut :

$$x_{bw} = \begin{cases} 1 & \text{jika } x \geq 128 \\ 0 & \text{jika } x < 128 \end{cases}$$

Sedangkan persamaan thresholding citra biner dengan thresholding nilai piksel rata pada citra \bar{x} sebagai berikut :

$$x_{bw} = \begin{cases} 1 & \text{jika } x \geq \bar{x} \\ 0 & \text{jika } x < \bar{x} \end{cases}$$

Thresholding juga dapat digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Dengan menggunakan thresholding maka derajat keabuan bisa diubah sesuai keinginan, misalkan diinginkan menggunakan derajat keabuan 16, maka tinggal membagi nilai derajat keabuan dengan 16. Proses thresholding disini adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan thresholding dengan derajat keabuan a dapat digunakan rumus:

$$x = b \cdot \text{int}\left(\frac{w}{b}\right)$$

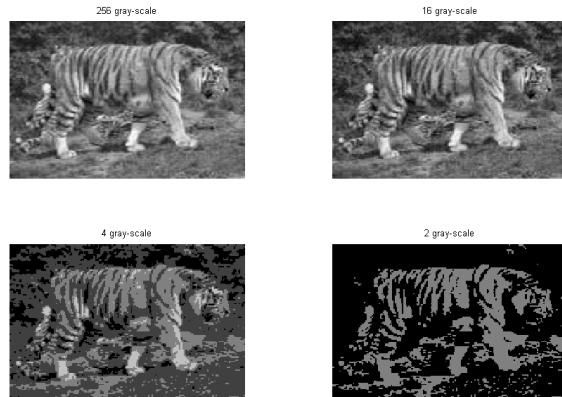
dimana :

w adalah nilai derajat keabuan sebelum thresholding

x adalah nilai derajat keabuan setelah thresholding

$$b = \text{int}\left(\frac{256}{a}\right)$$

Berikut ini contoh thresholding mulai di 256, 16, 4 dan 2 pada gambar.

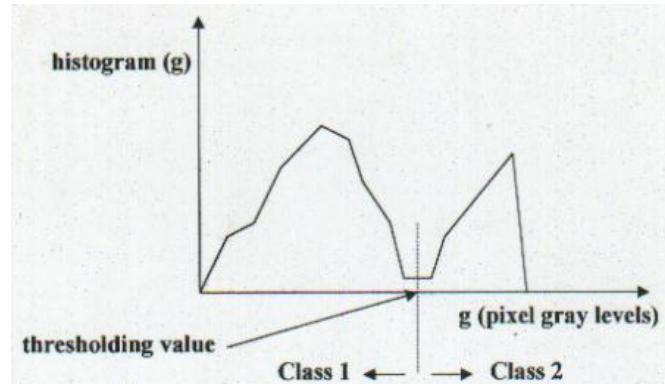


Gambar 3.7. Contoh Thresholding Pada Kuantisasi Citra

Pada threshold yang tinggi, hampir tidak tampak perbedaan karena keterbatasan mata, tetapi pada nilai threshold yang kecil seperti 2, 4, 8 dan 16 tampak sekali perbedaannya. Untuk threshold 2, ini sama artinya dengan mengubah citra menjadi citra biner.

Pemakaian thresholding yang lain dapat digunakan pada proses segmentasi citra dengan menggunakan histogram thresholding. Misalkan histogram gray-level dari citra, $f(x, y)$, yang terdiri dari obyek gelap dengan latar belakang terang, sedemikian sehingga piksel gray scale pada objek dan latar belakang dikelompokkan menjadi dua daerah dominan. Salah satu cara yang untuk mengekstrak objek dari latar belakang dengan memilih ambang batas 'T' yang memisahkan dua daerah. Maka

setiap piksel (x, y) dimana $f(x, y) > T$ disebut piksel obyek, jika tidak berarti piksel tersebut disebut piksel latar belakang. Contoh histogram thresholding pada gambar 3.8.



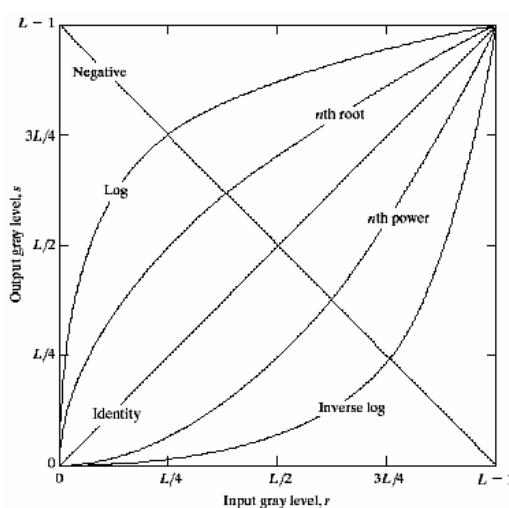
Gambar 3.8. Histogram Thresholding

3.7. Transformasi Tingkat Keabuan

Ada tiga macam transformasi tingkat keabuan yang digunakan untuk perbaikan citra:

1. Linier (negative dan identity transformations)
2. Logaritma (log dan inverse-log transformations)
3. Power-law (n th power dan n th root transformations)

Grafik perubahan tingkat keabuan pada transformasi citra ditunjukkan pada gambar 3.9.



Gambar 3.9. Grafik Perubahan Tingkat Keabuan pada Transformasi Citra

3.7.1. Fungsi Identitas

Fungsi identitas adalah dimana output intensitas pada citra identik dengan intensitas citra input. Pada grafik gambar 3.9. dimasukkan hanya untuk kelengkapan.

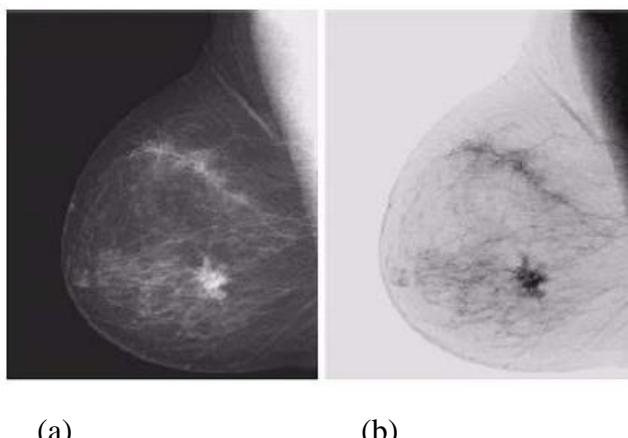
3.7.2. Citra Negatif

Citra dengan tingkat keabuan dalam rentang $[0, L-1]$, di mana $L = 2^n$; $n = 1, 2 \dots$

Transformasi negatif :

$$s = L - I_r$$

Melakukan invers terhadap nilai intensitas dari citra, hasilnya seperti klise foto. Dapat dipakai untuk meningkatkan mutu citra dengan detail warna putih atau keabuan yang ada dalam daerah gelap citra, terutama ketika daerah hitamnya dominan. Perhatikan contoh citra negatif pada gambar 3.10. a) citra mammogram terdapat luka kecil dan gambar 3.10 b) hasil citra negatif dilihat lebih baik untuk menganalisa citra.



(a)

(b)

Gambar 3.10. (a) Citra Mamogram Asal (b) Citra Negatif

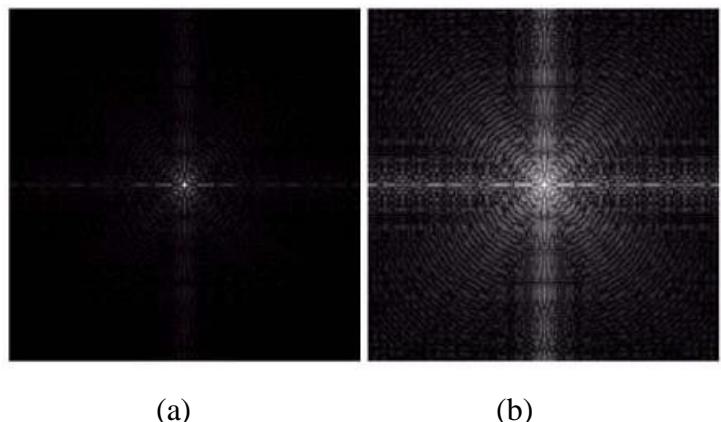
3.7.3. Transformasi Logaritma

Transformasi logaritma dengan formula : $s = c \log (1 + r)$

dimana c adalah sebuah konstanta dan $r \geq 0$

Kurva log memetakan citra input dengan level keabuan pada daerah gelap yang sempit menjadi citra output yang lebih lebar range level keabuannya dan didaerah terang. Digunakan untuk menaikkan nilai-nilai keabuan piksel gelap pada citra selama proses kompresi menjadi nilai keabuan yang tinggi.

Transformasi logaritma secara dinamis akan mengompres mengurangi range citra dengan variasi nilai piksel yang besar. Contoh citra dengan range yang dinamis citra spektrum fourier. Citra spektrum fourier mempunyai rentang intensitas $0-10^6$ atau lebih tinggi. Kita tidak dapat melihat secara signifikan detailnya karena akan hilang pada saat ditampilkan. Contoh citra hasil transformasi logaritma pada gambar 3.11. a) spektrum fourier dengan range 0 sampai dengan 1.5×10^6 dan gambar 3.11 b) hasil transformasi logaritma dengan $c=1$ dan range 0 sampai dengan 6.2.



Gambar 3.11. Contoh Citra Hasil Transformasi Logaritma

3.7.4. Transformasi Inverse Logaritma

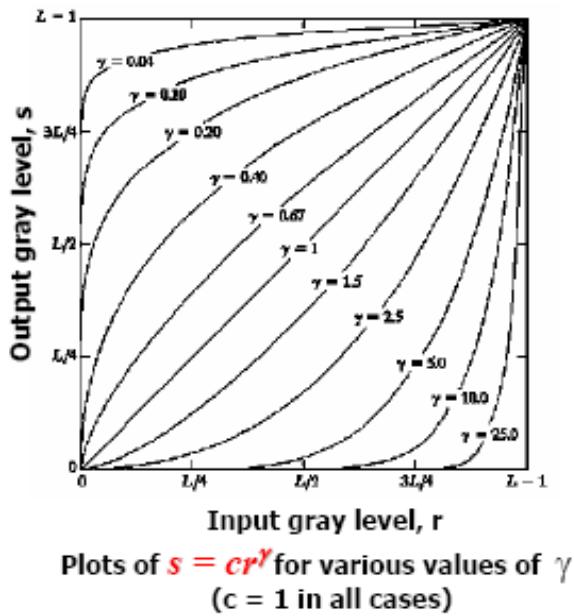
Formula transformasi inverse logaritma : $s = c \log ((L-1) + r)$

Transformasi inverse logaritma mengerjakan transformasi dengan kebalikan dari transformasi logaritma. Digunakan untuk memperluas nilai-nilai piksel tinggi pada citra selama proses kompresi menjadi nilai piksel yang lebih gelap.

3.7.5. Transformasi Algoritma Power Law

Formula transformasi power-law : $s = cr^\gamma$, c dan γ adalah positif konstanta

Kurva power-law dengan nilai pecahan γ memetakan citra input dengan range nilai gelap yang sempit ke nilai output terang yang lebih lebar rangenya, dan kebalikannya pemetaan kurva pada citra input dengan range nilai terang yang lebih lebar menjadi range nilai yang lebih sempit. Pada $c = \gamma = 1$ hasil transformasi sama dengan fungsi identitas. Contoh pemetaan kurva transformasi power law pada gambar 3.12.

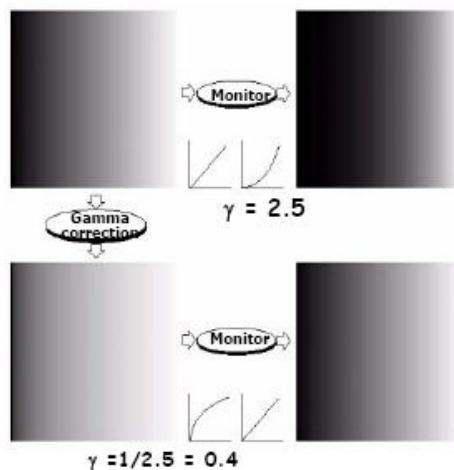


Gambar 3.12. Kurva Mapping Transformasi Power Law ($\gamma = 0.01$ s/d $\gamma = 25$.)

3.7.5.1. Koreksi Gamma

Perangkat CRT (Catode Ray Tube) yang mempunyai intensitas-tegangan respons merupakan fungsi power, dengan γ bervariasi 1.8 sampai dengan 2.5. Citra akan menjadi lebih gelap.

Koreksi gamma dilakukan dengan preprocessing citra sebelum dimasukkan ke monitor dengan $s = cr^{1/\gamma}$. Contoh koreksi gamma pada gambar 3.13. gambar pada baris I lebih gelap hasil di CRT monitor dengan $\gamma = 2.5$ dengan koreksi gamma $\gamma=0.4$ akan menghasilkan CRT monitor lebih terang pada gambar baris ke-2.



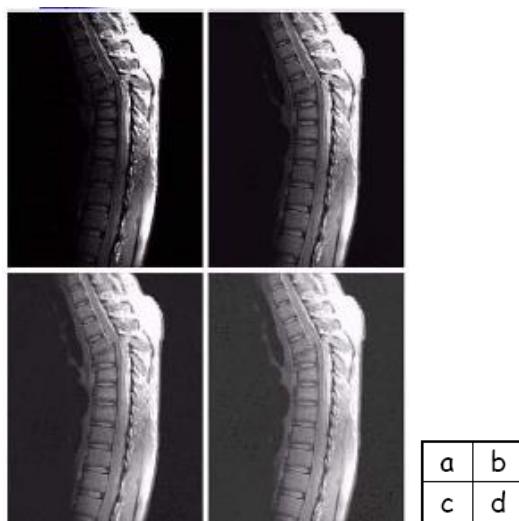
Gambar 3.13. Koreksi Gamma dan Hasil CRT di Monitor

3.7.5.2. Citra MRI

Gambar 3.14. bagian a) adalah citra *Magnetic Resonance Image* (MRI) pada torak bagian atas tulang belakang manusia dengan *fracture dislocation* dan *spinal cord impingement* (rekahan dan pergeseran sumsum tulang belakang) :

- Citra didominasi warna gelap
- Perluasan tingkat keabuan yang dibutuhkan $\rightarrow \gamma < 1$

Gambar 3.14. bagian b) citra hasil setelah transformasi power-law dengan $\gamma = 0.6$ dan $c = 1$. Gambar 3.14. bagian c) citra hasil setelah transformasi power-law dengan $\gamma = 0.4$ (dinyatakan citra dengan hasil terbaik). Gambar 3.14. bagian d) citra hasil setelah transformasi power-law dengan $\gamma = 0.3$ (di bawah tingkat yang dapat diterima).

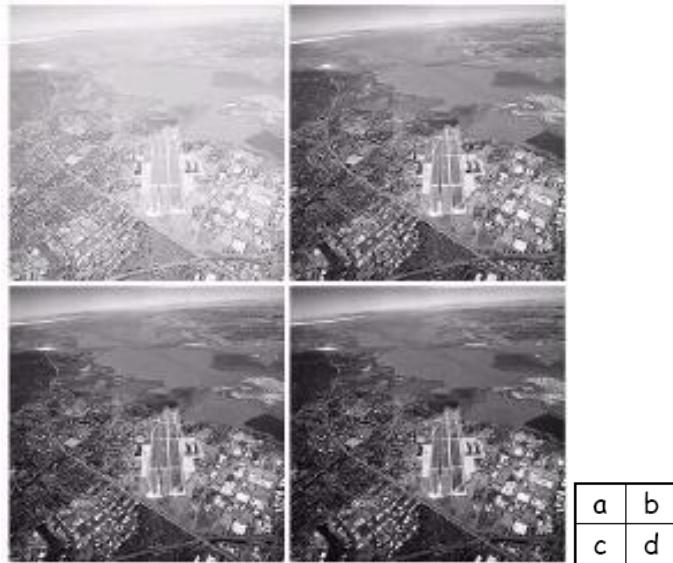


Gambar 3.14. Citra Transformasi Power Law Pada MRI

3.7.5.3. Efek Penurunan Gamma

Terjadi pada saat γ dikurangi terlalu banyak, citra mulai mengurangi kontras pada piksel di mana citra mulai kelihatan sangat sedikit "wash-out" (membersihkan), khususnya di daerah latar belakang. Perhatikan contoh efek penurunan gamma pada gambar 3.15. bagian a) citra memiliki tampilan "wash-out", diperlukan sebuah kompresi tingkat keabuan membutuhkan $\gamma > 1$. Gambar 3.15. bagian b) citra hasil setelah transformasi power-law dengan $\gamma = 3,0$ citra hasil tepat, begitu pula gambar 3.15. bagian c) transformasi power law dengan γ

$\gamma = 4.0$ citra hasil tepat. Sedangkan gambar 3.15. bagian d) transformasi power law dengan $\gamma = 5.0$ menghasilkan citra dengan kontras tinggi, citra memiliki area yang terlalu gelap, dan beberapa detail citra hilang.



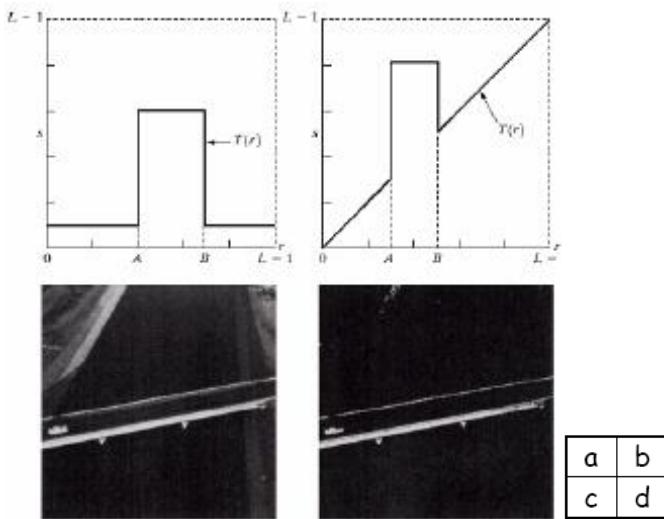
Gambar 3.15. Citra Transformasi Power Law dengan Efek Penurunan Gamma

3.8. Tranformasi Fungsi Linier Piecewise

Keuntungan transformasi fungsi linier piecewise adalah bentuk fungsi piecewise dapat sembarang komplek. Kekurangannya spesifikasi fungsi ini memerlukan lebih banyak input dari user.

3.8.1. Gray Level Slicing

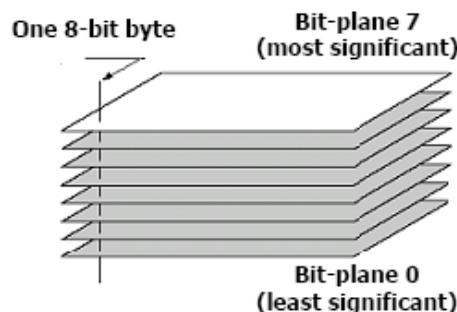
Gray level slicing adalah menekankan range tingkat keabuan tertentu pada sebuah citra, contoh menampilkan dengan nilai tingkat keabuan yang tinggi pada range yang dianggap penting dan menampilkan nilai tingkat keabuan yang rendah pada semua tingkat kelabuan yang lainnya. Dicontohkan pada gambar 3.16. bagian a) grafik fungsi transformasi gray level slicing menekankan pada rentang [A, B] dengan nilai keabuan yang lebih tinggi dan mengurangi nilai tingkat keabuan semua level keabuan di luar rentang [A-B], gambar 3.16. bagian c) hasil citra transformasinya. Sedangkan gambar 3.16. bagian b) grafik fungsi transformasi gray level slicing menekankan pada rentang [A, B] dengan nilai keabuan yang lebih tinggi, tetapi nilai keabuan di luar rentang [A-B] dipertahankan, gambar 3.16. bagian d) hasil citra transformasinya.



Gambar 3.16. Gray Level Slicing

3.8.2. Bit Plane Slicing

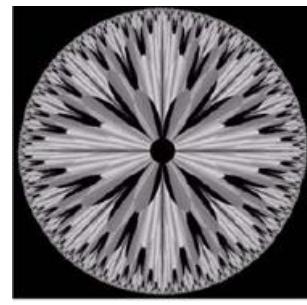
Bit plane slicing adalah menekankan terhadap total penampilan citra dengan bit tertentu. Misalkan setiap piksel diwakili oleh 8 bit, urutan bit tertinggi mengandung mayoritas data visual yang penting. Berguna untuk menganalisis kepentingan relatif yang diperankan oleh setiap bit pada citra. Gambar 3.17. memberikan ilustrasi bit plane 8 bit setiap piksel keabuan citra.



Gambar 3.17. Bit Plane Slicing

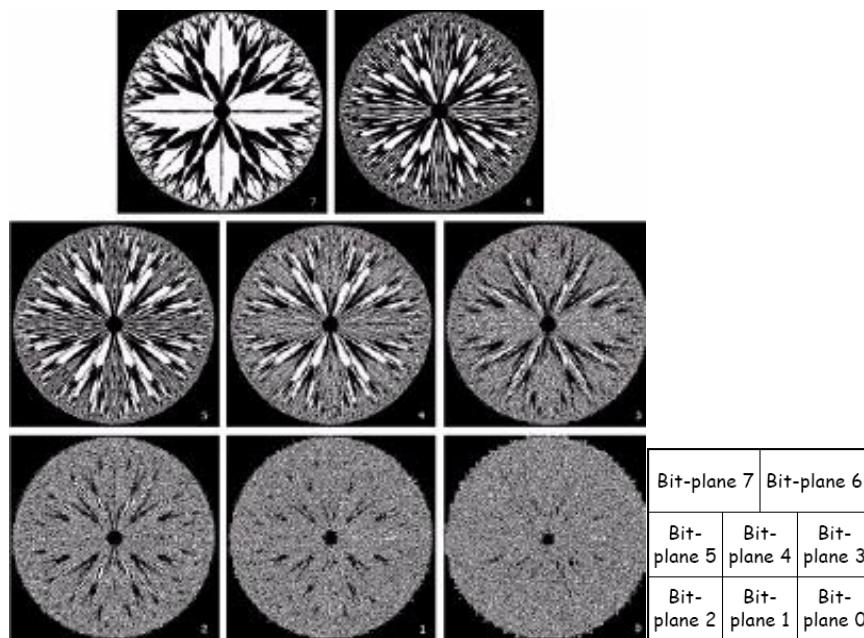
Contoh citra biner (citra 8 bit fraktal, gambar 3.18) bit-plane 7 dapat diperoleh dengan mengolah citra input dengan transformasi derajat keabuan :

- Memetakan semua level keabuan diantara 0 dan 127 ke 0.
- Memetakan semua level keabuan diantara 129 dan 255 ke 1.



Gambar 3.18. Citra Fraktal 8 Bit

Untuk mengetahui penekanan isi dari masing-masing bit dari citra fraktal dapat dilihat pada gambar 3.19.



Gambar 3.19. 8 Bit Planes

3.9. Pemrosesan Dengan Histogram

Histogram citra dengan tingkat keabuan rentang [0, L-1] adalah fungsi diskrit :

$$h(rk) = nk$$

Dimana :

rk : kth tingkat keabuan

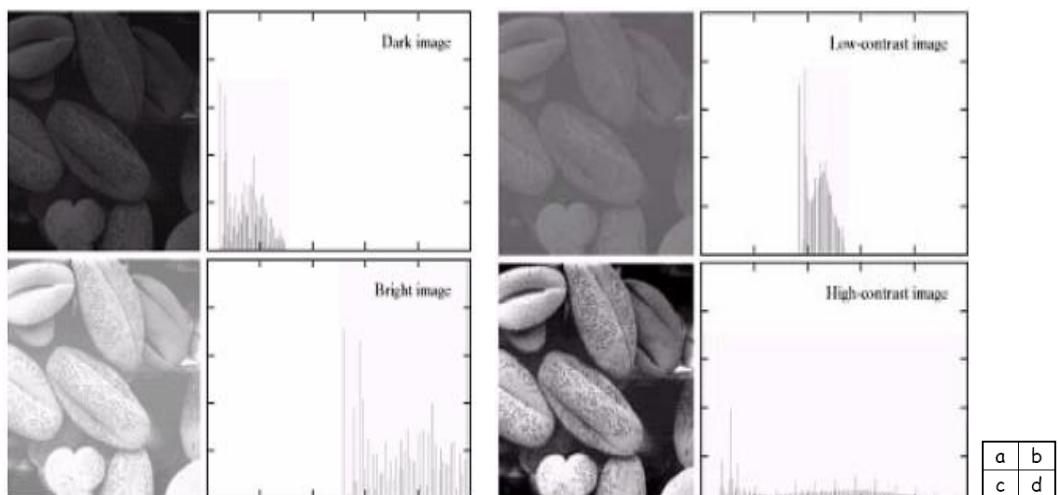
nk : jumlah piksel dalam gambar yang memiliki tingkat keabuan rk

$h(rk)$: histogram citra dengan tingkat keabuan rk

Pemrosesan histogram adalah proses mengubah bentuk histogram agar pemetaan gray level pada citra juga berubah. Proses pada histogram menjadi dasar atas sejumlah

teknik pemrosesan di domain spatial, dikenal juga sangat efektif digunakan untuk peningkatan mutu citra. Informasi yang melekat pada histogram juga berguna untuk kompresi dan segmentasi citra. Karakteristik histogram citra sebagai berikut :

1. Citra gelap : komponen histogram berkonsentrasi pada sisi rendah dari derajat keabuan. Contoh citra dan histogram pada gambar 3.20. bagian a).
2. Citra terang : komponen histogram berkonsentrasi pada sisi tinggi dari derajat keabuan. Contoh citra dan histogram pada gambar 3.20. bagian c).
3. Citra dengan kontras rendah : histogram sempit dan terpusat ke arah tengah derajat keabuan. Contoh citra dan histogram pada gambar 3.20. bagian b).
4. Citra dengan kontras tinggi: histogram mencakup luas rentang derajat keabuan dan distribusi piksel tidak terlalu jauh / distribusi uniform, dengan sangat sedikit garis vertikal yang banyak/lebih tinggi daripada yang lain. Contoh citra dan histogram pada gambar 3.20. bagian d).



Gambar 3.20. Histogram Citra

3.10. Histogram Normalisasi

Histogram normalisasi adalah membagi masing-masing histogram pada tingkat keabuan r_k dengan total jumlah piksel pada citra, n .

$$p(r_k) = n_k / n$$

- Untuk $k = 0, 1, \dots, L-1$
- $p(r_k)$ memberikan perkiraan probabilitas terjadinya tingkat keabuan pada r_k
- Jumlah dari semua komponen pada histogram normalisasi adalah sama dengan satu.

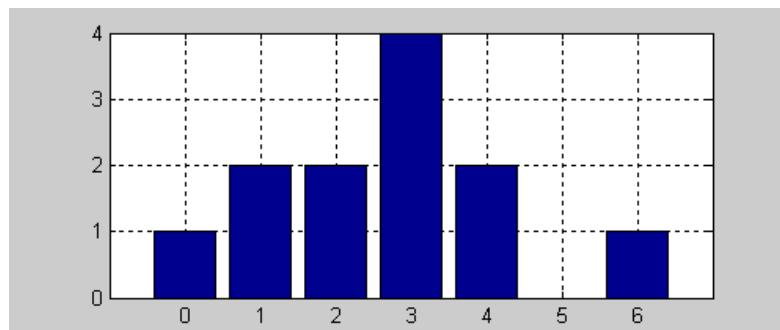
3.11. Histogram Equalisasi

Karena histogram citra dengan kontras rendah sempit dan terpusat di tengah derajat keabuan, jika kita mendistribusikan histogram ke range yang lebih lebar kualitas citra akan meningkat. Kita dapat melakukannya dengan menyesuaikan fungsi probabilitas kepadatan histogram asli citra sedemikian hingga probalitasnya tersebar secara merata. Histogram Equalization adalah suatu proses perataan histogram, dimana distribusi nilai derajat keabuan pada suatu citra dibuat rata. Untuk dapat melakukan histogram equalization ini diperlukan suatu fungsi distribusi kumulatif yang merupakan kumulatif dari histogram.

Misalkan diketahui data sebagai berikut:

2 4 3 1 3 6 4 3 1 0 3 2

Maka histogram dari data di atas adalah:



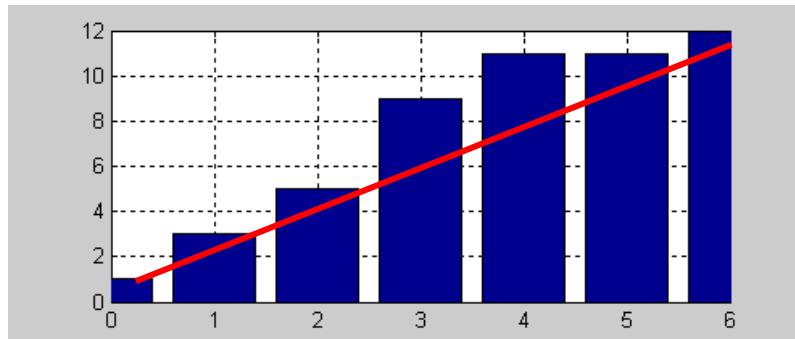
Gambar 3.21. Contoh histogram

Proses perhitungan distribusi kumulatif dapat dijelaskan dengan tabel berikut:

Tabel 3.1. Proses Perhitungan Distribusi Kumulatif

Nilai	Histogram	Distribusi kumulatif
0	1	1
1	2	1+2=3
2	2	3+2=5
3	4	5+4=9
4	2	9+2=11
5	0	11+0=11
6	1	11+1=12

Dari hasil perhitungan di atas diperoleh histogram kumulatif sebagai berikut:



Gambar 3.22. Histogram Kumulatif

Histogram ekualisasi (perataan histogram) adalah suatu proses dimana histogram diratakan berdasarkan suatu fungsi linier (garis lurus) seperti terlihat pada gambar 3.20. Teknik perataan histogram adalah sebagai berikut:

Tabel 3.2. Histogram Ekualisasi

Nilai asal	Histogram Kumulatif	Nilai hasil
0	1	$\frac{1}{2} \rightarrow 0$
1	3	$\frac{3}{2} \rightarrow 1$
2	5	$\frac{5}{2} \rightarrow 2$
3	9	$\frac{9}{2} \rightarrow 4$
4	11	$\frac{11}{2} \rightarrow 5$
5	11	$\frac{11}{2} \rightarrow 5$
6	12	$\frac{12}{2} \rightarrow 6$

Nilai hasil histogram ekualisasi adalah sebagai berikut:

$$w = \frac{c_w \cdot th}{n_x n_y}$$

Dimana

w = nilai keabuan hasil histogram equalization

cw = histogram kumulatif dari w

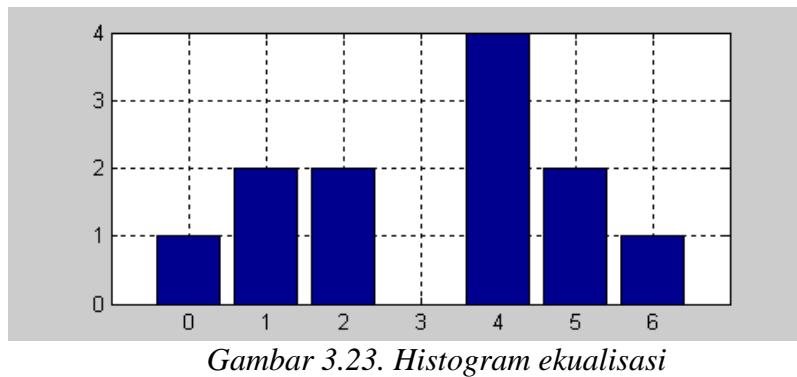
th = threshold derajat keabuan (256)

nx dan ny = ukuran gambar

Hasil setelah histogram ekualisasi adalah sebagai berikut:

2 5 4 1 4 6 5 4 1 0 4 2

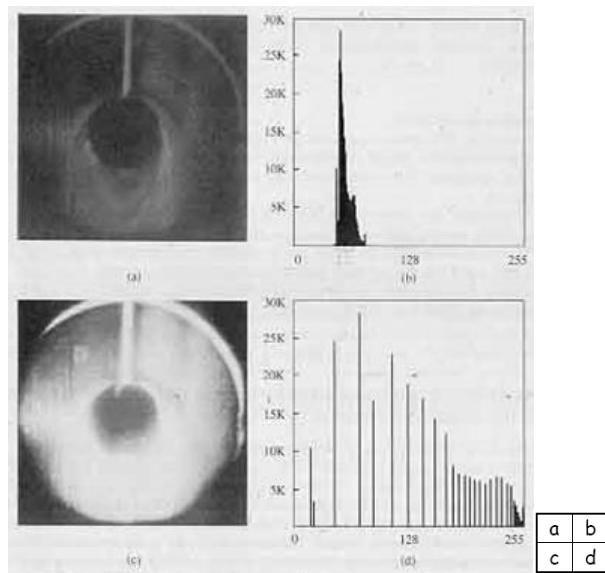
Histogram dari hasil histogram ekualisasi:



Gambar 3.23. Histogram ekualisasi

Sifat histogram ekualisasi sebagai berikut, gambar 3.24.:

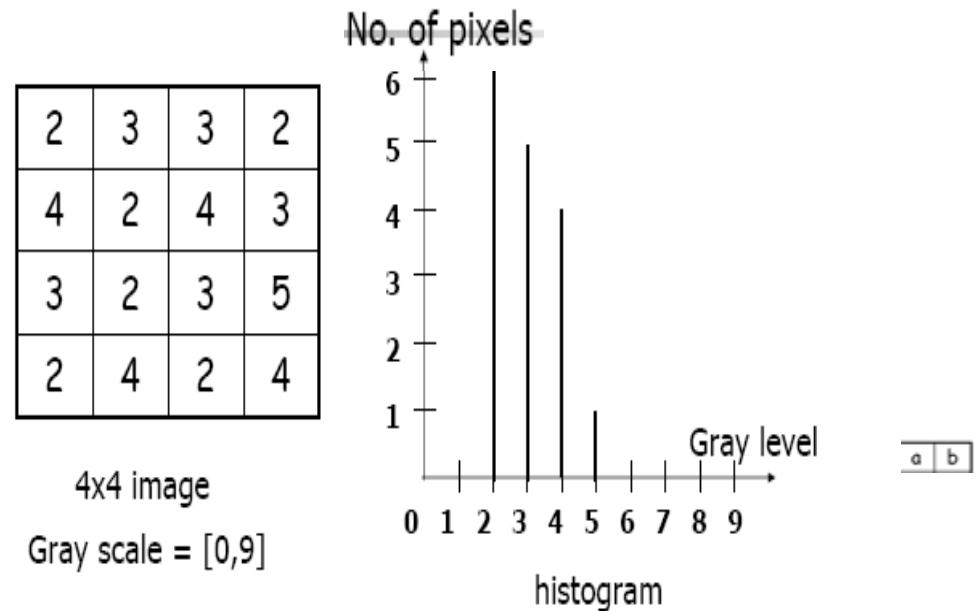
- Grey level yang sering muncul lebih dijarangkan jaraknya dengan grey level sebelumnya
- Grey level yang jarang muncul bisa lebih dirapatkan jaraknya dengan grey level sebelumnya
- Histogram baru pasti mencapai nilai maksimal keabuan (contoh: 255) , gambar



Gambar 3.24. Citra (Asal dan Hist. Ekualisasi) dan Histogram (Asal dan Ekualisasi)

Langkah-langkah proses histogram ekualisasi :

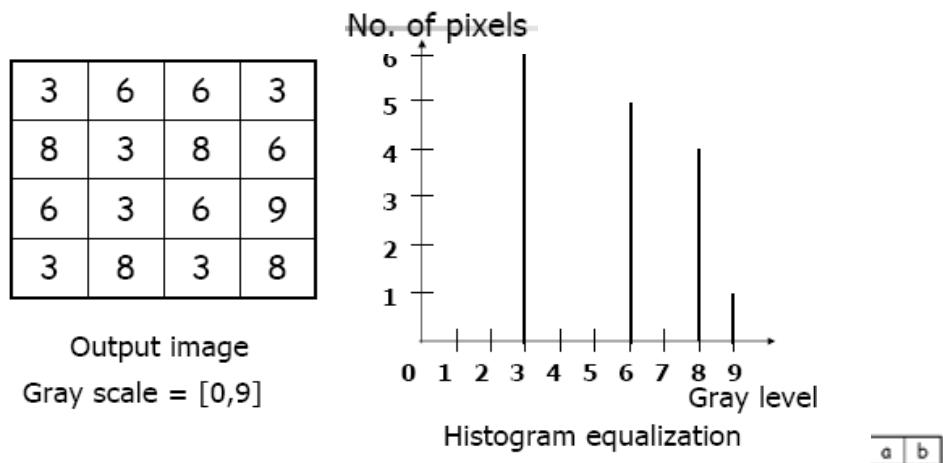
1. Diketahui citra asal, 4x4 dengan range keabuan [0-9] pada gambar 3.25. bagian a.
2. Buat histogram citra asal, gambar 3.25. bagian b.
3. Buat tabel proses histogram ekualisasi, tabel 3.3.
4. Mengacu pada hasil tabel 3.3. buatlah citra output, gambar 3.26. bagian a.
5. Buat histogram ekualisasi mengacu pada citra output no 4, gambar 3.26. bagian b.



Gambar 3.25. Citra Asal dan Histogram Citra Asal

Tabel 3.3. Proses Histogram Ekualisasi

Gray Level(j)	0	1	2	3	4	5	6	7	8	9
No. of pixels	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^k n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^k \frac{n_j}{n}$	0	0	6 / 16	11 / 16	15 / 16	16 / 16	16 / 16	16 / 16	16 / 16	16 / 16
$s \times 9$	0	0	3.3 ≈ 3	6.1 ≈ 6	8.4 ≈ 8	9	9	9	9	9



Gambar 3.26. Citra Output dan Histogram Ekualisasi

Kesimpulan dari histogram ekualisasi :

- Histogram ekualisasi mendistribusikan gray level mencapai maksimum gray level (putih) karena fungsi distribusi kumulatif sama dengan 1 ketika $0 \leq r \leq L-1$
- Jika jumlah kumulatif gray level sedikit berbeda, mereka akan dipetakan ke gray level yang sedikit berbeda atau gray level yang sama karena itu kita mungkin harus membuat proses pendekatan gray level citra keluaran ke nilai bulat.
- Dengan demikian fungsi transformasi diskrit tidak dapat menjamin pemetaan relasi yang one-to-one.

3.12. Histogram Spesifik

Histogram ekualisasi memiliki kelemahan yaitu dapat menghasilkan hanya satu jenis output citra. Dengan histogram spesifik, kita dapat menentukan bentuk histogram yang kita inginkan dimiliki oleh citra output. Histogram spesifik tidak mempunyai histogram yang seragam (uniform).

Prosedur histogram spesifik :

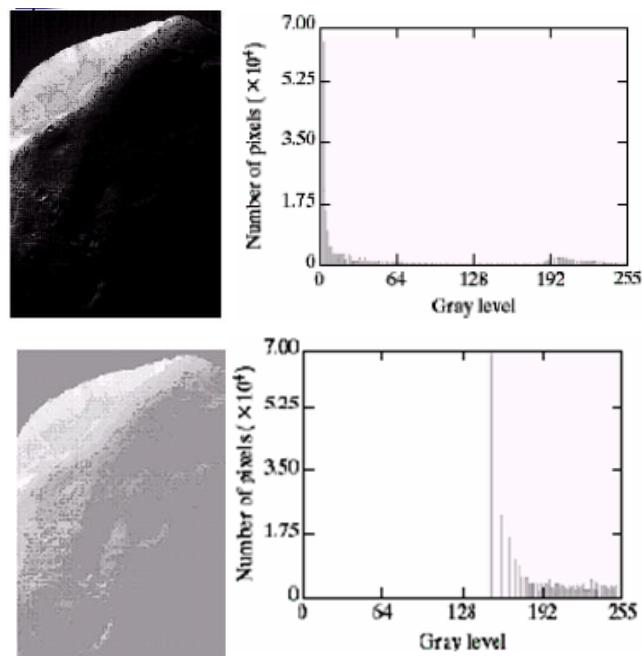
1. Dapatkan fungsi transformasi $T(r)$ dengan menghitung histogram ekualisasi dari citra input

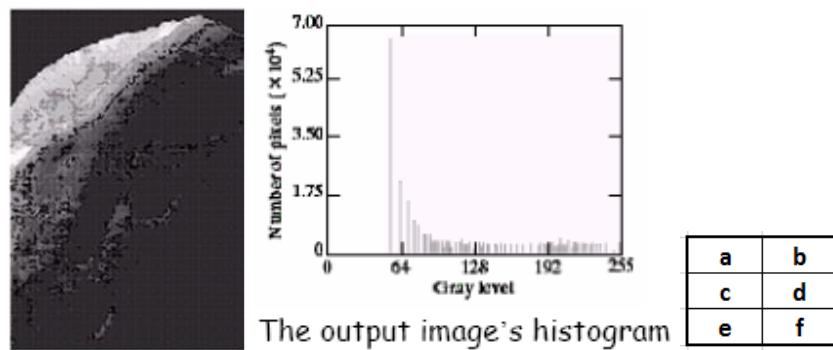
$$s = T(r) = \int_0^r p_r(w) dw$$

2. Dapatkan fungsi transformasi $G(z)$ dengan menghitung histogram ekualisasi dari fungsi kepadatan yang dikehendaki

$$\underline{g(z) = \int_0^z p_z(t) dt = s}$$

3. Dapatkan fungsi transformasi inverse $G^{-1}(s) = G^{-1}[T(r)]$
4. Dapatkan citra output dengan menerapkan proses gray level dari fungsi transformasi inverse ke semua piksel pada citra input



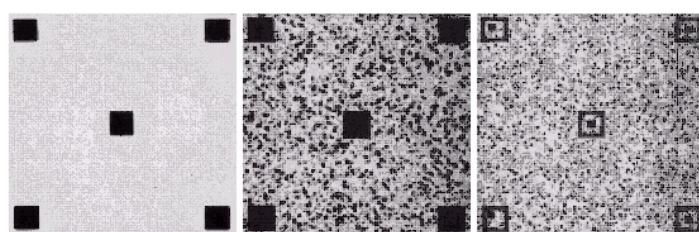


Gambar 3.27. Perbandingan Citra Output dan Histogram Ekualisasi

Gambar 3.27. bagian a) menunjukkan citra asal mars, dan histogram nya ditampilkan di gambar 3.27. b) tampak citra didominasi warna gelap konsentrasi piksel pada gray level gelap, sisi kiri. Gambar 3.27. bagian c) adalah citra hasil histogram ekualisasi begitu pula bagian d) adalah histogram ekualisasinya tampak histogram ekualisasi tidak membuat citra output terlihat lebih baik daripada citra asal. Metode histogram ekualisasi memapping range yang sempit dari piksel gelap ke gray scale terakhir dari citra output sehingga konsekuensinya citra output akan terang dan tampilan citra menjadi wash-out. Gambar 3.27. bagian e) adalah citra hasil histogram ekualisasi spesifik dan bagian f) adalah histogram ekualisasinya Perhatikan bahwa output histogram pada daerah gelap bergeser ke arah kanan daerah yang lebih terang dengan gray scale yang diinginkan.

3.13. Peningkatan Mutu Citra Secara Lokal

Peningkatan mutu citra dapat dilakukan secara global dan lokal, dicontohkan di gambar 3.28. bagian a) adalah citra asal (sedikit blur untuk mengurangi noise) bagian b) adalah citra hasil peningkatan mutu global dengan histogram ekualisasi (meningkatkan noise dan sedikit meningkatkan kontras namun susunannya tidak berubah) bagian c) adalah citra hasil peningkatan mutu secara lokal dengan histogram ekualisasi menggunakan 7×7 ketetanggaan (menyingkapkan kotak kecil di dalam yang lebih besar dari citra asal).



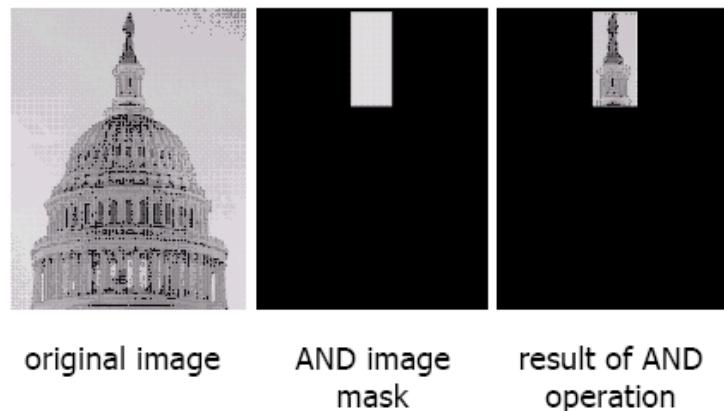
Gambar 3.28. Contoh Perbandingan Citra Hasil Peningkatan Mutu (Global dan Lokal)

3.14. Peningkatan Mutu Citra dengan Operasi Logika dan Aritmatika

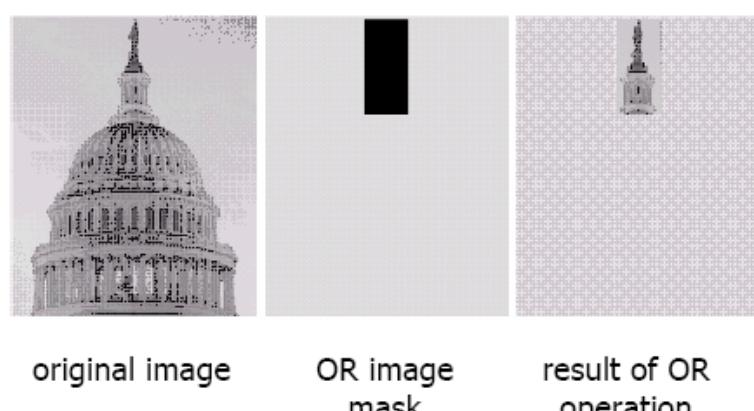
Operasi aritmatika atau logika dilakukan berbasis pada piksel demi piksel dasar di antara dua atau lebih citra. Kecuali pada operasi logika NOT yang dijalankan hanya pada satu citra.

3.14.1. Operasi Logika

Operasi logika pada citra dilakukan pada gray level citra, dimana nilai-nilai piksel yang diproses dalam bentuk bilangan biner. Terang merepresentasikan bit 1, dan gelap mewakili bit 0. Operasi NOT sama dengan transformasi negatif. Contoh output citra hasil operasi logika pada gambar 3.29 dan gambar 3.30.



Gambar 3.29. Contoh Operasi AND

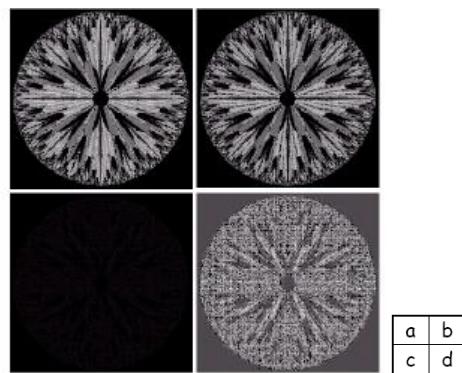


Gambar 3.30. Contoh Operasi OR

3.14.2. Operasi Aritmatika Pengurangan Citra

Formula yang digunakan : $\mathbf{g(x,y) = f(x,y) - h(x,y)}$

Pengurangan citra bertujuan jika kita ingin mengambil bagian tertentu saja dari citra, meningkatkan mutu dimana adanya perbedaan diantara citra. Contoh pengurangan citra pada gambar 3.31 dan gambar 3.32.

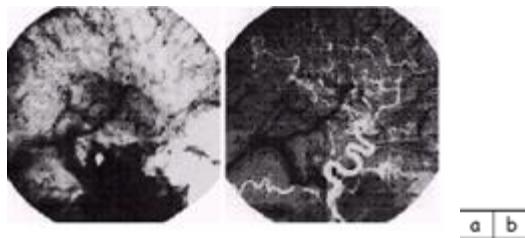


Gambar 3.31. Contoh Pengurangan Citra Fraktal

- a). Citra asal fraktal
- b). Citra hasil pengaturan empat bit plane ke nol
 - merujuk pada bit plane slicing
 - bit plane yang lebih tinggi berkontribusi memberikan detail yang signifikan
 - bit plane yang lebih rendah berkontribusi pada detail yang lebih bagus
 - citra b). hampir identik secara visual dengan citra a), dengan sangat sedikit penurunan dalam kontras keseluruhan karena variabilitas gray level pada citra kurang.
- c). Perbedaan antara a). dan b) (hampir hitam)
- d). Histogram ekualisasi dari citra c). (lakukan transformasi kontras stretching)

Penerapan pengurangan citra pada mask radiografi :

- $h(x, y)$ adalah mask, sebuah citra X-ray daerah tubuh pasien yang diambil oleh kamera TV intensif (tradisional film X-ray) yang terletak diseberang X-ray sumber
- $f(x, y)$ adalah citra X-ray yang diambil setelah injeksi kontras menengah ke darah pasien
- Citra diambil di TV, sehingga dokter dapat melihat bagaimana menjalar melalui media berbagai arteri di daerah yang sedang diamati (efek dari pengurangan) ditunjukkan dalam mode film.

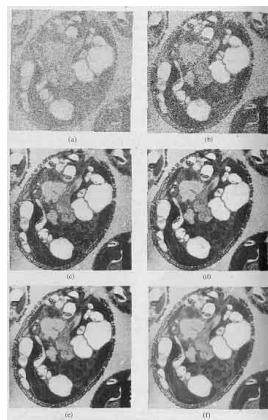


Gambar 3.32. Contoh Pengurangan Citra dengan Mask Radiografi

Gambar 3.32. bagian a) menunjukkan citra mask, dan bagian b) citra yang diambil setelah injeksi dengan medium kontras (iodine) pada aliran darah) dengan pengurangan citra mask. Dapat dilihat latar belakang gelap karena tidak ada perubahan yang berarti pada dua citra, perbedaan area ditandai dengan warna terang karena mempunyai perbedaan yang besar.

3.14.3. Operasi Aritmatika Averaging Citra

Dilakukan jika kita memiliki beberapa citra yang bergambar sama, namun semua citra memiliki noise (gangguan). Noise satu citra berbeda dengan noise citra lainnya (tidak berkorelasi). Cara memperbaikinya adalah dengan melakukan operasi rata-rata terhadap semua citra tersebut.



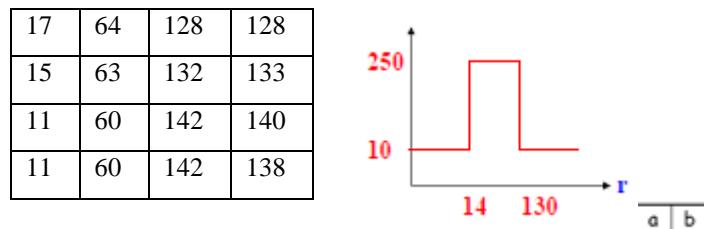
Gambar 3.33. Contoh Averaging Citra

3.15. Ringkasan

1. Peningkatan mutu citra pada domain spatial ada dua proses yaitu pemrosesan piksel, dan pemrosesan masking.
2. Pemrosesan piksel metodenya antara lain dengan kontras stretching, thresholding, transformasi gray level, gray level slicing, transformasi piecewise linier, bit plane slicing, pemrosesan histogram, histogram ekualisasi, histogram pemrosesan lokal, operasi aritmatika atau logika pada citra.
3. Pemrosesan masking akan dibahas di bab 5 Filtering.

3.16. Latihan

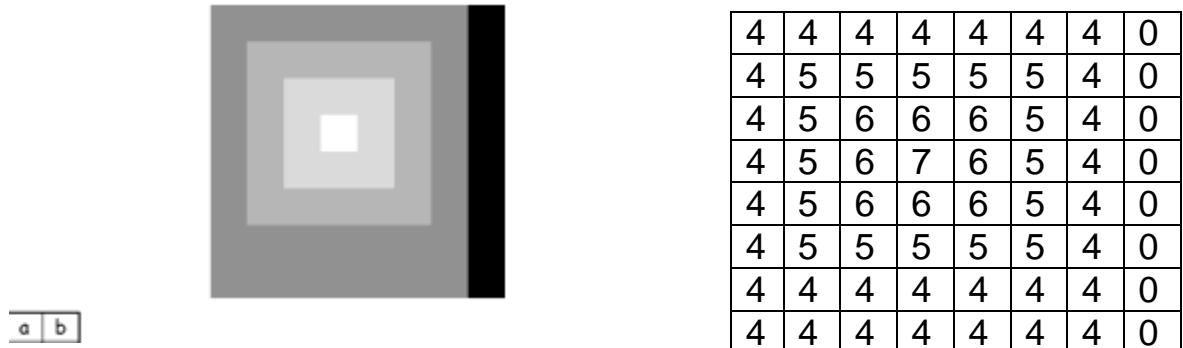
1. Diketahui citra gray level 4×4 pada gambar 3.34 a, lakukan transformasi pada setiap piksel mengacu pada fungsi trasformasi gray level pada gambar 3.34 b. Dapatkan citra output hasil trasformasi gray levelnya.



Gambar 3.34. Citra Input Gray Level 4×4 dan Fungsi Transformasi

2. Dengan citra input gambar 3.34 a., dapatkan citra output bit plane pada bit ke 7
3. Diketahui citra input gray level 8×8 dengan range $[0 - 7]$ pada gambar 3.35. a dan gambar 3.35 b gambarkan histogram citra input. Lakukan ekualisasi buat

tabelnya, dan dapatkan citra output hasil ekualisasi dan buat pula histogram citra hasil ekualisasi.



Gambar 3.35. Citra Input Gray Level 8x8 dan Matrik Citra Gray Levelnya

BAB 4

TRANFORMASI FOURIER

POKOK BAHASAN:

- ✓ Dasar-Dasar Transformasi Fourier
- ✓ Transformasi Fourier 1D
- ✓ Transformasi Fourier 2D
- ✓ Transformasi Fourier Diskrit
- ✓ Fast Fourier Transform (FFT)

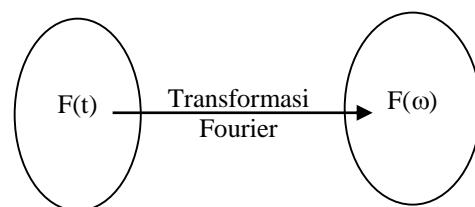
TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat:

- ✓ Mengerti dasar-dasar transformasi fourier
- ✓ Mengerjakan transformasi fourier 1D dan 2D diskrit dan FFT

4.1. Dasar-dasar Transformasi Fourier

Transformasi Fourier adalah suatu model transformasi yang memindahkan domain spasial atau domain waktu menjadi domain frekwensi.



Gambar 4.1. Transformasi Fourier

Transformasi Fourier merupakan suatu proses yang banyak digunakan untuk memindahkan domain dari suatu fungsi atau obyek ke dalam domain frekwensi. Di dalam pengolahan citra digital, transformasi fourier digunakan untuk mengubah domain spasial pada citra menjadi domain frekwensi. Analisa-analisa dalam domain frekwensi banyak digunakan seperti filtering. Dengan menggunakan transformasi fourier, sinyal atau citra dapat dilihat sebagai suatu obyek dalam domain frekwensi.

4.1.1. Transformasi Fourier 1D

Transformasi Fourier kontinu 1D dari suatu fungsi waktu $f(t)$ didefinisikan dengan:

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt$$

dimana

$F(\omega)$ adalah fungsi dalam domain frekwensi

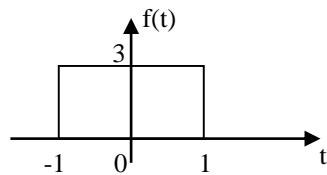
ω adalah frekwensi radial $0 - 2\pi f$,

atau dapat dituliskan bahwa :

$$\omega = 2\pi f$$

Contoh 4.1.

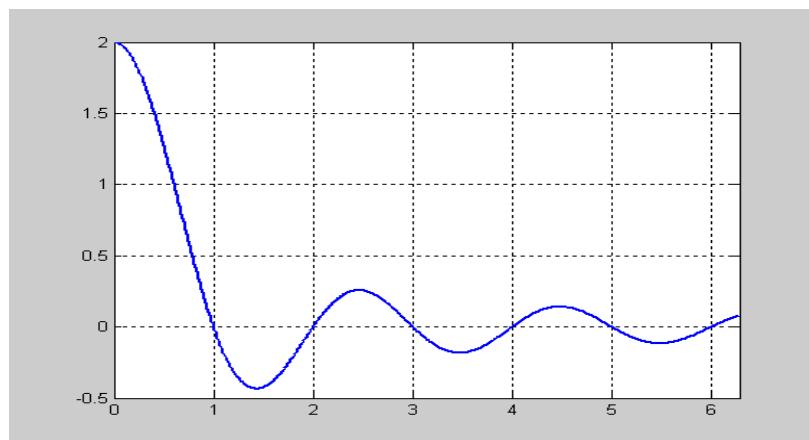
Diketahui fungsi $f(t)$ sebagai berikut:



Transformasi Fourier dari $f(t)$ di atas adalah:

$$\begin{aligned} F(\omega) &= \int_{-1}^1 (1) e^{-j\omega t} dt = \int_{-1}^1 e^{-j\omega t} dt \\ &= -\frac{1}{j\omega} e^{-j\omega t} \Big|_{-1}^1 \\ &= -\frac{1}{j\omega} [e^{-j\omega} - e^{j\omega}] = \frac{2 \sin(\omega)}{\omega} \end{aligned}$$

Hasil dari transformasi Fourier untuk $\omega = 0$ s/d $2\pi f$ adalah :



Gambar 4.2. Contoh hasil transformasi fourier

4.1.2. Transformasi Fourier 2D

Transformasi Fourier kontinu 2D dari suatu fungsi spasial $f(x,y)$ didefinisikan dengan:

$$F(\omega_1, \omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j(\omega_1 x + \omega_2 y)} dx dy$$

dimana

$F(\omega_1, \omega_2)$ adalah fungsi dalam domain frekwensi

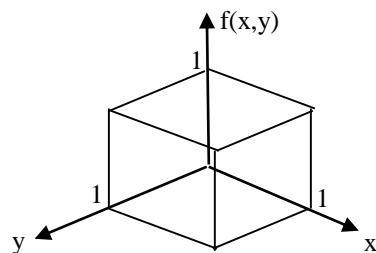
$f(x, y)$ adalah fungsi spasial atau citra

ω_1 dan ω_2 adalah frekwensi radial $0 - 2\pi f$

Transformasi fourier yang digunakan dalam pengolahan citra digital adalah transformasi fourier 2D.

Contoh 4.2.

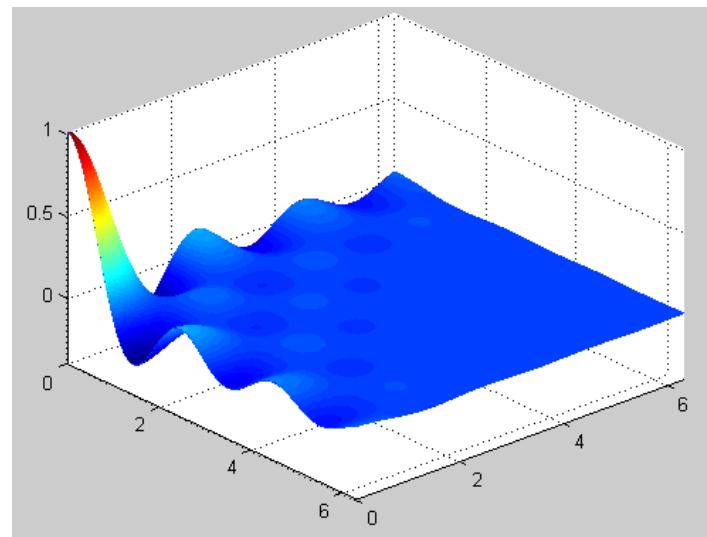
Diketahui fungsi spasial $f(x, y)$ berikut:



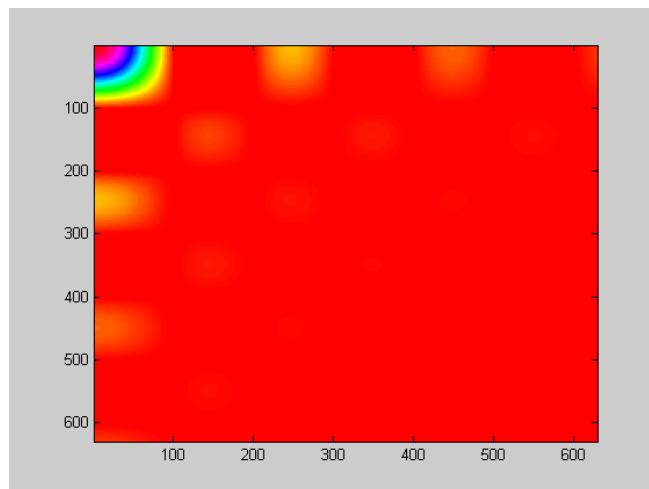
Transformasi fourier dari $f(x, y)$ di atas adalah:

$$\begin{aligned} F(\omega_1, \omega_2) &= \int_{-1}^{1} \int_{-1}^{1} (1) e^{-j(\omega_1 x + \omega_2 y)} dy dx \\ &= \int_{-1}^{1} \left[-\frac{e^{-j\omega_1 x}}{j\omega_2} e^{-j\omega_2 y} \right]_{-1}^{1} dx = \int_{-1}^{1} \frac{\sin(\omega_2)}{\omega_2} e^{-j\omega_1 x} dx \\ &= \frac{\sin(\omega_2)}{\omega_2} \left[-\frac{e^{-j\omega_1 x}}{j\omega_1} \right]_{-1}^{1} = \frac{\sin(\omega_2)}{\omega_2} \cdot \frac{\sin(\omega_1)}{\omega_1} \\ &= \frac{\sin(\omega_2) \sin(\omega_1)}{\omega_2 \omega_1} \end{aligned}$$

Hasil dari transformasi fourier untuk $0 < \omega_1, \omega_2 < 2\pi f$, adalah sebagai berikut :



Gambar 4.3. Contoh hasil transformasi fourier 2D



Gambar 4.4. Hasil transformasi fourier dalam surface

Transformasi Fourier semacam ini disebut dengan *continuous fourier transform*, dan sulit dikomputasi karena ada operasi integral dan sifat kontinunya itu sendiri.

4.2. Transformasi Fourier Diskrit

Transformasi fourier diskrit atau disebut dengan Discrete Fourier Transform (DFT) adalah model transformasi fourier yang dikenakan pada fungsi diskrit, dan hasilnya juga diskrit. DFT didefinisikan dengan :

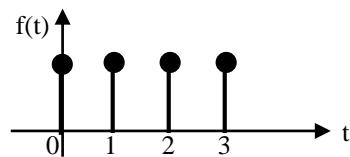
$$F(k) = \sum_{n=1}^N f(n) e^{-j2\pi knT/N}$$

4.2.1. DFT 1D

DFT seperti rumus di atas dinamakan dengan DFT 1 dimensi, DFT semacam ini banyak digunakan dalam pengolahan sinyal digital.

Contoh 4.3 :

Diketahui $f(t)$ dalam bentuk diskrit $f(n)$ sebagai berikut :



DFT dengan $T=1$ dari fungsi $f(n)$ di atas adalah :

$$\begin{aligned} k=0 \rightarrow F(0) &= \sum_{n=0}^3 f(n) e^{-jn0} = \sum_{n=0}^3 f(n) \\ &= 1 + 1 + 1 + 1 = 4 \end{aligned}$$

$$k=1 \rightarrow F(1) = \sum_{n=0}^3 f(n) e^{-j4\pi n/4} = \sum_{n=0}^3 f(n) e^{-jn\pi} = 0$$

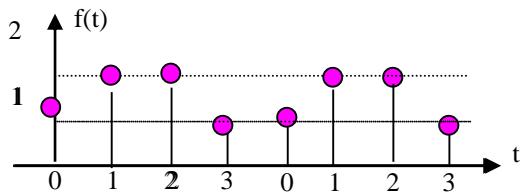
$$k=2 \rightarrow F(2) = \sum_{n=0}^3 f(n) e^{-j8n/4} = \sum_{n=0}^3 f(n) e^{-j2n\pi} = 0$$

$$k=3 \rightarrow F(3) = \sum_{n=0}^3 f(n) e^{-j12n/4} = \sum_{n=0}^3 f(n) e^{-j3n\pi} = 0$$

Hasil dari DFT untuk T (periode sampling) yang berbeda akan juga berbeda. Sehingga dalam proses perhitungan DFT, penentuan nilai T juga merupakan perhatian penting. Sebagai acuan dapat digunakan aturan frekwensi Niquist bahwa frekwensi sampling minimal dua kali frekwensi informasi (data), atau dengan kata lain periode sampling maksimal setengah kali periode dari nilai fungsinya.

Contoh 4.3 :

Diketahui $f(t)$ dalam bentuk diskrit $f(n)$ sebagai berikut :



DFT dengan $T=1$ dari fungsi $f(n)$ di atas adalah :

$$F(k) = \sum_{n=0}^7 f(n) e^{-j2\pi nk/8} = \sum_{n=0}^7 f(n) e^{-j\pi nk/8}$$

Hasil DFT fungsi $f(t)$ di atas adalah :

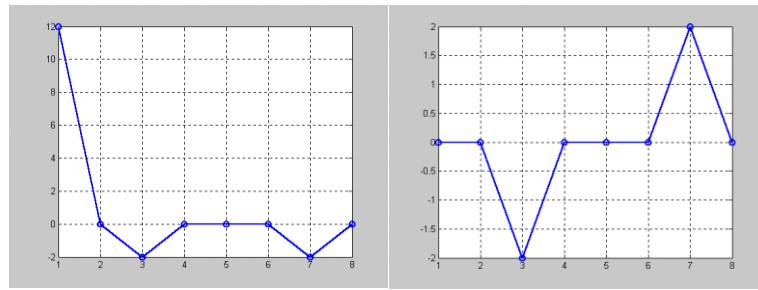
k	F(k)
0	12
1	0
2	-2 - 2j
3	0
4	0
5	0
6	-2 + 2j
7	0

Terlihat bahwa hasil dari DFT adalah bilangan komplek, yang terdiri dari unsur real dan imaginer. Sehingga dapat dipisahkan dalam unsur real dan imaginer sebagai berikut :

k	Real{F(k)}	Im{F(k)}
0	12	0
1	0	0
2	-2	-2
3	0	0
4	0	0

k	Real{F(k)}	Im{F(k)}
5	0	0
6	-2	2
7	0	0

Dan dapat digambarkan sebagai berikut :



Bagian Real

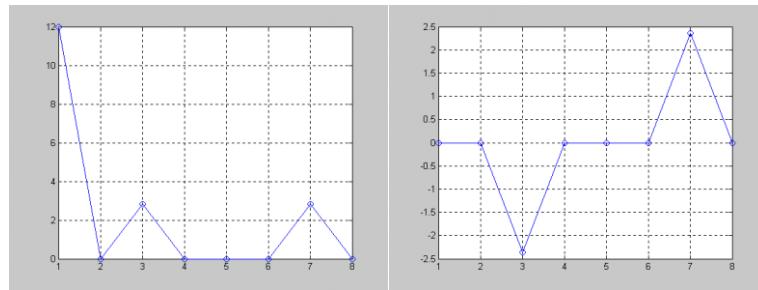
Bagian Imaginer

Gambar 4.5. Contoh DFT real dan imaginer

Atau dapat dinyatakan dalam *magnitude* dan *phase* dengan definisi sebagai berikut :

Magnitude : $|F(k)| = \sqrt{(\text{Re}\{f(k)\})^2 + (\text{Im}\{f(k)\})^2}$

Phase : $\text{Arg}\{F(k)\} = \frac{\text{Im}\{F(k)\}}{\text{Re}\{F(k)\}}$



Magnitude

Phase

Gambar 4.6. Contoh DFT real dan imaginer

Bila DFT dihitung untuk k=0 s/d 15 maka hasilnya adalah:

k	F(k)	K	F(k)
0	12	8	12
1	0	9	0
2	-2 - 2j	10	-2 - 2j

k	F(k)	K	F(k)
3	0	11	0
4	0	12	0
5	0	13	0
6	-2 + 2j	14	-2 + 2j
7	0	15	0

Terlihat terjadi pengulangan hasil, hal ini disebabkan proses DFT memang mengakibatkan terjadinya periodik. Ini sebagai akibat dari adanya unsur radial 2D dalam bentuk transformasi fourier. Sehingga dalam proses perhitungan DFT, perhitungan cukup dilakukan sampai 1/2 periodik saja. Dan perhitungan inilah yang dinamakan dengan FFT (Fast Fourier Transform).

4.2.2. Transformasi Fourier Diskrit 2D

Transformasi Fourier Diskrit (DFT) 2 Dimensi adalah transformasi fourier diskrit yang dikenakan pada fungsi 2D (fungsi dengan dua variabel bebas), yang didefinisikan sebagai berikut :

$$F(k_1, k_2) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} f(n_1, n_2) e^{-j2\pi T(k_1 n_1 / N_1 + k_2 n_2 / N_2)}$$

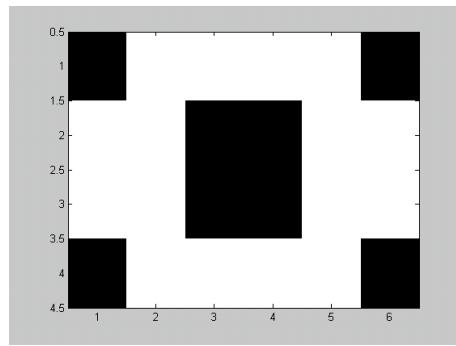
DFT 2D ini banyak digunakan dalam pengolahan citra digital, karena data citra dinyatakan sebagai fungsi 2D.

Contoh 4.4 :

Diketahui $f(x,y)$ adalah sebagai berikut :

0	1	1	1	1	0
1	1	0	0	1	1
1	1	0	0	1	1
0	1	1	1	1	0

Bila digambarkan hasilnya adalah sebagai berikut :



Gambar 4.7. Contoh citra dalam $f(x,y)$

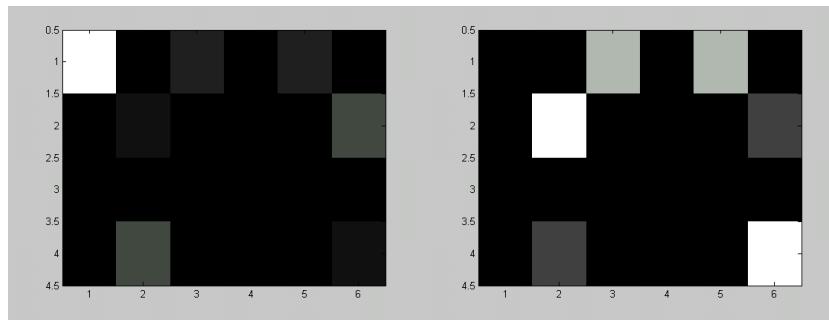
DFT dari fungsi $f(x,y)$ di atas adalah :

$$F(k_1, k_2) = \sum_{n_1=0}^4 \sum_{n_2=0}^6 f(n_1, n_2) e^{-j2\pi T(k_1 n_1 / 4 + k_2 n_2 / 6)}$$

Hasil dari DFT adalah sebagai berikut :

16	0	-2 - 3.46i	0	-2 + 3.46i	0
0	-1.27 - 4.73i	0	0	0	4.73 - 1.27i
0	0	0	0	0	0
0	-4.73 + 1.27i	0	0	0	1.27 + 4.73i

Secara Grafis dapat ditunjukkan bahwa :



Bagian Real

Bagian Imaginer

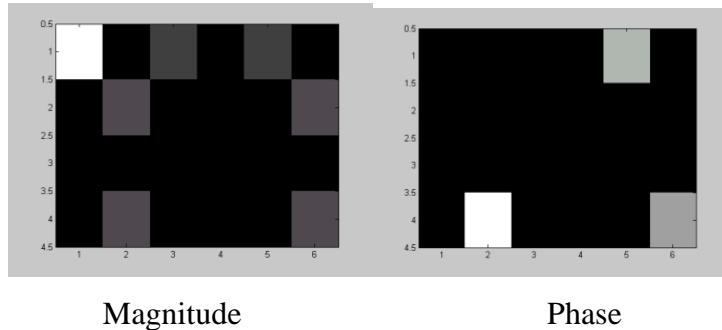
Gambar 4.8. Contoh hasil DFT 2D

Hasil DFT dalam bentuk magnitude dan phase adalah sebagai berikut :

```
Magnitude =
 16.0000   0   4.0000   0   4.0000   0
      0   4.8990   0   0   0   4.8990
      0   0   0   0   0   0
      0   4.8990   0   0   0   4.8990

Phase =
      0   0   -2.0944   0   2.0944   0
      0  -1.8326   0   0   0  -2.8798
      0   0   0   0   0   0
      0   2.8798   0   0   0   1.8326
```

Secara grafis dapat digambarkan sebagai berikut :



Gambar 4.9. Contoh hasil DFT 2D dalam magnitude dan phase

4.3. Fast Fourier Transform

FFT (Fast Fourier Transform) adalah teknik perhitungan cepat dari DFT. Untuk pembahasan FFT ini, akan dijelaskan FFT untuk 1D dan FFT untuk 2D. Dimana FFT 2D adalah pengembangan dari DFT 2D.

4.3.1. FFT 1D

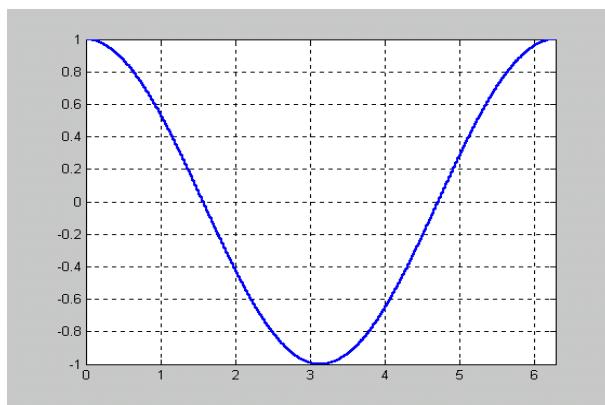
FFT adalah DFT dengan teknik perhitungan yang cepat dengan memanfaatkan sifat periodikal dari transformasi fourier. Perhatikan definisi dari DFT :

$$F(k) = \sum_{n=1}^N f(n) e^{-j2\pi knT/N}$$

Atau dapat dituliskan dengan :

$$F(k) = \sum_{n=1}^N f(n) \cos(2\pi nkT/N) - j \sum_{n=1}^N f(n) \sin(2\pi nkT/N)$$

Perhatikan fungsi cosinus berikut ini :



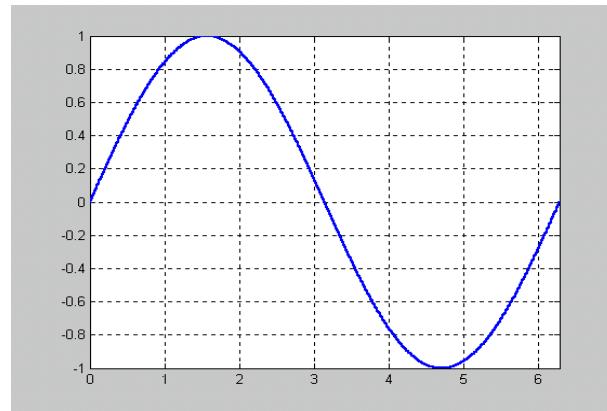
Gambar 4.10. Gambar fungsi cosinus 1 periode

Pada gambar tersebut, dapat dilihat bahwa nilai fungsi cosinus untuk setengah bagian bila dilihat dari kiri dan setengah bagian dari kanan akan sama, atau dapat dikatakan bahwa

nilai fungsi cosinus untuk setengah periode adalah kebalikan horisontal (shift) dari nilai setengah periode sebelumnya, atau dapat dituliskan bahwa ;

$$\cos(T/2-x) = -\cos(x), \text{ untuk } 0 < x < T/2$$

Perhatikan fungsi sinus berikut ini :



Gambar 4.11. Gambar fungsi sinus 1 periode

Pada gambar tersebut, dapat dilihat bahwa nilai fungsi sinus untuk setengah periode adalah kebalikan dari nilai setengah periode sebelumnya, atau dapat dituliskan bahwa ;

$$\sin(x+T/2) = -\sin(x), \text{ untuk } 0 < x < T/2$$

Dari kedua sifat di atas, maka perhitungan DFT dapat disederhanakan dengan cukup menghitung setengah periode saja, sedangkan setengah periode berikutnya dapat dihitung dengan menggunakan :

$$F(x+T/2) = \text{Real}\{F(T/2-x)\} - j \text{Im}\{F(x)\}$$

Contoh 4.5:

Perhatikan pada contoh 4.3, hasil dari DFT adalah sebagai berikut :

k	F(k)	k	F(k)
0	12	5	0
1	0	6	-2 + 2j
2	-2 - 2j	7	0
3	0	8	12
4	0		

Untuk menghitung hasil tersebut diperlukan $8 \times 8 = 64$ kali perhitungan (*looping*). Bila dihitung dengan menggunakan FFT seperti diatas, maka akan diperoleh $(5x + 4) = 39$

kali perhitungan, hal ini terlihat sangat menghemat perhitungan. Tentunya dengan bertambahnya ukuran data, maka perbedaan kecepatan perhitungannya akan semakin terasa.

Untuk n buah data, DFT memerlukan n^2 kali perhitungan, dan FFT memerlukan $(n/2 + 1)xn + n/2$ kali perhitungan. Misalkan jumlah data $n=100$ maka dengan menggunakan DFT diperlukan $100 \times 100 = 10.000$ kali perhitungan, dengan menggunakan FFT cukup dilakukan $(51 \times 100 + 50) = 5150$ kali perhitungan.

Catatan :

Proses FFT di atas adalah proses DFT dengan dibagi menjadi 2 bagian, sebenarnya hal ini masih bisa dilanjutkan dengan memperhatikan kembali sifat dari fungsi sinus dan cosinus pada setengah bagian, seperempat bagian dan seterusnya. Sehingga prosesnya menjadi lebih cepat.

Cara perhitungan FFT adalah sebagai berikut.

(1) Hitung dengan cara DFT

$$\text{Untuk } k=0 : F(0) = 12 + 0j$$

$$\text{Untuk } k=1 : F(1) = 0 + 0j$$

$$\text{Untuk } k=2 : F(2) = -2 - 2j$$

$$\text{Untuk } k=3 : F(3) = 0 + 0j$$

$$\text{Untuk } k=4 : F(4) = 0 + 0j \text{ (karena posisi ditengah)}$$

(2) Perhitungan selanjutnya dilakukan dengan menggunakan cara *konjugate* diatas.

$$\text{Untuk } k=5 : F(5) = \text{Real}\{F(3)\} - \text{Im}\{F(1)\} = 0 + 0j$$

$$\text{Untuk } k=6 : F(5) = \text{Real}\{F(2)\} - \text{Im}\{F(2)\} = -2 + 2j$$

$$\text{Untuk } k=7 : F(5) = \text{Real}\{F(1)\} - \text{Im}\{F(3)\} = 0 + 0j$$

4.3.2. FFT 2D

FFT 2D adalah DFT 2D dengan teknik perhitungan yang cepat dengan memanfaatkan sifat periodikal dari transformasi fourier. Seperti halnya FFT 1D, maka dengan menggunakan sifat fungsi sinus dan cosinus, algoritma dari FFT 2D ini adalah :

- (1) Hitung FFT 2D untuk $n_1 = 1$ s/d $N_1/2$ dan $n_2 = 1$ s/d $N_2/2$ menggunakan rumus DFT.
- (2) Untuk selanjutnya digunakan teknik konjugate 2D.

Contoh 4.6 :

Perhatikan contoh 4.4. dengan fungsi $f(x,y)$ sebagai berikut :

0	1	1	1	1	0
1	1	0	0	1	1
1	1	0	0	1	1
0	1	1	1	1	0

DFT dari fungsi ini adalah sebagai berikut :

16	0	-2 - 3.46i	0	-2 + 3.46i	0
0	-1.27 - 4.73i	0	0	0	4.73 - 1.27i
0	0	0	0	0	0
0	-4.73 + 1.27i	0	0	0	-1.27 + 4.73i

Jumlah perhitungan DFT di atas adalah $(4 \times 6)^2 = 576$ kali.

Dengan menggunakan FFT, maka perhitungannya adalah :

(1) Hitung DFT untuk ukuran 4x4 sehingga diperoleh :

$$\begin{array}{ll}
 F(0,0) = 16 & F(0,1) = 0 \\
 F(0,2) = -2 - 3.46i & F(0,3) = 0 \\
 F(1,0) = 0 & F(1,1) = -1.27 - 4.73i \\
 F(1,2) = 0 & F(1,3) = 0 \\
 F(2,0) = 0 & F(2,1) = 0 \\
 F(2,2) = 0 & F(2,3) = 0 \\
 F(3,0) = 0 & F(3,1) = -4.73 + 1.27i \\
 F(3,2) = 0 & F(3,3) = 0
 \end{array}$$

(2) Dengan cara konjugate dapat diperoleh :

$$\begin{aligned}
 F(0,4) &= \text{conjugate}(F(0,2)) = -2 + 3.46i \\
 F(0,5) &= \text{conjugate}(F(0,1)) = 0 \\
 F(1,4) &= \text{conjugate}(F(3,2)) = 0 \\
 F(1,5) &= \text{conjugate}(F(3,1)) = -4.73 - 1.27i \\
 F(2,4) &= \text{conjugate}(F(2,2)) = 0 \\
 F(2,5) &= \text{conjugate}(F(2,1)) = 0 \\
 F(2,4) &= \text{conjugate}(F(1,2)) = 0 \\
 F(2,5) &= \text{conjugate}(F(1,1)) = -1.27 + 4.73i
 \end{aligned}$$

Jumlah perhitungan dari FFT adalah $(4 \times 4 \times 4 \times 6) + 2 \times 4 = 392$ kali.

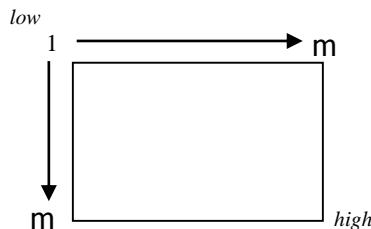
Catatan :

Jumlah perhitungan dari FFT untuk fungsi 2D yang berukuran $m \times n$ adalah $(m/2+1) \times (n/2+1) \times m \times n + m \times n / 4$

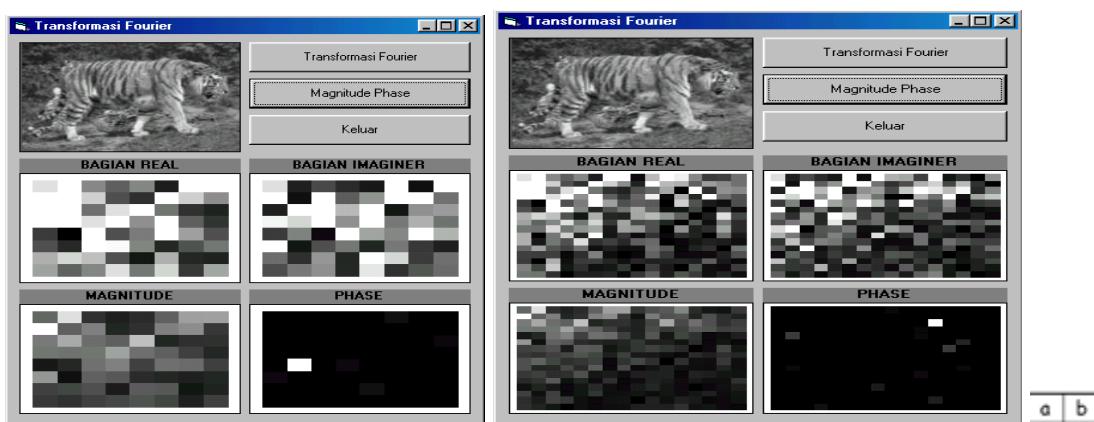
Pada beberapa citra program di atas menyebabkan hasilnya hanya memperlihatkan suatu titik putih di pojok kiri atas atau dipojok kanan bawah pada gambar magnitudenya. Hal ini tentunya membuat lebih sulit dalam melakukan analisa terhadap frekwensi suatu

citra. Hal ini disebabkan terjadinya nilai dominan pada suatu frekwensi. Untuk mengurangi dominansi tersebut digunakan fungsi logaritma yang dikenakan pada magnitude.

Dengan memperhatikan bahwa posisi kiri atas menyatakan frekwensi rendah dan posisi kanan bawah menyatakan frekwensi tinggi seperti format gambar berikut :



Gambar 4.12. Format koordinat frekwensi



Gambar 4.13. Contoh transformasi fourier

Perhatikan nilai magnitude dari gambar 4.13 a) dengan 8x8 dan gambar b) dengan 16x16, maka akan terlihat bahwa nilai pada daerah frekwensi rendah secara rata-rata lebih tinggi dari nilai-nilai pada frekwensi tinggi. Dari hal ini dapat dikatakan bahwa citra bekerja di frekwensi rendah. Sehingga bila frekwensi rendah ini dihilangkan maka gambarnya akan hilang, tetapi bila frekwensi tinggi yang dihilangkan maka hanya sedikit bagian citra yang hilang. Pada proses filtering citra, beberapa keperluan yang mempertahankan citra lebih banyak menggunakan Low Pass Filter (Filter yang mengambil frekwensi rendah dan membuang frekwensi tinggi) seperti reduksi noise, proses blur dan lain-lain. Meskipun demikian terkadang diperlukan proses High Pass Filter (Filter yang membuang frekwensi rendah dan mengambil frekwensi tinggi) seperti deteksi tepi.

4.5. Transformasi Cosinus Diskrit

Transformasi Cosinus Diskrit atau disebut dengan *Discrete Cosine Transform* (DCT) adalah model transformasi fourier yang dikenakan pada fungsi diskrit dengan hanya mengambil bagian cosinus dari eksponensial kompleks, dan hasilnya juga diskrit. DCT didefinisikan dengan :

$$F(k) = \sum_{n=1}^N f(n) \cos(2\pi n k / N)$$

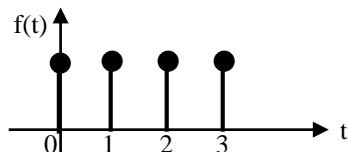
Berbeda dengan DFT yang hasilnya berupa variabel kompleks dengan bagian real dan imaginer, maka hasil DCT hanya berupa real tanpa ada imaginer. Hal ini banyak membantu karena dapat mengurangi perhitungan. Dalam DCT nilai magnitude adalah hasil dari DCT itu sendiri dan tidak diperlukan phase.

4.5.1. DCT 1D

DCT seperti rumus di atas dinamakan dengan DCT 1 dimensi, DCT semacam ini banyak digunakan dalam pengolahan sinyal digital.

Contoh 4.7 :

Diketahui $f(t)$ dalam bentuk diskrit $f(n)$ sebagai berikut :

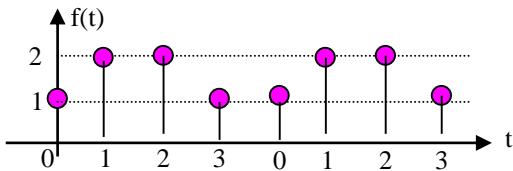


DFT dengan $T=1$ dari fungsi $f(n)$ di atas adalah :

$$\begin{aligned} k=0 \rightarrow F(0) &= \sum_{n=0}^3 f(n) \cos(0) = \sum_{n=0}^3 f(n) \\ &= 1 + 1 + 1 + 1 = 4 \\ k=1 \rightarrow F(1) &= \sum_{n=0}^3 f(n) \cos(4n\pi/4) = \sum_{n=0}^3 f(n) \cos(n\pi) = 0 \\ k=2 \rightarrow F(2) &= \sum_{n=0}^3 f(n) \cos(8n\pi/4) = \sum_{n=0}^3 f(n) \cos(2n\pi) = 0 \\ k=3 \rightarrow F(3) &= \sum_{n=0}^3 f(n) \cos(12n\pi/4) = \sum_{n=0}^3 f(n) \cos(3n\pi) = 0 \end{aligned}$$

Contoh 4.3 :

Diketahui $f(t)$ dalam bentuk diskrit $f(n)$ sebagai berikut :



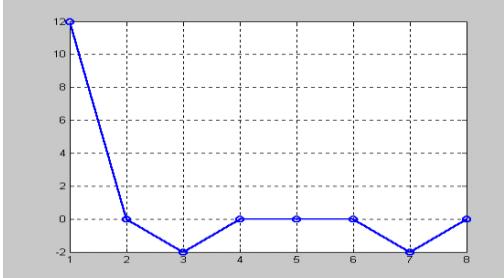
DCT dengan $T=1$ dari fungsi $f(n)$ di atas adalah :

$$F(k) = \sum_{n=0}^7 f(n) \cos(2\pi k n / 8) = \sum_{n=0}^7 f(n) \cos(\pi k n / 4)$$

Hasil DCT fungsi $f(t)$ di atas adalah :

k	F(k)
0	12
1	0
2	-2
3	0
4	0
5	0
6	-2
7	0

Secara grafik dapat digambarkan hasil DCT seperti gambar berikut :



Gambar 4.14. Contoh hasil DCT

4.5.2. DCT 2D

DCT 2 dimensi adalah transformasi fourier diskrit yang dikenakan pada fungsi 2D (fungsi dengan dua variabel bebas) dengan hanya mengambil bagian cosinus dari eksponensial imaginer, yang didefinisikan sebagai berikut :

$$F(k_1, k_2) = \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} f(n_1, n_2) \cos(2\pi k_1 n_1 / N_1) \cos(2\pi k_2 n_2 / N_2)$$

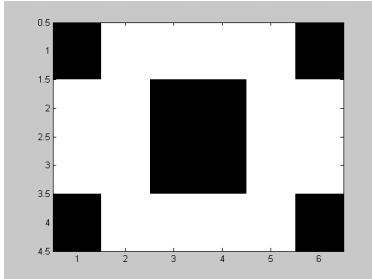
DCT 2D ini banyak digunakan dalam pengolahan citra digital, karena data citra dinyatakan sebagai fungsi 2D.

Contoh 4.8 :

Diketahui $f(x,y)$ adalah sebagai berikut :

0	1	1	1	1	0
1	1	0	0	1	1
1	1	0	0	1	1
0	1	1	1	1	0

Bila digambarkan hasilnya adalah sebagai berikut :



Gambar 4.15. Contoh citra dalam $f(x,y)$

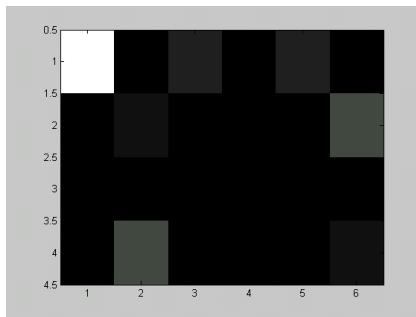
DCT dari fungsi $f(x,y)$ di atas adalah :

$$F(k_1, k_2) = \sum_{n_1=0}^4 \sum_{n_2=0}^6 f(n_1, n_2) \cdot \cos(2\pi n_1 k_1 / 4) \cdot \cos(2\pi n_2 k_2 / 6)$$

Hasil dari DFT adalah sebagai berikut :

16	0	-2	0	-2	0
0	-1.27	0	0	0	4.73
0	0	0	0	0	0
0	-4.73	0	0	0	1.27

Secara Grafis dapat ditunjukkan bahwa :



Gambar 4.16. Contoh hasil DCT 2D

4.6. Ringkasan

1. Transformasi Fourier lebih berguna daripada *fourier series* dalam masalah yang paling praktis karena dapat menangani sinyal durasi terbatas (*finite signal*).

2. Transformasi Fourier membawa kita berada diantara domain spatial dan domain frekuensi yang memungkinkan untuk representasi ganda sinyal untuk keperluan filtering dan analisis citra.
3. Transformasi fourier dapat memberikan informasi frekuensi dari sinyal 2D.
4. Tabel properti transformasi fourier 2D

Property	Expression(s)
Fourier transform	$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)}$
Inverse Fourier transform	$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M+vy/N)}$
Polar representation	$F(u, v) = F(u, v) e^{-j\phi(u, v)}$
Spectrum	$ F(u, v) = [R^2(u, v) + I^2(u, v)]^{1/2}, \quad R = \text{Real}(F) \text{ and } I = \text{Imag}(F)$
Phase angle	$\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]$
Power spectrum	$P(u, v) = F(u, v) ^2$
Average value	$\bar{f}(x, y) = F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$
Translation	$f(x, y) e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \Leftrightarrow F(u, v) e^{-j2\pi(ux_0/M+vy_0/N)}$ <p>When $x_0 = u_0 = M/2$ and $y_0 = v_0 = N/2$, then</p> $f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u+v}$
Conjugate symmetry	$F(u, v) = F^*(-u, -v)$ $ F(u, v) = F(-u, -v) $
Differentiation	$\frac{\partial^n f(x, y)}{\partial x^n} \Leftrightarrow (ju)^n F(u, v)$ $(-jx)^n f(x, y) \Leftrightarrow \frac{\partial^n F(u, v)}{\partial u^n}$
Laplacian	$\nabla^2 f(x, y) \Leftrightarrow -(u^2 + v^2) F(u, v)$
Distributivity	$\Im[f_1(x, y) + f_2(x, y)] = \Im[f_1(x, y)] + \Im[f_2(x, y)]$ $\Im[f_1(x, y) \cdot f_2(x, y)] \neq \Im[f_1(x, y)] \cdot \Im[f_2(x, y)]$
Scaling	$af(x, y) \Leftrightarrow aF(u, v), f(ax, by) \Leftrightarrow \frac{1}{ ab } F(u/a, v/b)$
Rotation	$x = r \cos \theta \quad y = r \sin \theta \quad u = \omega \cos \varphi \quad v = \omega \sin \varphi$ $f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$
Periodicity	$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$ $f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$
Separability	See Eqs. (4.6-14) and (4.6-15). Separability implies that we can compute the 2-D transform of an image by first computing 1-D transforms along each row of the image, and then computing a 1-D transform along each column of this intermediate result. The reverse, columns and then rows, yields the same result.

4.7. Latihan

1. Tuliskan definisi transformasi fourier pada citra, dan apa yang dapat diharapkan dengan melakukan transformasi fourier ini.
2. Pada citra-citra yang bergradiasi tinggi (mempunyai warna yang banyak), bagaimana hasil transformasi fouriernya ?
3. Pada citra-citra yang bergradiasi rendah (mempunyai warna yang sedikit atau berupa sketsa), bagaimana hasil transformasi fouriernya ?
4. Hitunglah transformasi fourier diskrit 2D dari data berikut ini :

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

5. Hitunglah transformasi cosinus dari data pada soal no 4. Apa perbedaan transformasi fourier dan transformasi cosinus ?
6. Dengan menggunakan program transformasi fourier yang sudah dibuat, gambarkan hasil transformasi fourier dari gambar-gambar berikut ini. Apa perbedaan hasil transformasi fourier dari keempat gambar di atas ?



Gambar 4.17. Contoh gambar untuk DFT dan DCT

7. Dengan menggunakan program transformasi cosinus yang sudah dibuat, gambarkan hasil transformasi cosinus dari gambar-gambar pada gambar 4.30 diatas. Apa perbedaan hasil transformasi cosinus dari keempat gambar di atas ?

BAB 5

FILTERING

POKOK BAHASAN:

- ✓ Prinsip-prinsip Filtering
- ✓ Konvolusi
- ✓ Low Pass Filter
- ✓ Noise Uniform
- ✓ Noise Gausian
- ✓ Noise Salt n Papper
- ✓ Noise Speckle
- ✓ High Pass Filter
- ✓ Operator Robert
- ✓ Operator Prewit
- ✓ Operator Sobel

TUJUAN BELAJAR:

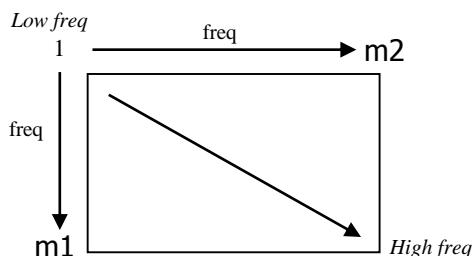
Setelah mempelajari bab ini, mahasiswa diharapkan dapat:

- ✓ Mengerti prinsip filtering dan konvolusi
- ✓ Melakukan low pass filter pada citra
- ✓ Memberikan noise pada citra
- ✓ Melakukan high pass filter pada citra

5.1. Prinsip-Prinsip Filtering

Filtering adalah suatu proses dimana diambil sebagian sinyal dari frekwensi tertentu, dan membuang sinyal pada frekwensi yang lain. Filtering pada citra juga menggunakan prinsip yang sama, yaitu mengambil fungsi citra pada frekwensi-frekwensi tertentu dan membuang fungsi citra pada frekwensi-frekwensi tertentu.

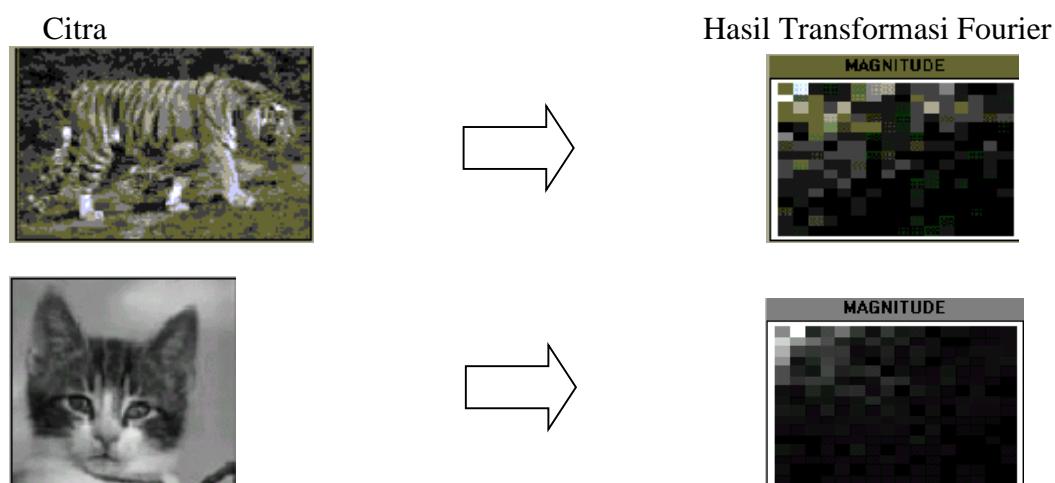
Berdasarkan sifat transformasi fourier dari suatu citra dan format koordinat frekwensi seperti Gambar 5.1. berikut ini:



Gambar 5.1. Format koordinat frekwensi pada citra

Berikutnya kita perhatikan bagaimana pengaruh frekwensi renda dan frekwensi tinggi pada citra dengan memanfaatkan hasil dari transformasi fourier. Dimana frekwensi pada citra dipengaruhi oleh gradiasi warna yang ada pada citra tersebut.

Perhatikan hasil transformasi fourier dari beberapa citra berikut ini:

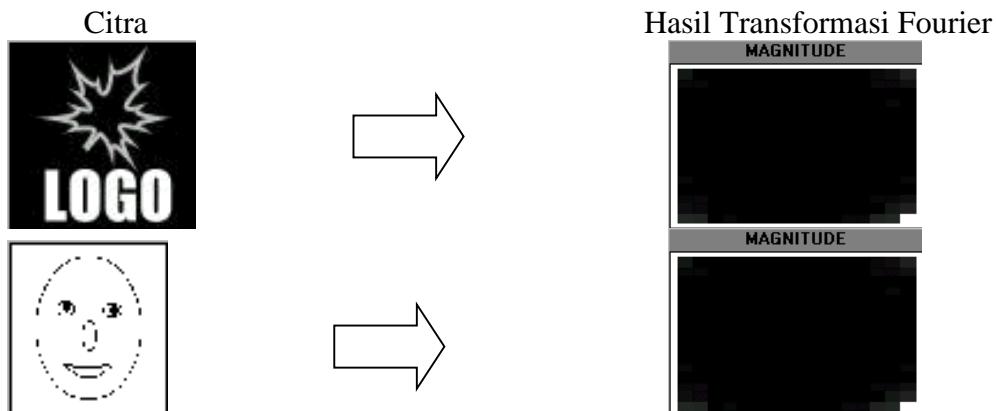


Gambar 5.2. Contoh transformasi fourier citra bergradiasi tinggi

Perhatikan bahwa warna putih (terang) pada gambar hasil transformasi fourier menunjukkan level atau nilai fungsi yang tinggi dan warna hitam (gelap) menunjukkan level atau nilai fungsi yang rendah. Berdasarkan hal ini dan format koordinat frekwensi (pada Gambar 5.1) terlihat bahwa pada Gambar 5.2 nilai-nilai yang tinggi berada pada frekwensi rendah, ini menunjukkan bahwa citra dengan gradiasi (level threshold) tinggi cenderung berada pada frekwensi rendah. Sehingga dapat disimpulkan bahwa citra dengan gradiasi tinggi berada pada frekwensi rendah.

Berikutnya dengan menggunakan citra-citra yang bergradiasi rendah seperti gambar logo data sketsa dimana nilai treshold yang digunakan merupakan nilai-nilai yang kecil dapat dilihat pada Gambar 5.3. berikut.

Pada Gambar 5.3. terlihat bahwa hasil transformasi fourier menunjukkan nilai fungsi hanya berada pada frekwensi tinggi. Dengan demikian dapat disimpulkan bahwa citra yang bergradiasi rendah berada pada frekwensi tinggi. Demikian pula citra biner, citra dengan threshold tertentu merupakan citra-citra yang bergradiasi rendah, dan citra-citra ini berada pada frekwensi tinggi.

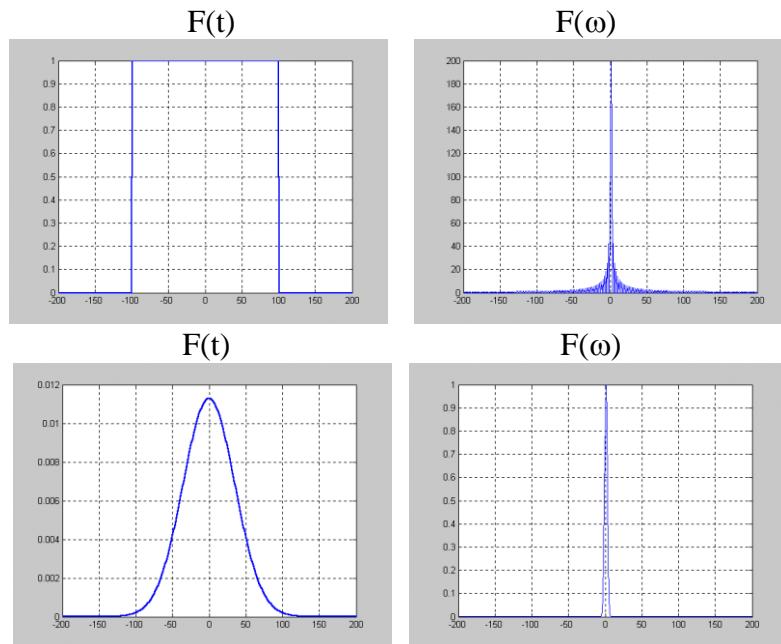


Gambar 5.3. Contoh transformasi fourier citra bergradiasi rendah

Dari sifat-sifat citra pada bidang frekwensi, maka prinsip-prinsip filtering dapat dikembangkan adalah sebagai berikut:

- Bila ingin mempertahankan gradiasi atau banyaknya level warna pada suatu citra, maka yang dipertahankan adalah frekwensi rendah dan frekwensi tinggi dapat dibuang atau dinamakan dengan Low Pass Filter. Hal ini banyak digunakan untuk reduksi noise dan proses blur.
- Bila ingin mendapatkan threshold atau citra biner yang menunjukkan bentuk suatu gambar maka frekwensi tinggi dipertahankan dan frekwensi rendah dibuang atau dinamakan dengan High Pass Filter. Hal ini banyak digunakan untuk menentukan garis tepi (edge) atau sketsa dari citra.
- Bila ingin mempertahankan gradiasi dan bentuk, dengan tetap mengurangi banyaknya bidang frekwensi (bandwidth) dan membuang sinyal yang tidak perlu maka frekwensi rendah dan frekwensi tinggi dipertahankan, sedangkan frekwensi tengahan dibuang atau dinamakan dengan Band Stop Filter. Teknik yang dikembangkan dengan menggunakan Wavelet Transform yang banyak digunakan untuk kompresi, restorasi dan denoising.

Teknik-teknik filtering pada citra dikembangkan menggunakan prinsip-prinsip di atas. Perhatikan beberapa transformasi fourier dari fungsi yang semuanya positif sebagai berikut:



Gambar 5.4. transformasi fourier untuk fungsi positif integer

Dari Gambar 5.4. di atas yang masing-masing menyatakan transformasi fourier dari fungsi rata-rata dan fungsi gaussian, dapat dilihat bahwa fungsi-fungsi ini menghasilkan transformasi fourier yang nilainya hanya berada pada frekwensi rendah. Hal ini dapat dinyatakan bahwa filter yang berjenis Low Pass Filter dapat diperoleh dengan suatu kernel filter H (koefisien filter dalam spasial atau koordinat x dan y) yang semua nilainya positif (positif absolut).

Contoh kernel dari Low Pass Filter yang terkenal adalah untuk ukuran $n \times n$ adalah:

(1) Filter rata-rata yang dirumuskan dengan:

$$H(i, j) = \frac{1}{n^2}, \forall i, j$$

Salah satu filter rata-rata yang banyak digunakan adalah filter rata-rata dengan ukuran 3×3 dan 5×5 sebagai berikut:

$$H = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \text{ dan}$$

$$H = \begin{bmatrix} 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \end{bmatrix}$$

(2) Filter Gaussian yang dirumuskan dengan:

$$H(x, y) = \frac{1}{s} e^{-\frac{x^2+y^2}{s^2}}$$

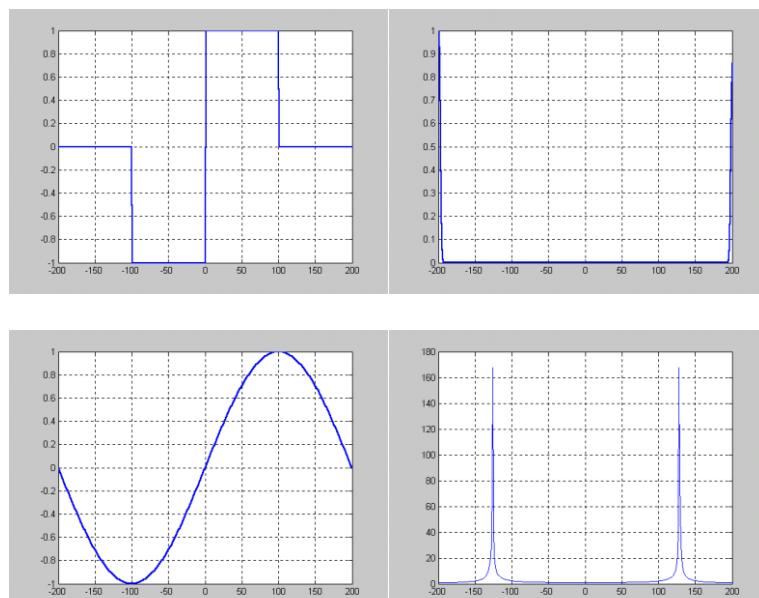
dimana s menyatakan lebar dari fungsi gaussian yang digunakan.

Salah satu filter gaussian yang banyak digunakan pada ukuran 3x3 dan 5x5 sebagai berikut:

$$H = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \text{ dan}$$

$$H = \begin{bmatrix} 1 & 3 & 4 & 3 & 1 \\ 3 & 7 & 9 & 7 & 3 \\ 4 & 9 & 12 & 9 & 4 \\ 3 & 7 & 9 & 7 & 3 \\ 1 & 3 & 4 & 3 & 1 \end{bmatrix}$$

Perhatikan gambar untuk transformasi fourier dari fungsi-fungsi yang bila dijumlahkan hasilnya nol sebagai berikut:



Gambar 5.6. Transformasi fourier untuk fungsi simetri nol

Dari Gambar 5.2. di atas yang masing-masing menyatakan transformasi fourier dari fungsi differensial dan fungsi laplacian, dapat dilihat bahwa fungsi-fungsi ini menghasilkan transfromasi fourier yang nilainya hanya berada pada frekwensi tinggi. Berdasarkan hal ini dapat dinyatakan bahwa filter yang berjenis High Pass Filter dapat diperoleh dengan suatu kernel H (koefisien filter dalam spasial atau koordinat x dan y) yang terdiri dari nilai yang simetri terhadap nol.

Contoh kernel dari Low Pass Filter yang terkenal adalah untuk ukuran nxn adalah:

- (1) Filter Differensial seperti pada filter Robert, dan filter Prewitt yang banyak digunakan untuk deteksi tepi.

$$\text{Robet} \rightarrow H = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \text{ atau } H = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$$

$$\text{Prewitt} \rightarrow H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \text{ atau } H = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

- (2) Filter Laplacian yang merupakan differensial dari fungsi gaussian dan dirumuskan dengan:

$$H = \frac{d^2}{dxdy} G(x, y)$$

dimana G(x,y) adalah fungsi Gaussian

5.2. Konvolusi

Konvolusi adalah perkalian total dari dua buah fungsi f dan h yang didefinisikan dengan:

$$f * h = \int_0^T f(t)h(T-t)dt$$

Untuk fungsi f dan h yang berdimensi 2, maka konvolusi dua dimensi didefinisikan dengan:

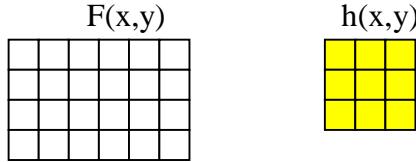
$$f * h = \iint_0^{T_x T_y} f(x, y)h(T_x - x, T_y - y)dxdy$$

Konvolusi 2D inilah yang banyak digunakan pengolahan citra digital, sayangnya rumus diatas sangat sulit diimplementasikan menggunakan komputer, karena pada dasarnya komputer hanya bisa melakukan perhitungan pada data yang diskrit sehingga tidak dapat digunakan untuk menghitung intregral di atas \ominus

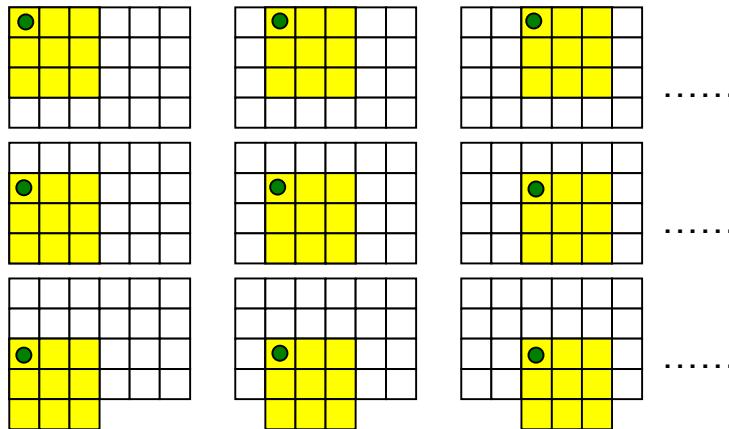
Konvolusi pada fungsi diskrit $f(n,m)$ dan $h(n,m)$ didefinisikan dengan:

$$y(k_1, k_2) = \sum_{n=1}^{T_n} \sum_{m=1}^{T_m} f(k_1 + n, k_2, m) h(n, m)$$

Perhitungan konvolusi semacam ini dapat digambarkan dengan:



Bila ingin dihitung $y = f * h$, maka proses perhitungannya dapat dilakukan dengan:



Gambar 5.6. Perhitungan konvolusi secara grafis

Contoh 5.1.

Diketahui :

$$f = \begin{bmatrix} 3 & 6 & 4 \\ 0 & 3 & 7 \end{bmatrix} \text{ dan } h = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Konvolusi antara f dan h adalah:

$$y = f * h$$

dapat dihitung dengan:

$$\begin{aligned} f(1,1) &= f(1,1).h(1,1) + f(1,2).h(1,2) + h(2,1).f(2,1) + f(2,2).h(2,2) \\ &= (3)(1) + (6)(1) + (0)(1) + (3)(1) \\ &= 3 + 6 + 0 + 3 = 12 \end{aligned}$$

$$\begin{aligned} f(1,2) &= f(1,2).h(1,1) + f(1,3).h(1,2) + h(2,2).f(2,1) + f(2,3).h(2,2) \\ &= (6)(1) + (4)(1) + (3)(1) + (7)(1) \\ &= 6 + 4 + 3 + 7 = 20 \end{aligned}$$

$$\begin{aligned} f(1,3) &= f(1,3).h(1,1) + f(1,4).h(1,2) + h(2,3).f(2,1) + f(2,4).h(2,2) \\ &\quad f(1,4) \text{ dan } f(2,4) \text{ dianggap nol karena tidak terdapat dalam } f \\ &= (4)(1) + (0)(1) + (7)(1) + (0)(1) \\ &= 4 + 0 + 7 + 0 = 11 \end{aligned}$$

$$\begin{aligned} f(2,1) &= f(2,1).h(1,1) + f(2,2).h(1,2) + h(3,1).f(2,1) + f(3,2).h(2,2) \\ &\quad f(3,1) \text{ dan } f(3,2) \text{ dianggap nol karena tidak terdapat dalam } f \end{aligned}$$

$$\begin{aligned}
&= (0)(1) + (3)(1) + (0)(1) + (0)(1) \\
&= 0 + 3 + 0 + 0 = 3 \\
f(2,2) &= f(2,2).h(1,1) + f(2,3).h(1,2) + h(3,2).f(2,1) + f(3,3).h(2,2) \\
&\quad f(3,2) \text{ dan } f(3,3) \text{ dianggap nol karena tidak terdapat dalam } f \\
&= (3)(1) + (7)(1) + (0)(1) + (0)(1) \\
&= 3 + 7 + 0 + 0 = 10 \\
f(2,3) &= f(2,3).h(1,1) + f(2,4).h(1,2) + h(3,3).f(2,1) + f(3,4).h(2,2) \\
&\quad f(2,4), f(3,3) \text{ dan } f(3,4) \text{ dianggap nol karena tidak terdapat dalam } f \\
&= (7)(1) + (0)(1) + (0)(1) + (0)(1) \\
&= 7 + 0 + 0 + 0 = 7
\end{aligned}$$

Hasil dari konvolusi:

$$f = \begin{bmatrix} 12 & 20 & 11 \\ 3 & 10 & 7 \end{bmatrix}$$

Contoh 5.2.

Diketahui :

$$f = \begin{bmatrix} 3 & 6 & 4 \\ 0 & 3 & 7 \end{bmatrix} \text{ dan } h = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Konvolusi antara f dan h adalah:

$$y = f * h$$

dapat dihitung dengan:

$$\begin{aligned}
f(1,1) &= f(1,1).h(1,1) + f(1,2).h(1,2) + h(2,1).f(2,1) + f(2,2).h(2,2) \\
&= (3)(-1) + (6)(0) + (0)(0) + (3)(1) \\
&= -3 + 0 + 0 + 3 = 0
\end{aligned}$$

$$\begin{aligned}
f(1,2) &= f(1,2).h(1,1) + f(1,3).h(1,2) + h(2,2).f(2,1) + f(2,3).h(2,2) \\
&= (6)(-1) + (4)(0) + (3)(0) + (7)(1) \\
&= -6 + 0 + 0 + 7 = 1
\end{aligned}$$

$$\begin{aligned}
f(1,3) &= f(1,3).h(1,1) + f(1,4).h(1,2) + h(2,3).f(2,1) + f(2,4).h(2,2) \\
&\quad f(1,4) \text{ dan } f(2,4) \text{ dianggap nol karena tidak terdapat dalam } f \\
&= (4)(-1) + (0)(0) + (7)(0) + (0)(1) \\
&= -4 + 0 + 0 + 0 = -4
\end{aligned}$$

$$\begin{aligned}
f(2,1) &= f(2,1).h(1,1) + f(2,2).h(1,2) + h(3,1).f(2,1) + f(3,2).h(2,2) \\
&\quad f(3,1) \text{ dan } f(3,2) \text{ dianggap nol karena tidak terdapat dalam } f \\
&= (0)(-1) + (3)(0) + (0)(0) + (0)(1) \\
&= 0 + 0 + 0 + 0 = 0
\end{aligned}$$

$$\begin{aligned}
f(2,2) &= f(2,2).h(1,1) + f(2,3).h(1,2) + h(3,2).f(2,1) + f(3,3).h(2,2) \\
&\quad f(3,2) \text{ dan } f(3,3) \text{ dianggap nol karena tidak terdapat dalam } f \\
&= (3)(-1) + (7)(0) + (0)(1) + (0)(1) \\
&= 3 + 0 + 0 + 0 = 3
\end{aligned}$$

$$\begin{aligned}
f(2,3) &= f(2,3).h(1,1) + f(2,4).h(1,2) + h(3,3).f(2,1) + f(3,4).h(2,2) \\
&\quad f(2,4), f(3,3) \text{ dan } f(3,4) \text{ dianggap nol karena tidak terdapat dalam } f \\
&= (7)(-1) + (0)(0) + (0)(0) + (0)(1) \\
&= -7 + 0 + 0 + 0 = -7
\end{aligned}$$

Hasil dari konvolusi:

$$f = \begin{bmatrix} 0 & 1 & -4 \\ 0 & 3 & -7 \end{bmatrix}$$

Contoh 5.3.

Diketahui :

$$f = \begin{bmatrix} 3 & 6 & 4 \\ 0 & 3 & 7 \end{bmatrix} \text{ dan } h = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix}$$

Konvolusi antara f dan h adalah:

$$y = f * h$$

dapat dihitung dengan:

$$\begin{aligned} f(1,1) &= f(1,1).h(1,1) + f(1,2).h(1,2) + h(2,1).f(2,1) + f(2,2).h(2,2) \\ &= (3)(0.25) + (6)(0.25) + (0)(0.25) + (3)(0.25) \\ &= 0.75 + 1.5 + 0 + 0.75 = 3 \end{aligned}$$

$$\begin{aligned} f(1,2) &= f(1,2).h(1,1) + f(1,3).h(1,2) + h(2,2).f(2,1) + f(2,3).h(2,2) \\ &= (6)(0.25) + (4)(0.25) + (3)(0.25) + (7)(0.25) \\ &= 1.5 + 1 + 0.75 + 1.75 = 5 \end{aligned}$$

$$\begin{aligned} f(1,3) &= f(1,3).h(1,1) + f(1,4).h(1,2) + h(2,3).f(2,1) + f(2,4).h(2,2) \\ &\quad f(1,4) \text{ dan } f(2,4) \text{ dianggap nol karena tidak terdapat dalam } f \\ &= (4)(0.25) + (0)(0.25) + (7)(0.25) + (0)(0.25) \\ &= 1 + 0 + 1.75 + 0 = 2.75 \end{aligned}$$

$$\begin{aligned} f(2,1) &= f(2,1).h(1,1) + f(2,2).h(1,2) + h(3,1).f(2,1) + f(3,2).h(2,2) \\ &\quad f(3,1) \text{ dan } f(3,2) \text{ dianggap nol karena tidak terdapat dalam } f \\ &= (0)(0.25) + (3)(0.25) + (0)(0.25) + (0)(0.25) \\ &= 0 + 0.75 + 0 + 0 = 0.75 \end{aligned}$$

$$\begin{aligned} f(2,2) &= f(2,2).h(1,1) + f(2,3).h(1,2) + h(3,2).f(2,1) + f(3,3).h(2,2) \\ &\quad f(3,2) \text{ dan } f(3,3) \text{ dianggap nol karena tidak terdapat dalam } f \\ &= (3)(0.25) + (7)(0.25) + (0)(0.25) + (0)(0.25) \\ &= 0.75 + 1.75 + 0 + 0 = 2.5 \end{aligned}$$

$$\begin{aligned} f(2,3) &= f(2,3).h(1,1) + f(2,4).h(1,2) + h(3,3).f(2,1) + f(3,4).h(2,2) \\ &\quad f(2,4), f(3,3) \text{ dan } f(3,4) \text{ dianggap nol karena tidak terdapat dalam } f \\ &= (7)(0.25) + (0)(0.25) + (0)(0.25) + (0)(0.25) \\ &= 1.75 + 0 + 0 + 0 = 1.75 \end{aligned}$$

Hasil dari konvolusi:

$$f = \begin{bmatrix} 3 & 5 & 2.75 \\ 0.75 & 2.5 & 1.75 \end{bmatrix}$$

Filter pada citra pada bidang spasial dapat dilakukan dengan menggunakan konvolusi dari citra (I) dan fungsi filternya (H), dan dituliskan dengan:

$$I' = H \otimes I$$

Dan dirumuskan dengan:

$$I'(x, y) = \sum_{i=-n}^n \sum_{j=-m}^m h(i, j)I(x+i, y+j)$$

dimana : m,n adalah ukuran dari fungsi filter dalam matrik

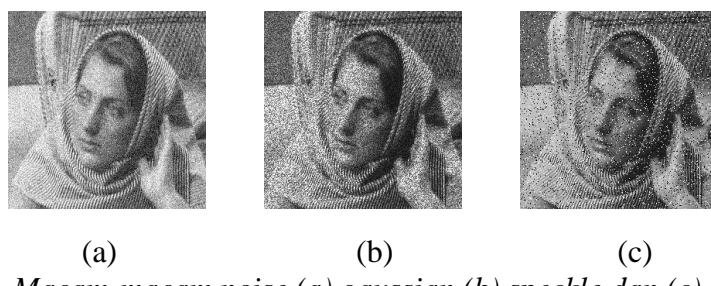
5.3. Low Pass Filter (Reduksi Noise)

Pada saat proses capture (pengambilan gambar), ada beberapa gangguan yang mungkin terjadi, seperti kamera tidak fokus atau munculnya bintik-bintik yang bisa jadi disebabkan oleh proses capture yang tidak sempurna. Setiap gangguan pada citra dinamakan dengan noise. Noise pada citra tidak hanya terjadi karena ketidak-sempurnaan dalam proses capture, tetapi bisa juga disebabkan oleh kotoran-kotoran yang terjadi pada citra.

Berdasarkan bentuk dan karakteristiknya, noise pada citra dibedakan menjadi beberapa macam yaitu:

- 1) Gaussian
- 2) Speckle
- 3) Salt & Pepper

Macam-macam noise ini dapat dilihat pada Gambar 5.7 berikut ini:



Gambar 5.7. Macam-macam noise (a) gaussian (b) speckle dan (c) salt & pepper

Noise gaussian merupakan model noise yang mengikuti distribusi normal standard dengan rata-rata nol dan standard deviasi 1. Efek dari gaussian noise ini, pada gambar muncul titik-titik berwarna yang jumlahnya sama dengan prosentase noise. Noise speckle merupakan model noise yang memberikan warna hitam pada titik yang terkena noise. Sedangkan noise salt & pepper seperti halnya taburan garam, akan memberikan warna putih pada titik yang terkena noise.

Pada beberapa pengolahan citra, terkadang untuk menguji suatu algoritma untuk dapat mereduksi noise, maka noise dihasilkan dari proses pembangkitan noise. Untuk membangkitkan noise digunakan suatu bilangan acak sebagai pengganti noise yang dihasilkan.

5.3.1. Membangkitkan Noise Uniform

Noise Uniform seperti halnya noise gaussian dapat dibangkitkan dengan cara membangkitkan bilangan acak [0,1] dengan distribusi uniform. Kemudian untuk titik-titik yang terkena noise, nilai fungsi citra ditambahkan dengan nilai noise yang ada, atau dirumuskan dengan:

$$y(i, j) = x(i, j) + p.a$$

dimana:

a = nilai bilangan acak berdistribusi uniform dari noise

p = prosentase noise

y(i,j) = nilai citra terkena noise.

x(i,j) = nilai citra sebelum terkena noise.

Catatan:

Fungsi rnd dalam visual basic merupakan fungsi pembangkitan bilangan acak berdistribusi uniform, sehingga noise dapat dibangkitkan secara langsung dengan mengalikannya dengan besaran noise yang dibutuhkan.

Noise uniform ini merupakan noise sintesis yang sebenarnya dalam penerapannya jarang digunakan, tetapi secara pemrograman pembangkitan noise uniform ini merupakan jenis pembangkitan noise yang paling mudah.

5.3.2. Membangkitkan Noise Gaussian

Noise gaussian dapat dibangkitkan dengan cara membangkitkan bilangan acak [0,1] dengan distribusi gaussian. Kemudian untuk titik-titik yang terkena noise, nilai fungsi citra ditambahkan dengan nilai noise yang ada, atau dirumuskan dengan:

$$y(i, j) = x(i, j) + p.a$$

dimana:

a = nilai bilangan acak berdistribusi gaussian

p = prosentase noise

y(i,j) = nilai citra terkena noise.

x(i,j) = nilai citra sebelum terkena noise.

Untuk membangkitkan bilangan acak berdistribusi gaussian, tidak dapat langsung menggunakan fungsi rnd, tetapi diperlukan suatu metode yang digunakan untuk mengubah distribusi bilangan acak ke dalam fungsi f tertentu. Dalam buku ini digunakan

metode rejection untuk memudahkan dalam alur pembuatan programnya. Metode rejection dikembangkan dengan cara membangkitkan dua bilangan acak (x,y) dan ditolak bila $y > f(x)$.

5.3.3. Membangkitkan Noise Salt & Pepper

Noise *salt & pepper* dapat dibangkitkan dengan cara membangkitkan bilangan 255 (warna putih) pada titik-titik yang secara probabilitas lebih kecil dari nilai probabilitas noise, dan dirumuskan dengan:

$$F(x,y)=255 \text{ jika } p(x,y) < ProbNoise$$

Dimana: $F(x,y)$ adalah nilai gray-scale pada titik (x,y)
 $p(x,y)$ adalah probabilitas acak

5.3.4. Membangkitkan Noise Speckle

Noise *speckle* dapat dibangkitkan dengan cara membangkitkan bilangan 0 (warna hitam) pada titik-titik yang secara probabilitas lebih kecil dari nilai probabilitas noise, dan dirumuskan dengan:

$$F(x,y)=0 \text{ jika } p(x,y) < ProbNoise$$

Dimana: $F(x,y)$ adalah nilai gray-scale pada titik (x,y)
 $p(x,y)$ adalah probabilitas acak

5.3.5. Reduksi Noise Menggunakan Filter Rata-Rata

Ada berbagai macam teknik untuk mengurangi (reduksi) noise, salah satunya menggunakan filter rata-rata. Dalam pengertian noise sebagai suatu nilai yang berbeda dengan semua tetangganya maka dapat dikatakan noise merupakan nilai-nilai yang berada pada frekwensi tinggi, untuk mengurangi noise digunakan Low Pass Filter (LPF). Salah satu dari bentuk LPF adalah filter rata-rata.

Filter rata-rata merupakan filter H dalam bentuk matrik yang berukuran mxn , dan nilainya adalah sama untuk setiap eleme, dan karena bersifat LPF maka jumlah seluruh elemen adalah satu, dan dituliskan dengan:

$$H(i, j) = \frac{1}{m \cdot n}, 1 \leq i \leq m, 1 \leq j \leq n$$

Filter rata-rata berukuran 3x3 adalah:

$$H = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \text{ atau dituliskan } H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} / 9$$

Filter rata-rata berukuran 5x5 adalah:

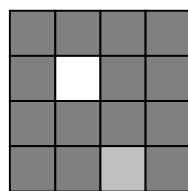
$$H = \begin{bmatrix} \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \\ \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} & \frac{1}{25} \end{bmatrix}$$

Contoh 6.1.

Perhatikan potongan gambar dalam bentuk matrik berikut ini:

$$F = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 3 & 1 \end{bmatrix}$$

Bila digambarkan adalah sebagai berikut:

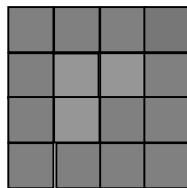


Dengan menggunakan filter rata-rata 3x3, maka proses filter dilakukan dengan cara menghitung konvolusi dari kernel filter rata-rata 3x3 dan H, diperoleh:

$$Y = H \otimes F$$

$$\begin{aligned}
Y &= \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 3 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{9} & 1 & 1 & \frac{1}{9} \\ 1 & \frac{12}{9} & \frac{12}{9} & \frac{1}{9} \\ 1 & \frac{14}{9} & \frac{12}{9} & \frac{1}{9} \\ \frac{4}{9} & \frac{8}{9} & \frac{8}{9} & \frac{1}{9} \end{bmatrix}
\end{aligned}$$

Bila digambarkan hasil filter rata-rata adalah sebagai berikut:



Hasilnya menunjukkan titik-titik yang berbeda (dianggap sebagai noise) akan dikurangi hingga mendekati nilai-nilai yang dominan. Salah satu efek dari hasil filter ini adalah terjadinya efek blur.

Untuk mengukur kinerja dari filter ini digunakan nilai pengaruh noise terhadap suatu sinyal (dalam hal ini citra adalah sinyal 2 dimensi) yang dinyatakan sebagai nilai SNR (Signal to Noise Ratio), dan dirumuskan dengan:

$$SNR = 10^{10} \log \left(\frac{I}{N} \right)$$

dimana: I adalah jumlah nilai citra

N adalah jumlah nilai noise (citra terkena noise In dikurangi dengan citra I)

Untuk menghitung SNR pada contoh di atas adalah sebagai berikut:

$$I = \sum_x \sum_y F'(x, y) = 16$$

dengan dianggap bahwa semua nilai F adalah 1 (seragam), sedangkan nilai yang lain dianggap noise

$$\begin{aligned}
N &= \sum_x \sum_y |Y(x, y) - F(x, y)| \\
&= \sum_x \sum_y \left[\begin{matrix} \frac{1}{9} & 1 & 1 & \frac{1}{9} \\ 1 & \frac{12}{9} & \frac{12}{9} & \frac{1}{9} \\ 1 & \frac{14}{9} & \frac{12}{9} & \frac{8}{9} \\ \frac{1}{9} & \frac{8}{9} & \frac{8}{9} & \frac{1}{9} \end{matrix} \right] - \left[\begin{matrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{matrix} \right] \\
&= \sum_c \sum_y \left[\begin{matrix} \frac{1}{9} & 0 & 0 & \frac{5}{9} \\ 0 & \frac{3}{9} & \frac{3}{9} & \frac{3}{9} \\ 0 & \frac{5}{9} & \frac{3}{9} & \frac{1}{9} \\ \frac{5}{9} & \frac{1}{9} & \frac{1}{9} & \frac{3}{9} \end{matrix} \right] = \frac{35}{9} = 3.89
\end{aligned}$$

$$SNR = 10^{10} \log \left(\frac{16}{3.89} \right) = 6.14$$

Dibandingkan dengan gambar sebelum filter, SNR sebelum filter dapat dihitung dengan:

N=5

$$SNR = 10^{10} \log \left(\frac{16}{5} \right) = 5.05$$

Ada peningkatan SNR setelah dilakukan filter rata-rata, hal ini berarti jumlah noise berkurang.

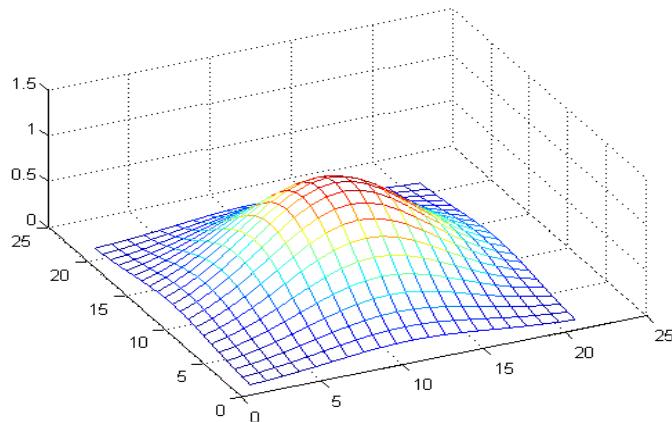
5.3.6. Reduksi Noise Menggunakan Filter Gaussian

Filter lain yang banyak digunakan dalam mereduksi noise selain filter rata-rata adalah filter gaussian. Filter gaussian ini sebenarnya hampir sama dengan filter rata-rata hanya ada nilai bobot yang tidak rata seperti pada filter rata-rata, tetapi mengikuti fungsi gaussian sebagai berikut:

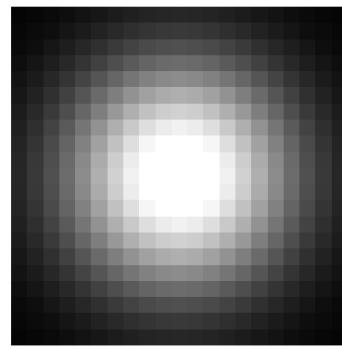
$$G(x, y) = \frac{1}{s\sqrt{\pi}} e^{-((x-m_x)^2 + (y-m_y)^2)/2s}$$

dimana: s adalah sebaran dari fungsi gaussian

(mx,my) adalah titik tengah dari fungsi gaussian



Gambar 5.8 Model Fungsi Gaussian dalam ruang.



Gambar 5.9. Model contour dari filter gaussian.

Berdasarkan rumus dari fungsi gaussian di atas untuk ukuran 3x3 akan diperoleh matrik kernel filter gaussian:

$$H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 1 \end{bmatrix} / 13 \quad \text{atau} \quad H = \begin{bmatrix} 0.077 & 0.077 & 0.077 \\ 0.077 & 0.308 & 0.077 \\ 0.077 & 0.077 & 0.077 \end{bmatrix}$$

Kernel filter gaussian untuk ukuran 5x5 adalah:

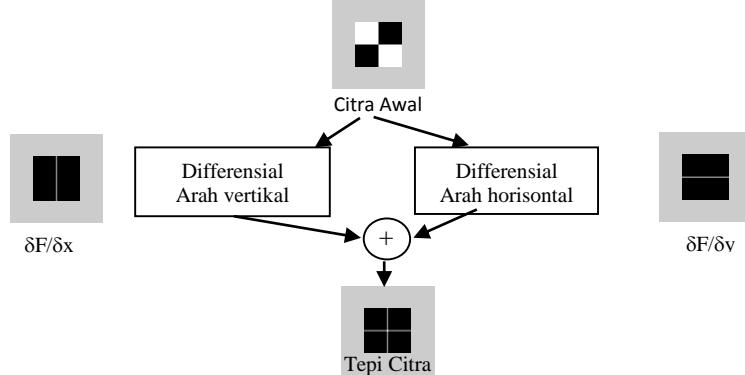
H =	0.0030	0.0133	0.0219	0.0133	0.0030
	0.0133	0.0596	0.0983	0.0596	0.0133
	0.0219	0.0983	0.1621	0.0983	0.0219
	0.0133	0.0596	0.0983	0.0596	0.0133
	0.0030	0.0133	0.0219	0.0133	0.0030

Kernel filter gaussian untuk ukuran 7x7 adalah:

```
0.0013 0.0041 0.0079 0.0099 0.0079 0.0041 0.0013  
0.0041 0.0124 0.0241 0.0301 0.0241 0.0124 0.0041  
0.0079 0.0241 0.0470 0.0587 0.0470 0.0241 0.0079  
0.0099 0.0301 0.0587 0.0733 0.0587 0.0301 0.0099  
0.0079 0.0241 0.0470 0.0587 0.0470 0.0241 0.0079  
0.0041 0.0124 0.0241 0.0301 0.0241 0.0124 0.0041  
0.0013 0.0041 0.0079 0.0099 0.0079 0.0041 0.0013
```

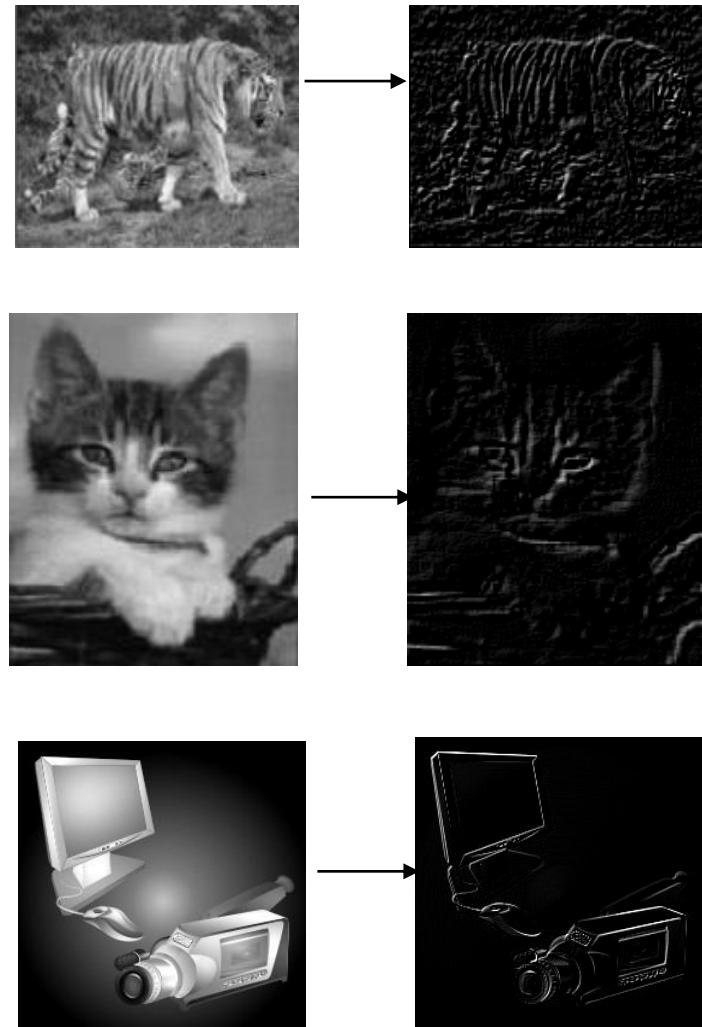
5.4. High Pass Filter (Deteksi Tepi)

Deteksi tepi (Edge Detection) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek gambar. Suatu titik (x,y) dikatakan sebagai tepi (edge) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangga. Gambar 8.1 berikut ini meng-gambarkan bagaimana tepi suatu gambar diperoleh.



Gambar 5.10. Proses Deteksi Tepi

Perhatikan hasil deteksi dari beberapa citra menggunakan model differensial di atas:



Gambar 5.11. Hasil beberapa deteksi tepi

Pada gambar 5.11. terlihat bahwa hasil deteksi tepi berupa tepi-tepi dari suatu gambar. Bila diperhatikan bahwa tepi suatu gambar terletak pada titik-titik yang memiliki perbedaan tinggi. Berdasarkan prinsip-prinsip filter pada citra maka tepi suatu gambar dapat diperoleh menggunakan High Pass Filter (HPF), yang mempunyai karakteristik:

$$\sum_{y} \sum_{x} H(x, y) = 0$$

Contoh:

Diketahui fungsi citra $f(x,y)$ sebagai berikut:

1	1	1	1	1
1	1	1	1	0
1	1	1	0	0
1	1	0	0	0
1	0	0	0	0

Dengan menggunakan filter : $H(x, y) = \begin{bmatrix} -1 & 1 \end{bmatrix}$

Maka Hasil filter adalah :

0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
1	0	0	0	0

Bila digambarkan maka proses filter di atas mempunyai masukan dan keluaran sebagai berikut



Catatan:

Untuk mencoba perhitungan di atas dapat dilakukan dengan cara manual menggunakan perhitungan konvolusi yang telah dibahas pada bab 5, atau dengan memanfaatkan program konvolusi.

Macam-macam metode untuk proses deteksi tepi ini, antara lain:

1. Metode Robert
2. Metode Prewitt
3. Metode Sobel

Metode yang banyak digunakan untuk proses deteksi tepi adalah metode Robert, Prewitt dan Sobel, Gonzalez[1].

5.4.1. Metode Robert

Metode Robert adalah nama lain dari teknik differensial yang dikembangkan di atas, yaitu differensial pada arah horisontal dan differensial pada arah vertikal, dengan ditambahkan proses konversi biner setelah dilakukan differensial. Teknik konversi biner yang disarankan adalah konversi biner dengan meratakan distribusi warna hitam dan putih [5], seperti telah dibahas pada bab 3. Metode Robert ini juga disamakan dengan teknik DPCM (Differential Pulse Code Modulation)

Kernel filter yang digunakan dalam metode Robert ini adalah:

$$H = \begin{bmatrix} -1 & 1 \end{bmatrix} \text{ dan } H = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

5.4.2. Metode Prewitt

Metode Prewitt merupakan pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF.

Kernel filter yang digunakan dalam metode Robert ini adalah:

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad H = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

5.4.3. Metode Sobel

Metode Sobel merupakan pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi.

Kernel filter yang digunakan dalam metode Robert ini adalah:

$$H = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{dan} \quad H = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

5.5. Ringkasan

1. Jenis-jenis filter :
 - a. Smoothing filters:
 - Lowpass filter (linear filter, mengambil nilai rata-rata)
 - Median filter (non-linear filter, mengambil median dari setiap jendela ketetanggan)
 - b. Sharpening filters:
 - Highpass filter
 - Roberts, Prewitt, Sobel (edge detection)
2. Window operator 3x3 untuk filtering citra di domain spatial :

Table 5.2 Major Spatial Filters

Filters	3×3 Window Operator	Effects
Mean	$A = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ or $\frac{1}{5} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	smoothing without noises but with loss of sharpness
Sobel	$A_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$, $A_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$ $ A_x + A_y $ or $\sqrt{A_x^2 + A_y^2}$	gradient and linear features are enhanced
Laplacian	$A = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ or $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	differential and linear features are enhanced
Median	Replaced by the median value of 3×3 window	smoothed image with a little loss of sharpness
Highpass	$A = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$ or $\frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	edge enhancement
Sharpening	$A = \frac{1}{9} \begin{bmatrix} 1 & -8 & 1 \\ -8 & 37 & -8 \\ 1 & -8 & 1 \end{bmatrix}$	sharpened with clearness

5.6. Latihan

1. Apa pengaruh frekwensi terhadap suatu gambar ?
2. Gambar-gambar bagaimana yang mempunyai frekwensi rendah ?
3. Gambar-gambar bagaimana yang mempunyai frekwensi tinggi ?

4. Jelaskan prinsip-prinsip Low Pass Filter pada citra, dan apa akibatnya terhadap citra?
5. Jelaskan prinsip-prinsip High Pass Filter pada citra, dan apa akibatnya terhadap citra?
6. Jelaskan rumus konvolusi dan proses perhitungannya.
7. Hitunglah konvolusi dari H dan F berikut ini:

$$(a) H = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$(b) H = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

(c) Apa perbedaan hasil dari (a) dan (b)

8. Apa ciri-ciri kernel filter pada LPF? Berikan contohnya.
9. Apa ciri-ciri kernel filter pada HPF? Berikan contohnya.
10. Dengan menggunakan program HPF di atas, ubahlah kernel filternya menjadi:

$$H = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Bagaimana hasilnya dibandingkan dengan sebelumnya.

11. Apa pengertian noise pada citra? Jelaskan dengan contoh.
12. Sebutkan 3 macam noise yang dibangkitkan untuk pengolahan citra, dan apa ciri dari masing-masing noise tersebut.
13. Bagaimana cara membangkitkan noise gaussian?
14. Dengan menggunakan program pembangkitan noise gaussian di atas, hitung SNR untuk prosentase noise 10%, 20%, 30% sampai dengan 90%, dan gambarkan hubungan prosentase noise dengan nilai SNR.
15. Bagaimana cara membangkitkan noise salt & pepper?
16. Dengan menggunakan program pembangkitan noise salt & pepper di atas, hitung SNR untuk prosentase noise 10%, 20%, 30% sampai dengan 90%, dan gambarkan hubungan prosentase noise dengan nilai SNR.
17. Bagaimana cara membangkitkan noise speckle?

18. Dengan menggunakan program pembangkitan noise speckle di atas, hitung SNR untuk prosentase noise 10%, 20%, 30% sampai dengan 90%, dan gambarkan hubungan prosentase noise dengan nilai SNR.
19. Mengapa untuk mereduksi noise digunakan low pass filter (LPF)?
20. Jelaskan ciri-ciri dari filter rata-rata untuk mereduksi noise?
21. Dengan program reduksi noise gaussian menggunakan filter rata-rata 3x3, hitunglah perbaikan SNR bila noise dibangkitkan dengan prosentase noise 5%, 10%, 15%, 20%, sampai dengan 50%.
22. Jelaskan ciri-ciri dari filter gaussian untuk mereduksi noise?
23. Dengan program reduksi noise gaussian menggunakan filter gaussian 3x3, hitunglah perbaikan SNR bila noise dibangkitkan dengan prosentase noise 5%, 10%, 15%, 20%, sampai dengan 50%.
24. Bandingkan hasil pada soal no 11 dan 13 untuk gambar yang sama. Jelaskan apa perbedaan kedua macam filter untuk mereduksi noise.
25. Apa pengertian deteksi tepi pada citra? Jelaskan dengan contoh.
26. Jelaskan fungsi differensial pada proses deteksi tepi.
27. Dengan menggunakan program deteksi tepi dengan metode Robert di atas, lakukan deteksi tepi untuk gambar-gambar pemandangan, wajah dan sidik jari. Perhatikan bagaimana hasilnya, apa yang dapat saudara simpulkan?
28. Dengan menggunakan program deteksi tepi dengan metode Prewitt di atas, lakukan deteksi tepi untuk gambar-gambar pemandangan, wajah dan sidik jari. Perhatikan bagaimana hasilnya, apa yang dapat saudara simpulkan?
29. Dengan menggunakan program deteksi tepi dengan metode Sobel di atas, lakukan deteksi tepi untuk gambar-gambar pemandangan, wajah dan sidik jari. Perhatikan bagaimana hasilnya, apa yang dapat saudara simpulkan?
30. Jelaskan perbedaan dari hasil masing-masing metode di atas.
31. Dengan menggunakan program deteksi tepi sobel, bagaimana jika filternya diganti dengan filter Laplacian yang mempunyai kernel

$$H = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

BAB 6

CITRA BERWARNA

POKOK BAHASAN:

- ✓ Konsep Warna
- ✓ Ruang Warna
- ✓ Color Gamut
- ✓ Konversi Warna

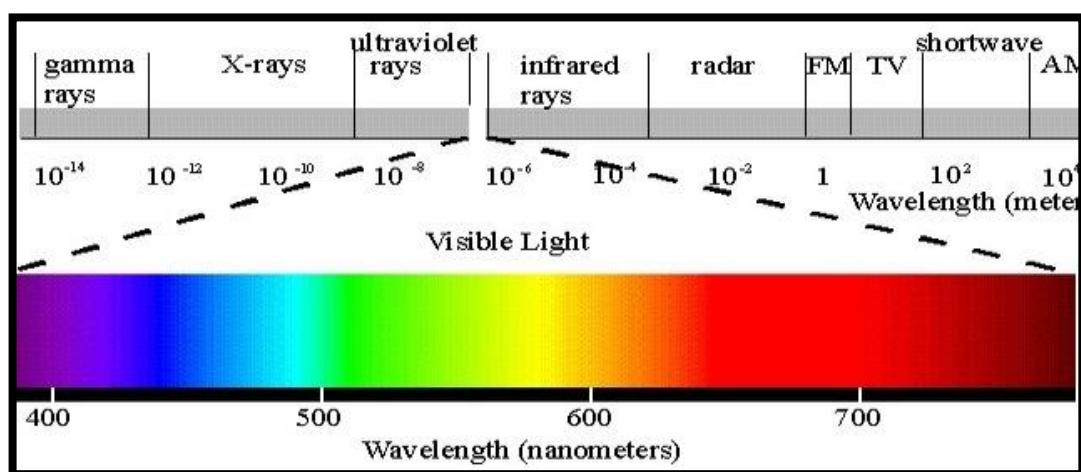
TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat:

- ✓ Mengerti konsep warna pada citra
- ✓ Bekerja dengan bermacam-macam format warna yang ada

6.1. Konsep Warna

Secara teknik, warna adalah spektrum tertentu yang terdapat di dalam suatu cahaya sempurna (berwarna putih). Identitas suatu warna ditentukan panjang gelombang cahaya tersebut. Panjang gelombang warna yang masih bisa ditangkap mata manusia berkisar antara 380-780 nanometer. Sebagai contoh warna biru memiliki panjang gelombang 460 nanometer.



Gambar 6.1 Spektrum Gelombang Elektromagnetik

Gambar 6. 1 adalah spektrum gelombang elektromagnetik. Spektrum gelombang warna berada di antara infra-merah dan ultra-violet. Infra-merah memiliki panjang gelombang $> 760\text{nm}$ dan ultra-violet $<400\text{nm}$. Jadi range panjang gelombang untuk warna yang dapat kita lihat adalah $400\text{nm} >$ warna $<760\text{nm}$. Warna merah mempunyai range yang sangat lebar dan panjang gelombang yang paling tinggi (frekwensi paling rendah).

Cahaya yang kita lihat adalah bukan cahaya dengan satu panjang gelombang melainkan kumpulan panjang gelombang tertentu. Warna terbentuk dari kumpulan gelombang dengan panjang gelombang yang berbeda-beda. Ini bisa diartikan bahwa warna yang kita lihat adalah kombinasi dari beberapa elemen dasar warna. Cara penyajian campuran elemen dasar untuk menghasilkan warna dinamakan dengan **Color Space** atau Ruang Warna.

Sebuah warna bisa dihasilkan dengan kombinasi elemen dasar warna. Kombinasi elemen dasar warna bukan hitungan matematis, tetapi lebih pada pengalaman manusia dalam mengenali warna. Kadang-kadang untuk bisa menyajikan secara matematis, harus menggunakan lebih dari 3 komponen warna.

Setiap warna sekurang-kurangnya mempunyai 3 elemen dasar warna. RGB mempunyai komponen Red, Green dan Blue, yang sering disebut sebagai additive color. CMY(K) mempunyai komponen Cyan, Magenta dan Yellow, yang sering disebut dengan subtractive color.

6.2. Ruang Warna

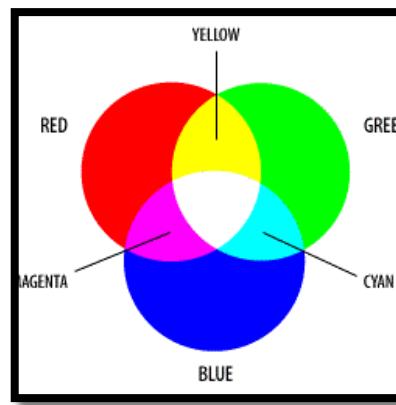
Ruang warna adalah cara penyajian campuran elemen dasar untuk menghasilkan warna. Ada beberapa Color space yang dikenal dan banyak digunakan. Berikut adalah macam-macam dari Color space:

- RGB
- CMY(K)
- HSV
- CIE XYZ
- Lab
- Yuv
- YCrCb

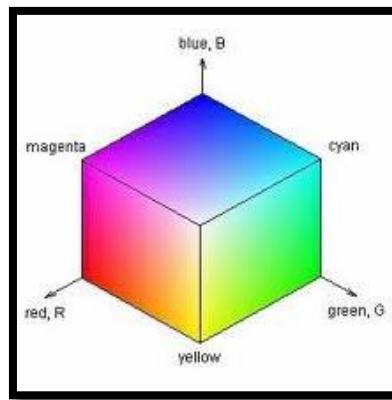
6.2.1. RGB

Seperti terlihat pada Gambar 6.2 RGB menggunakan elemen dasar Red, Green, Blue untuk menghasilkan warna-warna. RGB disebut juga additive color. RGB digunakan

untuk menghasilkan warna pada CRT, monitor dan display. Untuk menghasilkan sebuah warna kombinasi antara r, g, dan b dijelaskan dengan RGB Cube seperti terlihat pada Gambar 6.3. RGB CUBE banyak digunakan untuk menghasilkan fitur warna pada citra untuk aplikasi pencarian citra berdasarkan warna. Dalam RGB Cube masih ada warna-warna yang sulit untuk dinyatakan seperti warna “merah darah”.



Gambar 6.2 Elemen Warna Dasar RGB



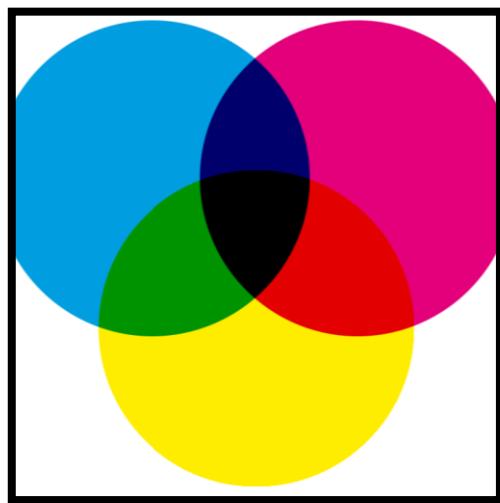
Gambar 6.3 RGB Cube

6.2.2. CMY(K)

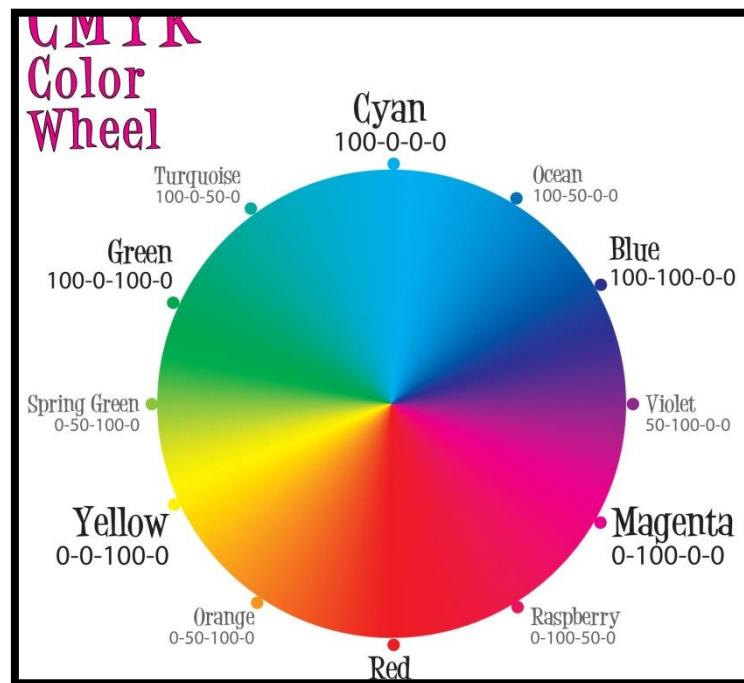
Seperti terlihat pada Gambar 6.4 CMY(K) menggunakan elemen dasar Cyan, Magenta dan Yellow. Untuk lebih lengkapnya ditambahkan elemen K (Chroma). CMY(K) adalah kombinasi warna yang digunakan dalam pencetakan (*printing*). Dikenal sebagai subtractive color.

Gambar 6.5 adalah CMY(K) Color Wheel yang digunakan untuk menjelaskan kombinasi warna pada CMY(K). Kombinasi setiap elemen warna pada CMY(K)

menggunakan skala persentase. Pada color space CMY(K) ini warna biru dan merah mempunyai area yang lebih luas.



Gambar 6.4 Elemen Dasar CMY(K)



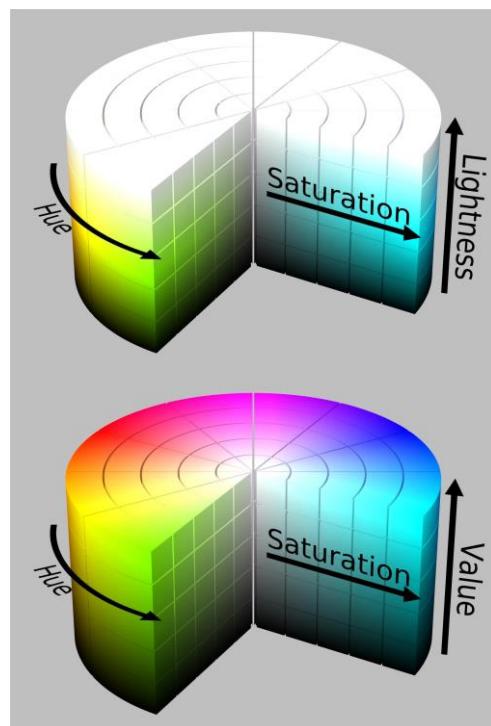
Gambar 6.5 CMY(K) Color Wheel

6.2.3. HSV

HSV mempunyai elemen dasar **Hue**, **Saturation** dan **Value**. Hue menyatakan keluarga warna (dalam satuan derajat). Saturation menyatakan sensasi/intensitas warna. Value menyatakan derajat keabuan atau terang/gelap gambar. HSV dikembangkan menggunakan sistem koordinat Polar Seperti Terlihat pada Gambar 6.6. HSV banyak digunakan untuk fitur warna pada gambar.

Berikut adalah arang nilai untuk H, S, dan V:

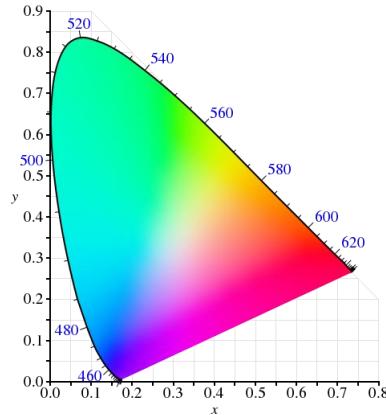
- Nilai H → 0 s/d 360
- Nilai S → 0 s/d 1
- Nilai V → 0 s/d 255



Gambar 6.6 Koordinat Polar HSV

6.2.4. CIE XYZ

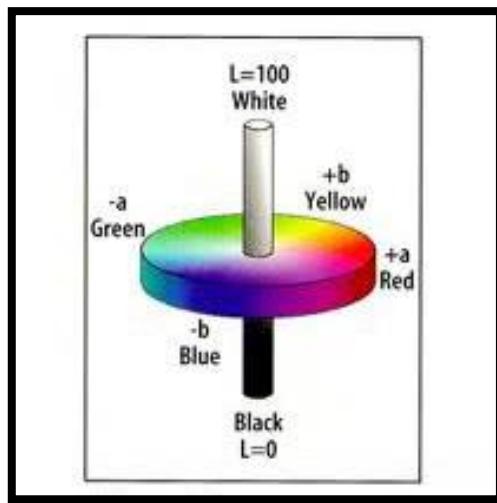
CIE atau CIE 1931 adalah Color space pertama yang didefinisikan secara matematis. CIE diusulkan oleh International Commission on Illumination (Comission Internationale de l'Eclairage). CIE merupakan standar yang disusun berdasarkan persepsi manusia dan baik untuk percobaan perbandingan warna.



Gambar 6.7 CIE Coor Space Chromaticity Diagram

6.2.4. Lab - Photoshop

Lab atau CIE Lab merupakan color space yang dikenalkan oleh Photoshop. Merupakan perbaikan dari model CIE. Lab mempunyai elemen L (Luminance), a (range dari hijau ke merah) dan b (range dari biru ke kuning). Gambar 6.8 menjelaskan kombinasi warna pada Lab.



Gambar 6.8 Lab - Photoshop

6.2.5. YUV dan YCrCb

Yuv pada awalnya digunakan untuk video analog PAL, namun sekarang dipakai juga untuk standard CCIR 601 pada video digital.

Y (luminance) didapat dari nilai RGB :

$$Y = 0.299R + 0.587G + 0.114B$$

Chrominance adalah perbedaan warna dan putih. Ini dapat dinyatakan dalam nilai U dan V.

$$U = B - Y; V = R - Y$$

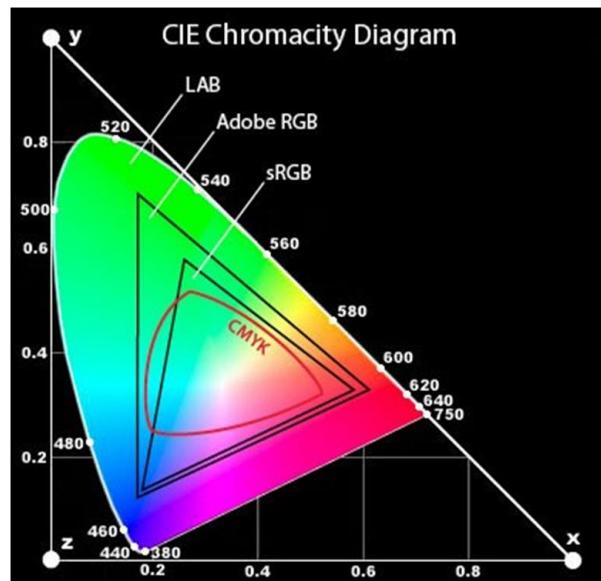
Sedangkan YCrCb adalah versi skala dari YUV dan digunakan dalam JPEG dan MPEG dimana semua komponen bernilai positif.

$$Cb = (B - Y) / 1.772 + 0.5$$

$$Cr = (R - Y) / 1.402 + 0.5$$

6.3. Color Gamut

Ada beberapa warna yang ada pada RGB namun tidak ada pada CMYK, demikian juga sebaliknya. Untuk melihat interseksi dan warna-warna spesifik dalam ruang warna diperlukan Color Gamut. Color Gamut adalah sebuah range yang menyatakan batasan-batasan warna yang bisa dihasilkan oleh setiap color space Seperti terlihat pada Gambar 6.9. dalam gambar tersebut ada Colo space Adobe RGB. Adobe RGB merupakan color space baru yang dikembangkan dari RGB dengan lebih banyak warna yang mampu disajikan.



Gambar 6.9 Color Gamut

6.4. Konversi Warna

Kita dapat melakukan konversi dari RGB ke CMYK, HSV ataupun YCrCb. Ataupun sebaliknya dari CMYK, HSV, YCrCb menjadi RGB. Adapun rumusan yang digunakan untuk konversi adalah sebagai berikut:

1. RGB – CMYK

Untuk mengubah RGB menjadi CMYK terlebih dahulu kita harus mendapatkan nilai R' G' dan B' kemudian menggunakan nilai tersebut untuk mendapatkan nilai K. Terakhir kita mendapatkan nilai C, M dan Y dari nilai K tersebut.

- $R' = R/255$
- $G' = G/255$
- $B' = B/255$
- $K = 1 - \max(R', G', B')$
- $C = (1-R'-K)/(1-K)$
- $M = (1-G'-K)/(1-K)$
- $Y = (1-B'-K)/(1-K)$

Dimana:

- R adalah nilai Red (R)
- G adalah Green (G)
- B adalah Blue (B)
- K adalah Chroma
- C adalah Cyan
- M adalah Magenta
- Y adalah Yellow

2. RGB – HSV

Untuk mendapatkan nilai HSV terlebih dahulu kita harus mendapatkan Nilai R' G' dan B'. Kemudian mencari nilai Cmax Cmin dan D. Berikut adalah rumusan untuk konversi nilai RGB menjadi HSV

- $R' = R/255$
- $G' = G/255$
- $B' = B/255$
- $Cmax = \max(R', G', B')$
- $Cmin = \min(R', G', B')$
- $D = Cmax - Cmin$

$$H = \begin{cases} 60 \cdot \left(\frac{G' - B'}{D} \bmod 6 \right) \\ 60 \cdot \left(\frac{B' - R'}{D} + 2 \right) \\ 60 \cdot \left(\frac{R' - G'}{D} + 4 \right) \end{cases}$$

$$S = \begin{cases} 0, \text{ jika } D = 0 \\ \frac{D}{Cmax}, \text{ jika } D \neq 0 \end{cases}$$

$$V = Cmax$$

3. RGB – Yuv

Berikut adalah rumusan untuk konversi RGB menjadi Yuv:

- $Y = 0.299R + 0.587G + 0.114B$
- $U = B - Y;$
- $V = R - Y$

4. RGB - YCrCb

Berikut adalah rumus untuk konversi RGB menjadi YCrCb:

- $Y = 0.299R + 0.587G + 0.114B$
- $Cb = (B - Y) / 1.772 + 0.5$
- $Cr = (R - Y) / 1.402 + 0.5$

6.5. Ringkasan

1. Warna adalah spektrum tertentu yang terdapat di dalam suatu cahaya sempurna (berwarna putih).
2. Identitas suatu warna ditentukan panjang gelombang cahaya tersebut. Panjang gelombang warna yang masih bisa ditangkap mata manusia berkisar antara 380-780 nanometer.
3. Color space atau ruang warna adalah cara penyajian campuran elemen dasar untuk menghasilkan warna. Ada beberapa Color space yang dikenal dan banyak digunakan. Berikut adalah macam-macam dari Color space:
 - RGB
 - CMY(K)
 - HSV
 - CIE XYZ
 - Lab
 - Yuv
 - YCrCb
4. RGB disebut juga additive color, yang menggunakan warna dasar Red, Green, dan Blue untuk membuat warna.
5. CMY(K) menggunakan elemen dasar Cyan, Magenta dan Yellow. Untuk lebih lengkapnya ditambahkan elemen K (Chroma). CMY(K) adalah kombinasi warna yang digunakan dalam pencetakan (*printing*). Dikenal sebagai subtractive color.
6. HSV mempunyai elemen dasar **Hue** yang menyatakan keluarga warna (dalam satuan derajat), **Saturation** yang menyatakan sensasi/intensitas warna, dan **Value** yang menyatakan derajat keabuan atau terang/gelap gambar.
7. CIE atau CIE 1931 adalah Color space pertama yang didefinisikan secara matematis. CIE merupakan standar yang disusun berdasarkan persepsi manusia dan baik untuk percobaan perbandingan warna.
8. Lab atau CIE Lab merupakan color space yang dikenalkan oleh Photoshop. Merupakan perbaikan dari model CIE. Lab mempunyai elemen L (Luminance), a (range dari hijau ke merah) dan b (range dari biru ke kuning).
9. Yuv menggunakan komponen Y (luminance) dan chrominance (dinyatakan dalam U dan V)
10. Y (luminance) didapat dari nilai RGB :
$$Y = 0.299R + 0.587G + 0.114B$$

11. Sedangkan Chrominance (U dan V) didapatkan dengan rumus berikut.

$$U = B - Y; V = R - Y$$

12. YCrCb adalah versi skala dari YUV dan digunakan dalam JPEG dan MPEG dimana semua komponen bernilai positif.

$$Cb = (B - Y) / 1.772 + 0.5$$

$$Cr = (R - Y) / 1.402 + 0.5$$

13. Color Gamut adalah sebuah range yang menyatakan batasan-batasan warna yang bisa dihasilkan oleh setiap color space.

6.6. Latihan

BAB 7

FITUR PADA CITRA

POKOK BAHASAN:

- ✓ Fitur Warna Citra
- ✓ Histogram Warna
- ✓ Histogram RGB
- ✓ Histogram HSV
- ✓ Histogram CMYK
- ✓ Fitur Bentuk
- ✓ DeteksiTepi
- ✓ Histogram Proyeksi
- ✓ Histogram Sudut

TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat:

- ✓ Mengerti fitur warna pada citra
- ✓ Mengekstraksi fitur warna dan menggunakannya pada aplikasi citra
- ✓ Mengerti fitur bentuk pada citra
- ✓ Mengekstraksi fitur bentuk dan menggunakannya pada aplikasi citra

7.1. Fitur Warna Citra

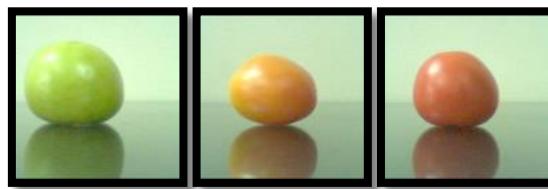
Ada tiga ciri/fitur dasar dalam citra yaitu: warna, bentuk dan tekstur. Pada bab ini dibahas mengenai ciri/fitur warna pada sebuah gambar. Warna bisa digunakan untuk membedakan obyek-obyek pada citra. Bahkan warna pada obyek mempunyai impresi tergantung pada budaya masyarakat.

Beberapa obyek yang bisa dibedakan berdasarkan warnanya contohnya adalah bunga dan buah. Warna-warna pada citra pemandangan (landscape) juga mempunyai ciri-ciri yang spesifik. Uang kertas juga memiliki warna yang unik tiap nominalnya.

Ciri warna pada citra ditunjukkan dengan distribusi warna dan warna dominan pada citra. Distribusi warna menyatakan frekwensi kemunculan warna atau histogram warna. Sedangkan warna dominan bisa dihasilkan dengan cara segmentasi atau clustering warna.

Beberapa aplikasi yang menggunakan fitur warna untuk mengenali atau membedakan objek gambar contohnya adalah aplikasi pengenalan buah, pengenalan bunga, segmentasi foto pemandangan, deteksi kematangan buah, deteksi nominal uang kertas, dan CBIR (Content Based Image Retrieval).

Selain itu ada aplikasi yang banyak kita gunakan yaitu google image search yang dikenalkan oleh google. Konsep pencarinya berdasarkan kemiripan gambar. Proses pencarian gambar ini dinamakan QBIC (Query Based Image Content) dimana salah satu fitur dominan yang digunakan adalah fitur warna. Selain menggunakan fitur warna google juga melakukan pencarian gambar sesuai content dari gambar. Sehingga tidak hanya bisa menghasilkan gambar namun juga informasi yang berkaitan dengan gambar tersebut.



Gambar 7.1 Deteksi Kematangan Buah Tomat

Salah satu aplikasi yang banyak diajarkan adalah pengenalan kematangan buah tomat. Dengan menggunakan fitur warna kita dapat menentukan tingkat kematangan buah tomat (matang, setengah matang, atau mentah). Seperti terlihat pada Gambar 7. 1 terlihat buah tomat mentah didominasi warna hijau. Buah tomat matang didominasi warna merah. Dan setengah matang didominasi warna oranye. Dari warna dominan ini kita bisa tentukan seberapa tingkat kematangan dari buah tomat.

7.2.Histogram Warna

Histogram warna $H(w)$ menyatakan jumlah titik/frekwensi munculnya warna w pada sebuah gambar. Warna w dinyatakan sebagai kombinasi elemen dasar. Ada beberapa histogram warna yang banyak digunakan:

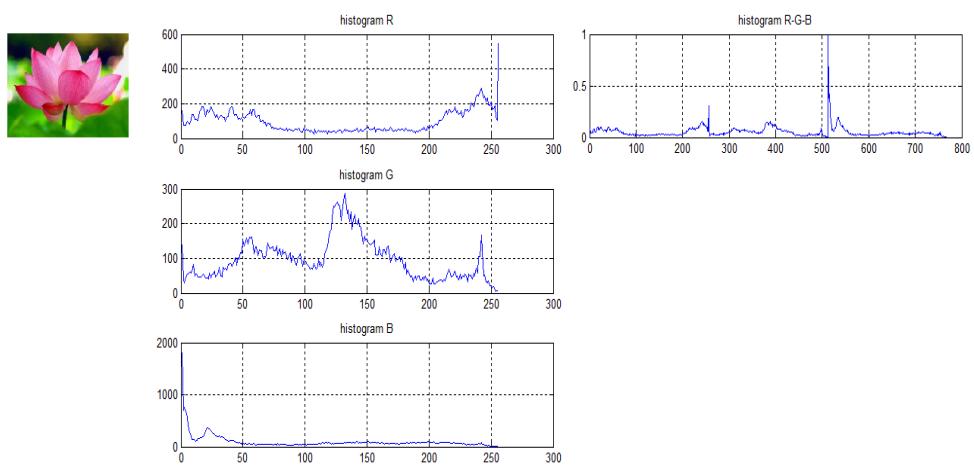
- Histogram RGB
- Histogram HSV
- Histogram CMYK

7.2.1. Histogram RGB

Ada beberapa macam histogram yang menggunakan elemen warna dasar RGB seperti Histogram RGB biasa, Histogram Indeks RGB, dan Histogram Referensi RGB.

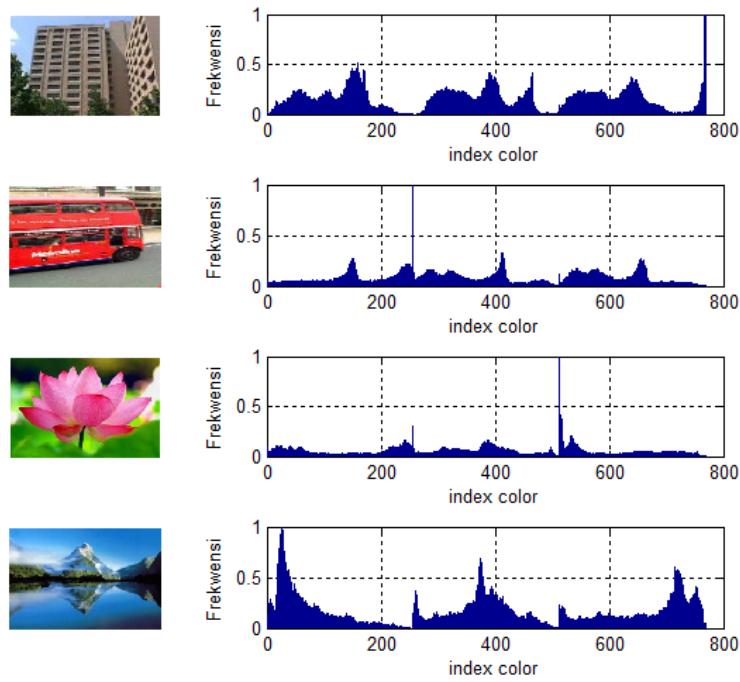
a) Histogram RGB

Setiap pixel mempunyai nilai R, G, dan B yang masing2 bernilai 0-255. Untuk membuat histogram RGB terlebih dahulu kita dapatkan histogram untuk asing-masing layer warna R, G dan B. Kemudian histogram asing-masing Ayer warna tersebut dijadikan satu menjadi Histogram RGB.



Gambar 7.2 Histogram RGB

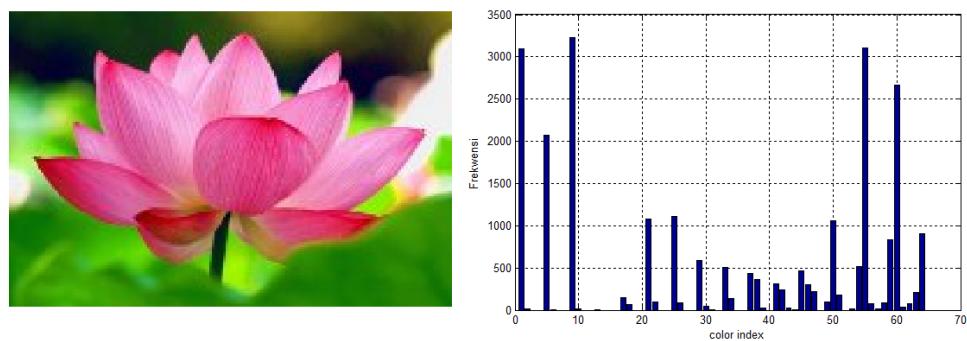
Seperti terlihat pada Gambar 7.2 langkah awal adalah mendapatkan histogram untuk asing-masing Ayer warna (Histogram Red, Histogram Green, Histogram Blue). Kemudian tiga histogram tersebut dijadikan satu menjadi Histogram RGB. Kita dapat membedakan gambar dari histogram RGB yang kita dapat. Namun hasilnya tidak signifikan seperti terlihat pada Gambar 7.3.



Gambar 7.3 Perbedaan Histogram RGB dari Beberapa Gambar

b) Histogram Indeks RGB

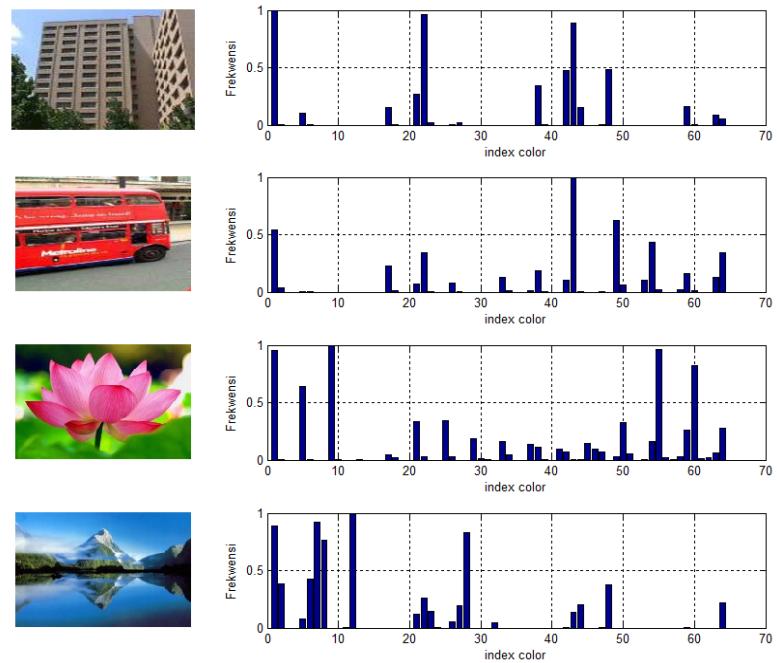
Setiap pixel pada sebuah gambar mempunyai nilai R, G, dan B yang masing2 bernilai 0-255. Bila setiap warna dinyatakan dalam $w(r,g,b)$ maka ada 256^3 indeks warna. Jumlahnya index warna ini terlalu besar untuk diolah. Untuk itu kita bisa menggunakan Histogram Indeks RGB. Caranya adalah dengan kuantisasi terhadap nilai r,g,b menjadi k (4,8,16, dll) sehingga jumlah warnanya menjadi tidak terlalu banyak. Namun jumlah index warna ini harus bisa mencerminkan perbedaan atau ciri dari sebuah gambar.



Gambar 7.4 Histogram Indeks RGB

Seperti terlihat pada Gambar 7.4 histogram pada gambar tersebut telah di kuantisasi sehingga index warna tidak terlalu banyak seperti pada Histogram RGB.

Namun Histogram tersebut masih bisa menunjukkan perbedaan atau ciri warna pada gambar seperti terlihat pada Gambar 7.5.



Gambar 7.5 Perbedaan Histogram Indeks RGB Pada Beberapa Gambar

c) Histogram Referensi RGB

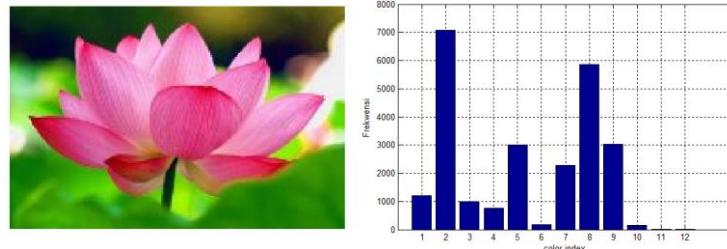
Histogram Referensi ini menggunakan warna referensi misalkan warna dari *paint/36 warna* (Gambar 7.6). Setiap pixel dengan warna $w(r,g,b)$ akan dicari warna referensi yang jaraknya paling dekat (paling mirip). Kemudian histogram dihitung berdasarkan warna referensi tersebut. Kemiripan warna dihitung menggunakan jarak kuadrat.

$$d = \sqrt{(r - r_{ref})^2 + (g - g_{ref})^2 + (b - b_{ref})^2}$$



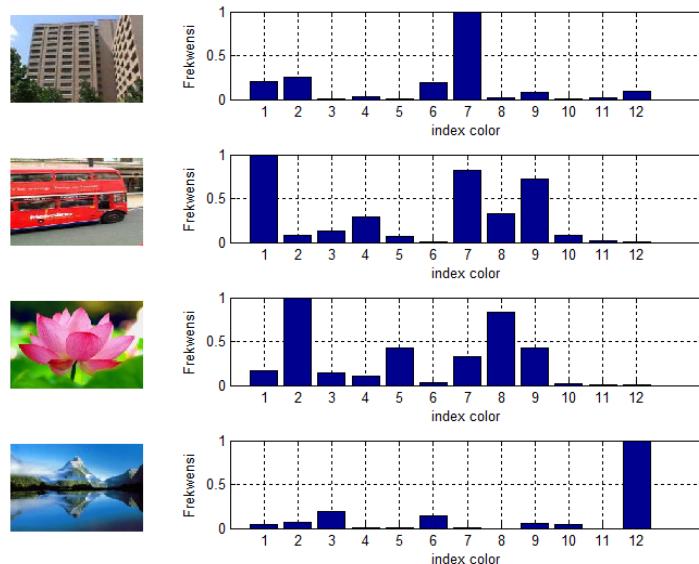
Gambar 7.6 Referensi warna paint/36 warna

Pada Gambar 7.7 histogram menunjukkan jumlah pixel yang muncul pada setiap warna referensi. Pada gambar tersebut digunakan warna referensi 12 warna.



Gambar 7.7 Histogram Referensi 12 Warna

Dengan menggunakan histogram referensi ini bisa didapatkan perbedaan yang cukup signifikan dari beberapa gambar berbeda seperti terlihat pada Gambar 7.8 di bawah ini.



Gambar 7.8 Perbedaan Histogram Referensi dari Beberapa Gambar

7.2.2. Histogram HSV

Untuk Histogram HSV pada dasarnya sama dengan Histogram RGB hanya saja menggunakan elemen dasar warna HSV. Ada beberapa histogram yang biasa digunakan dengan elemen HSV ini seperti histogram Hue Index, histogram HSV, dan Histogram Referensi HSV.

Untuk histogram hue index sama seperti histogram Red, Green atau Blue yang hanya menggunakan satu elemen saja yaitu Hue. Kemudian dilakukan kuantisasi untuk

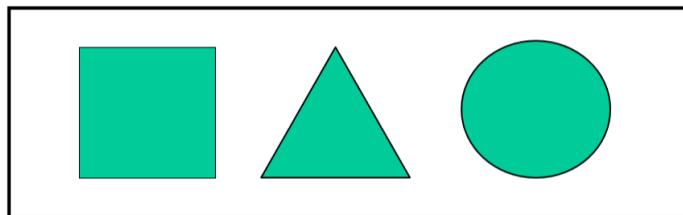
menyederhanakan index warnanya. Untuk Histogram HSV yang digunakan adalah gabungan dari histogram H, S dan V. Sedangkan untuk Histogram HSV dengan Referensi sama seperti Histogram Referensi RGB yang menggunakan referensi warna.

7.3.Fitur Bentuk

Fitur bentuk adalah fitur dasar dalam visual- content pada citra. Dimana setiap obyek gambar dapat dibedakan berdasarkan bentuk dari obyek tersebut. Bentuk dasar dalam geometri adalah bujur sangkar, persegi panjang, segitiga, lingkaran dan ellips. Tetapi bentuk pada gambar jarang sekali menggunakan bentuk dasar. Fitur bentuk dapat diperoleh dengan:

- Deteksi Tepi
- Histogram Proyeksi
- Histogram Sudut

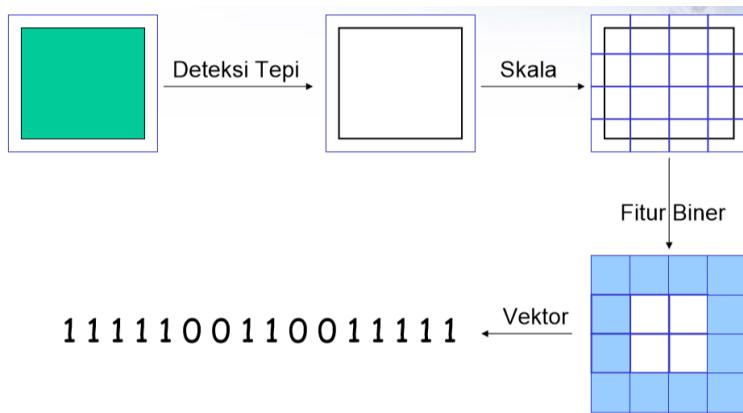
Persoalan dalam fitur bentuk ini adalah bagaimana mendapatkan vektor yang membedakan bentuk-bentuk obyek gambar. Misal kita memiliki object gambar seperti pada Gambar 7.9 dimana terdapat object persegi, segi tiga dan lingkaran bagaimana cara mendapatkan vektor dari tiap object sehingga kita dapat membedakan ketiga object tersebut dengan vector itu.



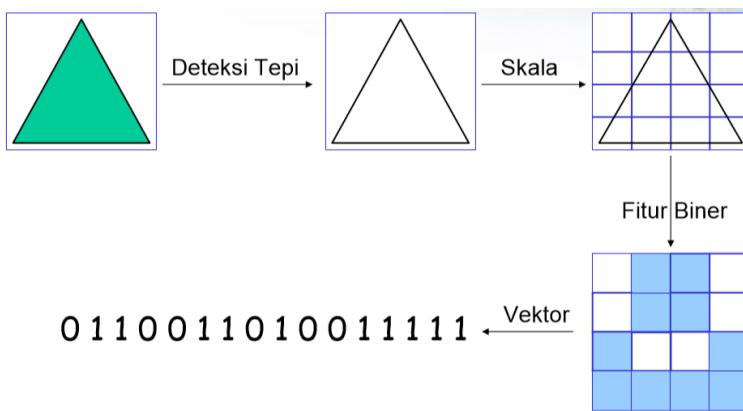
Gambar 7.10 Bentuk Dasar

7.3.1. Deteksi Tepi

Untuk mendapatkan fitur dari sebuah object gambar salah satunya kita dapat menggunakan deteksi tepi. Caranya adalah dengan mendapatkan tepi dari object pada gambar menggunakan deteksi tepi. Kemudian tentukan skala (panjang) fitur, misalkan gambar ukuran 300 x 200 menjadi 30 x 20 maka setiap 10 x 10 menjadi 1 nilai biner. Setiap sel ($S_x \times S_y$) akan bernilai 1 bila ada garis tapi akan bernilai 0 bila tidak ada garis tepi.



Gambar 7.11 Mendapatkan Fitur Object Persegi dengan Deteksi Tepi



Gambar 7.12 Mendapatkan Fitur Object Segi Tiga dengan Deteksi Tepi

Gambar 7.10 dan 13.3 adalah contoh untuk mendapatkan fitur dari sebuah object dengan deteksi tepi. Pada Gambar 7.10 kita mendapatkan vektor dari object persegi adalah : 1111100110011111. Sedangkan vektor untuk object segitiga adalah : 0110011010011001. Dari vektor tersebut kita sudah bisa membedakan kedua object gambar itu.

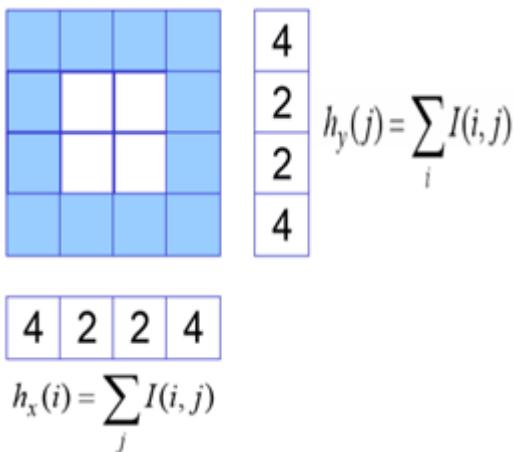
7.3.2. Histogram Proyeksi

Untuk mendapatkan histogram proyeksi kita gunakan integral proyeksi. Integral proyeksi adalah sebuah teknik yang menjumlahkan nilai setiap kolom atau setiap barisnya. Berikut adalah rumusannya:

$$h(j) = \sum_{i=1}^{N\text{baris}} x(i, j)$$

$$h(i) = \sum_{j=1}^{N\text{kolom}} x(i, j)$$

Gambar 7.13 adalah contoh untuk mendapatkan histogram proyeksi dari gambar object persegi atau segi empat.



Gambar 7.13 Histogram proyeksi object persegi

Dari Gambar 7.13 itu maka didapatkan fitur dari object persegi adalah : 42244224. Nilai tersebut didapat dari menggabungkan nilai hasil proyeksi kolom (h_x) dan proyeksi baris (h_y). $h_x = 4224$ dan $h_y = 4224$.

7.4.Ringkasan

1. Ada tiga ciri/fitur dasar dalam citra yaitu: warna, bentuk dan tekstur.
2. Untuk mendapatkan fitur warna kita dapat menggunakan Histogram warna $H(w)$ yaitu histogram yang menyatakan jumlah titik/frekvensi munculnya warna w pada sebuah gambar. Warna w dinyatakan sebagai kombinasi elemen dasar.
3. Ada beberapa histogram warna yang banyak digunakan:
 - Histogram RGB
 - Histogram HSV
 - Histogram CMYK
4. Perbedaan antara pemrosesan citra secara morfologis dengan pemrosesan biasa adalah pada pemrosesan biasa kita memandang sebuah citra sebagai suatu fungsi

intensitas terhadap posisi (x,y). Namun dengan pendekatan morfologi, kita memandang suatu citra sebagai himpunan.

5. Beberapa operasi morfologi yang dapat kita lakukan adalah:
 - a. Dilasi, Erosi
 - b. Opening, Closing
 - c. Thinning, shrinking, pruning, thickening, skeletonizing
6. Fitur bentuk adalah fitur dasar dalam visual- content pada citra. Dimana setiap obyek gambar dapat dibedakan berdasarkan bentuk dari obyek tersebut. Fitur bentuk dapat diperoleh dengan:
 - DeteksiTepi
 - Histogram Proyeksi
 - Histogram Sudut

7.5. Latihan

BAB 8

MORFOLOGI

POKOK BAHASAN:

- ✓ Pengertian Morfologi Citra
- ✓ Struktur Elelemen
- ✓ Dilasi
- ✓ Erosi
- ✓ Opening
- ✓ Closing
- ✓ Hit or Miss Transform
- ✓ Thinning

TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat:

- ✓ Mengerti konsep morfologi pada citra
- ✓ Mengerti beberapa teknik morfologi citra
- ✓ Menerapkan teknik morfologi pada aplikasi citra

8.1. Morfologi

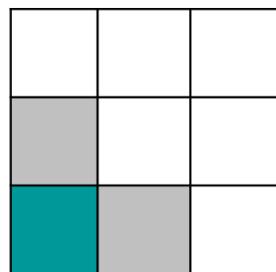
Perbedaan antara pemrosesan citra secara morfologis dengan pemrosesan biasa adalah pada pemrosesan biasa kita memandang sebuah citra sebagai suatu fungsi intensitas terhadap posisi (x,y). Namun dengan pendekatan morfologi, kita memandang suatu citra sebagai himpunan.

Pemrosesan citra secara morfologi biasanya dilakukan terhadap citra biner (hanya terdiri dari 0 dan 1), walaupun tidak menutup kemungkinan dilakukan terhadap citra dengan skala keabuan 0-255. Pada bab ini yang akan kita pelajari adalah pemrosesan morfologi terhadap citra biner.

Secara umum, pemrosesan citra secara morfologi dilakukan dengan cara mem-passing sebuah *structuring element* terhadap sebuah citra dengan cara yang hampir sama dengan konvolusi. *Structuring element* dapat diibaratkan dengan mask pada pemrosesan citra biasa (bukan secara morfologi).

8.2. Struktur Elemen

Struktur elemen dapat diibaratkan dengan mask pada pemrosesan citra biasa (bukan secara morfologi). *Structuring element* dapat berukuran sembarang. *Structuring element* juga memiliki titik poros (disebut juga titik origin/ titik asal/titik acuan). Contoh *structuring element* seperti seperti terlihat pada gambar 12.1 dimana objek S dengan titik poros di (0,0) -> warna merah



Gambar 8.1 Contoh Structuring Element

Beberapa operasi morfologi yang dapat kita lakukan adalah:

- Dilasi, Erosi
- Opening, Closing
- Thinning, shrinking, pruning, thickening, skeletonizing
- All

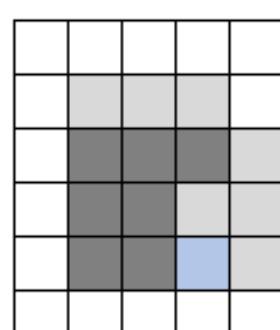
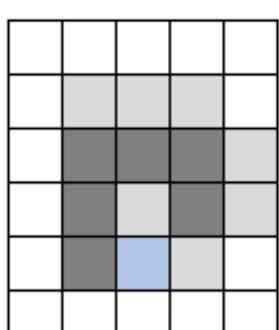
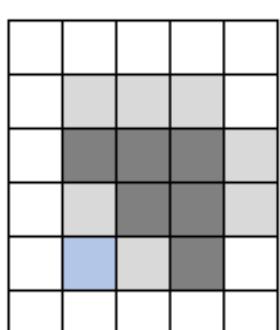
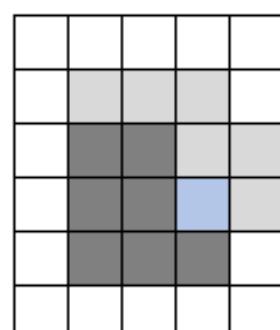
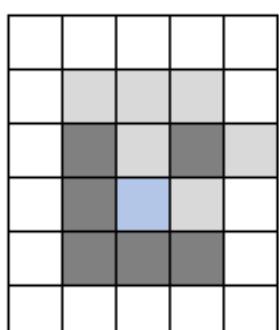
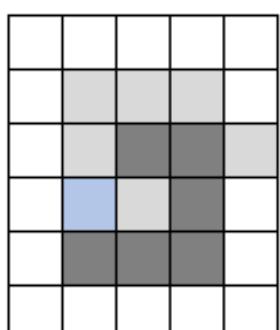
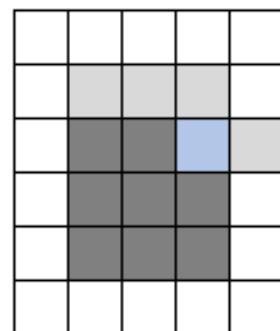
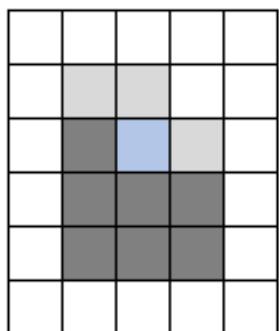
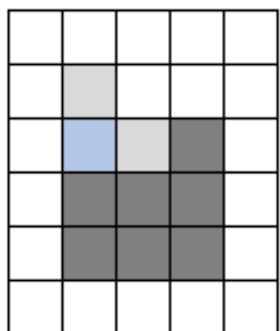
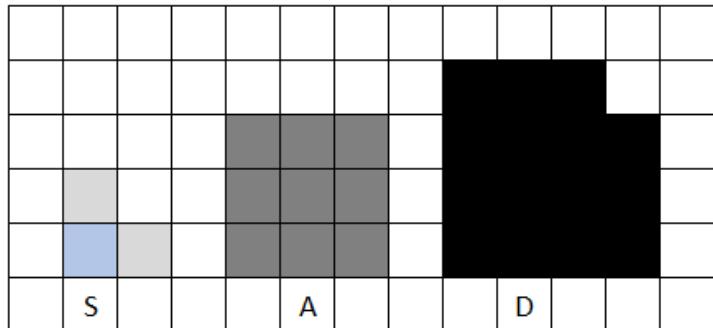
8.3. Dilasi

Dilasi merupakan proses penggabungan titik-titik latar (0) menjadi bagian dari objek (1), berdasarkan structuring elemen S yang digunakan. Berikut adalah rumusan untuk dilasi:

$$D(A, S) = A \oplus S$$

Untuk setiap titik pada A akan dilakukan proses sebagai berikut:

- Letakkan titik poros S pada titik A tersebut
- Beri angka 1 untuk semua titik(x,y) yang terkena/tertimpa oleh struktur S pada posisi tersebut.



Gambar 8.2 Proses dilasi objek S dan A menjadi D

$$\begin{aligned}
 S &= \{(0,0), (0,1), (1,0)\} \\
 &= \{\text{poros}, (+0,+1), (+1,+0)\} \\
 A &= \{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)\}
 \end{aligned}$$

Posisi poros ($(x,y) \in A$)	S_{xy}
(0,0)	$\{(0,0), (1,0), (0,1)\}$
(0,1)	$\{(0,1), (1,1), (0,2)\}$
(0,2)	$\{(0,2), (1,2), (0,3)\}$
.....
(2,2)	$\{(2,2), (2,3), (3,2)\}$

8.4. Erosi

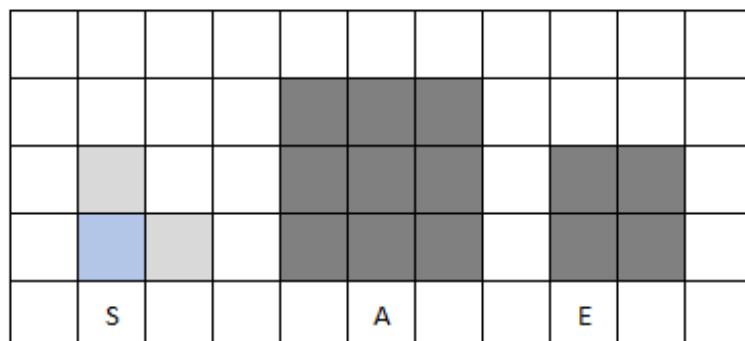
Rumusan untuk erosi adalah sebagai berikut:

$$E(A, S) = A \otimes S$$

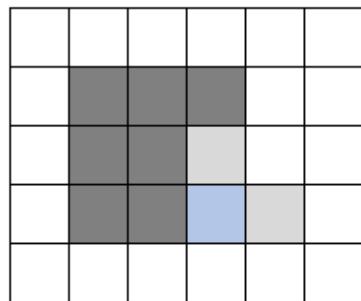
Erosi merupakan proses penghapusan titik-titik objek (1) menjadi bagian dari latar (0), berdasarkan structuring element S yang digunakan.

Cara erosi adalah:

- Untuk setiap titik pada A , lakukan hal berikut:
- Letakkan titik poros S pada titik A tersebut
- Jika ada bagian dari S yang berada di luar A , maka titik poros dihapus / dijadikan latar.



Gambar 8.3 Proses Erosi S terhadap A menjadi E



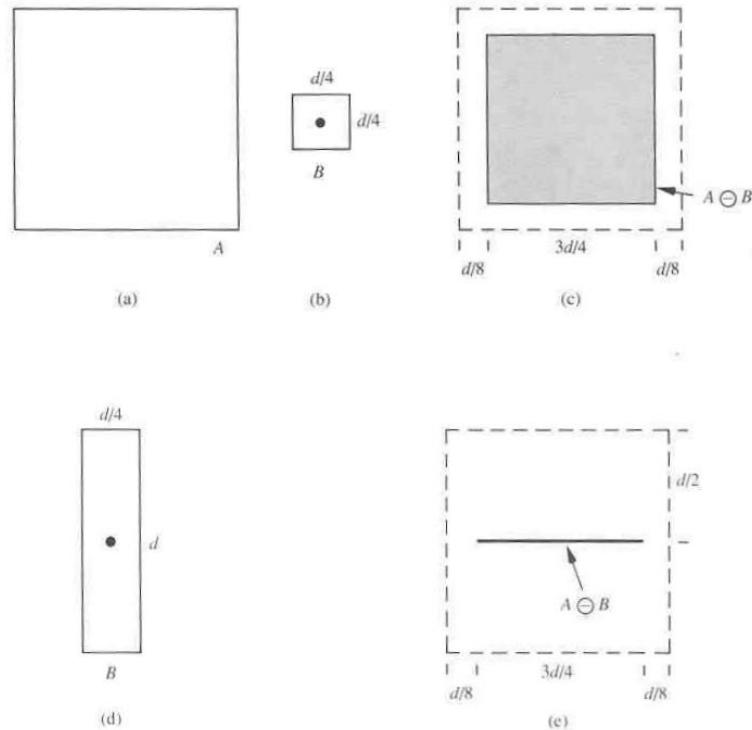
Gambar 8.4 Capture Proses Erosi

$$S = \{(0,0), (0,1), (1,0)\}$$

$$= \{\text{poros}, (+0,+1), (+1,+0)\}$$

$$A = \{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2), (2,0), (2,1), (2,2)\}$$

Posisi poros (x,y) $\in A$)	S_{xy}	Kode
(0,0)	$\{(0,0), (1,0), (0,1)\}$	1
(0,1)	$\{(0,1), (1,1), (0,2)\}$	1
(0,2)	$\{(0,2), (1,2), (0,3)\}$	0
.....	
(2,2)	$\{(2,2), (2,3), (3,2)\}$	0



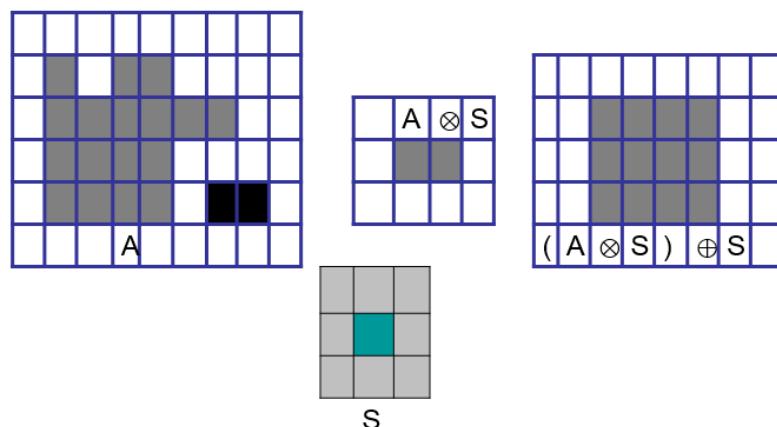
Gambar 8.5 Contoh Erosi

8.5.Opening

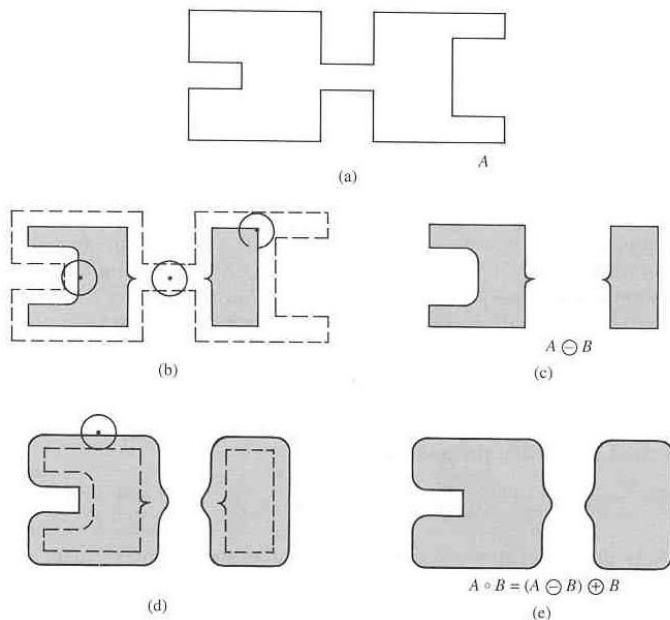
Opening adalah proses erosi yang diikuti dengan dilasi. Efek yang dihasilkan adalah menghilangnya objek-objek kecil dan kurus, memecah objek pada titik-titik yang kurus, dan secara umum men-smooth-kan batas dari objek besar tanpa mengubah area objek secara signifikan

Rumusnya adalah:

$$A \circ S = (A \ominus S) \oplus S$$



Gambar 8.6 Proses Opening



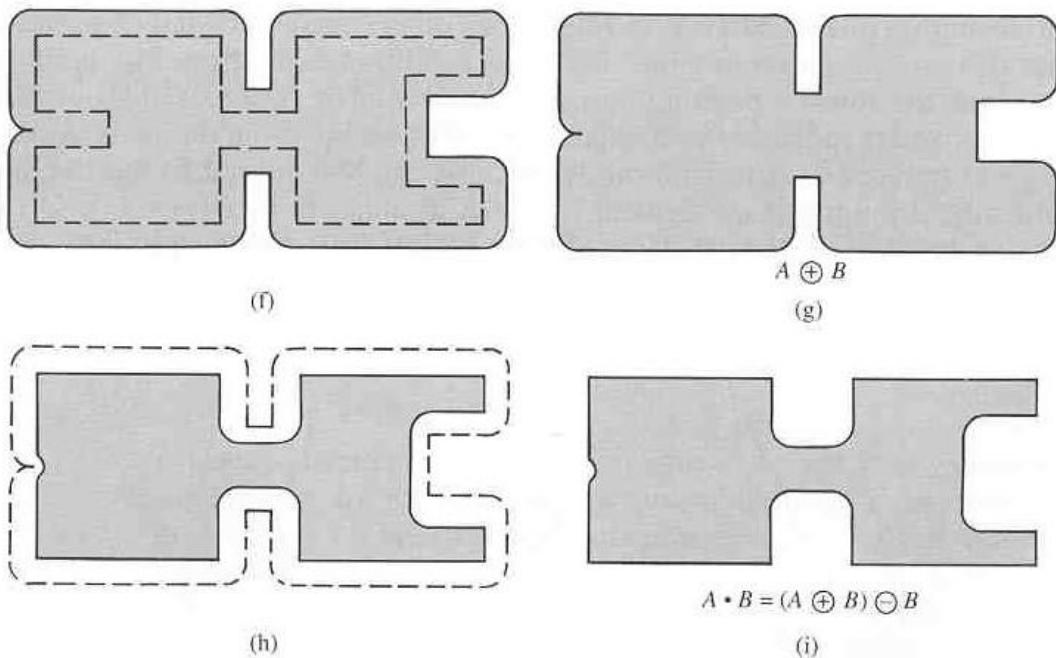
Gambar 8.7 Contoh Opening

8.6.Closing

Closing adalah proses dilasi yang diikuti dengan erosi. Efek yang dihasilkan adalah mengisi lubang kecil pada objek, menggabungkan objek-objek yang berdekatan, dan secara umum men-smooth-kan batas dari objek besar tanpa mengubah area objek secara signifikan. Berikut adalah rumusan untuk proses *Closing*.

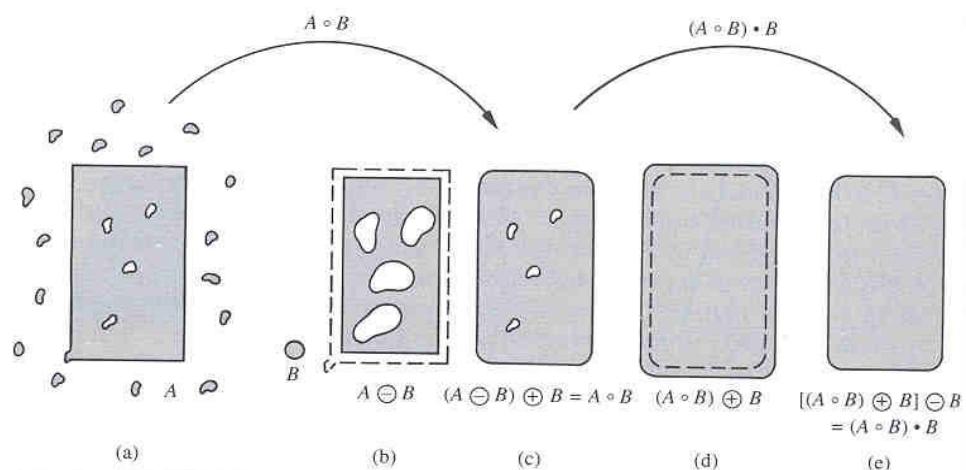
$$A \bullet S = (A \oplus S) \otimes S$$

Gambar 8.6 adalah contoh proses *closing*. Dimana dilakukan proses dilasi terlebih dahulu sehingga gambar akan terlihat semakin tebal. Kemudian baru dilakukan erosi.



Gambar 8.8. Contoh Proses Closing

Gambar 8.7 merupakan contoh perbedaan hasil dari proses openeing dan closing. Gambar (a) alah gambar awal yang memiliki noise. Kemudian hasil erosi terlihat pada (b) dimana noise putih pada tengah gambar terlihat semakin besar. Kemudian gambar (c) hasil dari proses dilasi terhadap gambar (b) yang merupakan hasil dari pening. Sedangkan gambar (d) adalah hasil dari proses dilasi dimana noise pada tengah gambar menghilang. Kemudian dilanjutkan dengan proses erosi yang hasilnya merupakan hasil proses closing (e).



Gambar 8.9 Perbedaan hasil proses opening dan closing

8.7.Hit-or-Miss Transform

Suatu structuring element S dapat direpresentasikan dalam bentuk (S₁,S₂) dimana S₁ adalah kumpulan titik-titik objek (hitam) dan S₂ adalah kumpulan titik-titik latar (putih). Bikut adalah rumusannya:

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

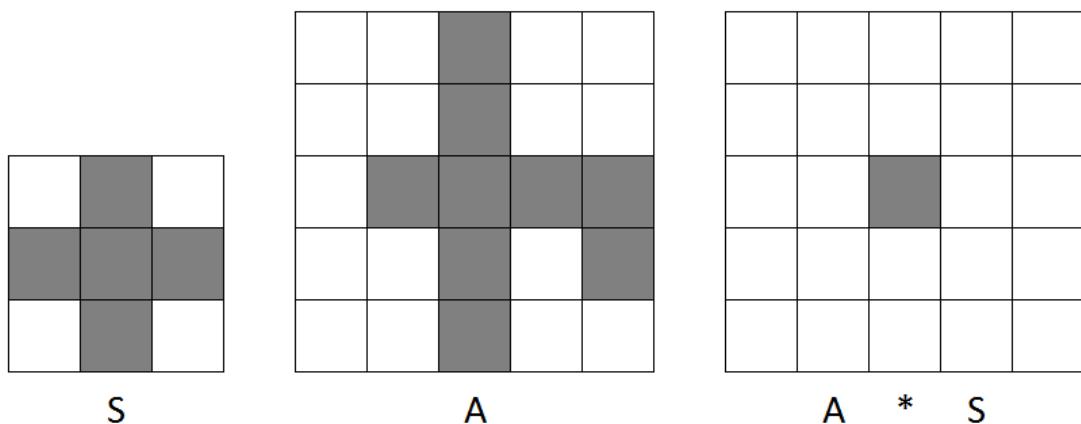
Contoh:

$$S_1 = \{b,e,h\}$$

$$S_2 = \{a,c,d,f,g,h\}$$

a	b	c
d	e	f
g	h	i

Hit-and-misss transform A*S adalah kumpulan titik-titik dimana S₁ menemukan match di A dan pada saat yang bersamaan S₂ juga menemukan match di luar A.



Gambar 8.10 Hit-orMiss Transform

Gambar 8.8 menunjukkan contoh Hit-or-Miss transform dari contoh itu terlihat titik yang match dipertahankan dan titik yang tidak match dihapus.

8.8.Thinning

Merupakan erosi yang dimodifikasi sehingga tidak boleh ada objek yang terpecah. Hasilnya adalah berupa garis yang menunjukkan topologi objek semula. Tujuannya me-remove piksel tertentu pada objek sehingga tebal objek tersebut menjadi hanya satu piksel.

Thinning tidak boleh:

- Menghilangkan end-point
- Memutus koneksi yang ada
- Mengakibatkan excessive erosi

Salah satu kegunaan thinning adalah pada proses pengenalan karakter/huruf Ada banyak cara mengimplementasikan thinning, salah satu diantaranya adalah dengan hit-or-miss transform.

Thinning dapat didefinisikan sebagai:

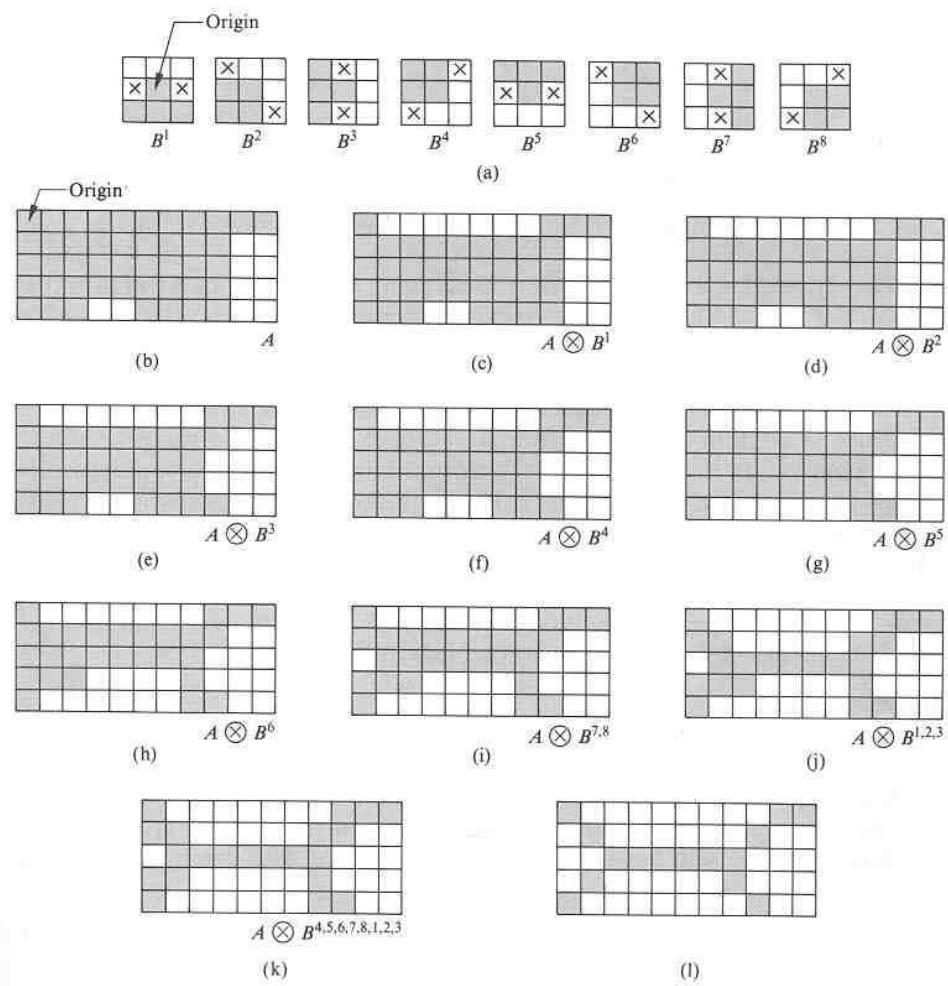
$$\begin{aligned} - \text{ Thinning}(A, \{B\}) &= A - (A * \{B\}) \\ &= A - ((\dots(A * B_1) * B_2) .. B_n) \end{aligned}$$

Dengan $B_1, B_2, B_3..B_n$ adalah Structuring element.

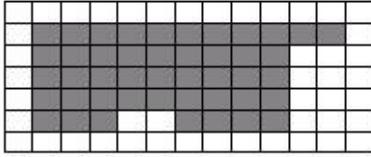
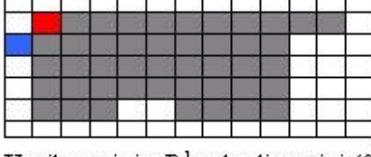
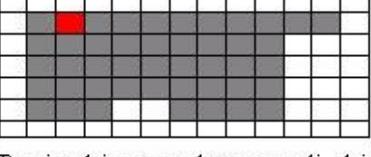
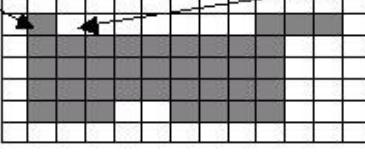
Note:

$A - (A * B)$ berarti kebalikan dari $A * B$

- Yang match dihapus
- Yang tidak match dipertahankan



Gambar 8.11 Contoh Thinning

 <p>Reference: pojok kiri atas adalah posisi (0,0)</p>	<p>B^1 origin: merah Objek: biru Latar: kuning Don't care: arsir</p>  
 <p>Ketika origin B^1 ada di posisi (0,0) Ada bagian biru yang tidak match, Tidak semua match \rightarrow (0,0) dipertahankan</p>	 <p>Ketika origin B^1 ada di posisi (0,1) Bagian biru match semua di objek, Bagian kuning match semua di latar Semua match \rightarrow (0,1) dihapus</p>
	

Gambar 8.12 Proses Thinning

8.9. Ringkasan

8.10. Latihan

8.9.

LATIHAN

11.

12.

BAB 9

APLIKASI PENGOLAHAN CITRA

POKOK BAHASAN:

- ✓ Pengenalan Kematangan Buah Tomat
- ✓ Pengenalan Nilai Nominal Uang Kertas Rupiah
- ✓ Deteksi Arah Gerak Robot Dalam Ruangan
- ✓ Pengenalan Bunga Indonesia

TUJUAN BELAJAR:

Setelah mempelajari bab ini, mahasiswa diharapkan dapat:

- ✓ Mahasiswa bisa membuat aplikasi pengolahan citra berdasarkan fitur warna dan bentuk dari citra tersebut

9.1. Pengenalan Kematangan Buah Tomat

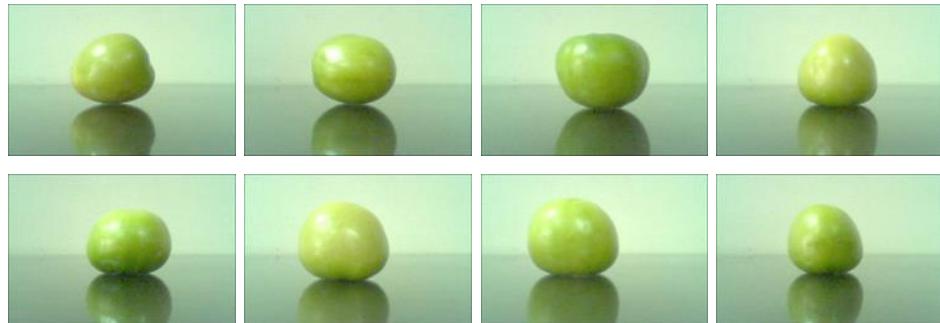
Pengiriman buah tomat dari sentra industri ke pasar atau agen penjualan membutuhkan waktu, sementara tomat juga akan mengalami perubahan warna seiring waktu tersebut. Di sisi lain, produsen buah tomat harus bisa menjamin buah tomat yang diterima oleh agen atau pelanggan dalam keadaan matang atau hampir matang dengan warna merah atau kemerahan, tidak dalam keadaan mentah (berwarna hijau) atau terlalu matang (kecoklatan). Ini sebabnya produsen harus bisa memilah buah tomat berdasarkan warna tomat yang akan dikirimkan sesuai dengan jarak dan waktunya. Untuk jarak yang dekat, tomat yang dikirimkan berwarna merah atau kemerahan. Untuk jarak yang jauh lebih baik tomat yang dikirimkan masih berwarna hijau atau kekuningan.

Pemilahan tomat berdasarkan warna ini bisa dilakukan secara manual, tetapi itu akan membutuhkan banyak waktu dan tenaga. Keberadaan mesin dengan kemampuan mengenali warna buah tomat akan sangat membantu produsen buah tomat untuk memilah dengan waktu yang cepat dan terjamin. Dalam buku ini kita tidak sedang membuat mesin secara lengkap, tetapi membahas bagaimana sistem mengenali kematangan buah tomat berdasarkan warnanya menggunakan kamera. Buah tomat akan dilewatkan di depan kamera, kemudian ditangkap gambarnya dan dikenali apakah buah tomat tersebut masih mentah (hijau), setengah matang (jingga) atau matang (merah), hingga akhirnya bisa dipilah.

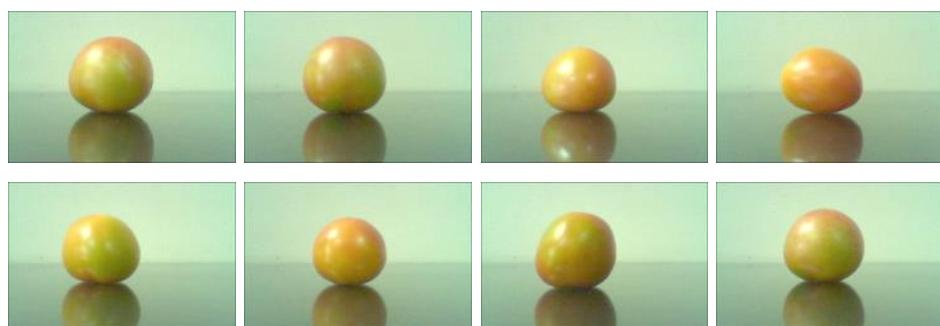
Sistem mengenali kematangan buah tomat ini menggunakan beberapa data referensi sesuai warnanya. Dari data referensi ini kemudian diambil fitur warnanya, untuk kemudian menjadi fitur referensi yang dimasukkan ke dalam database. Berikutnya ketika ada gambar buah tomat, diambil fiturnya lalu dibandingkan dengan fitur referensi untuk dicari kemiripannya. Berdasarkan kemiripan inilah, bisa ditentukan buah tomat tersebut matang, setengah matang atau mentah.

9.1.1. Data Referensi

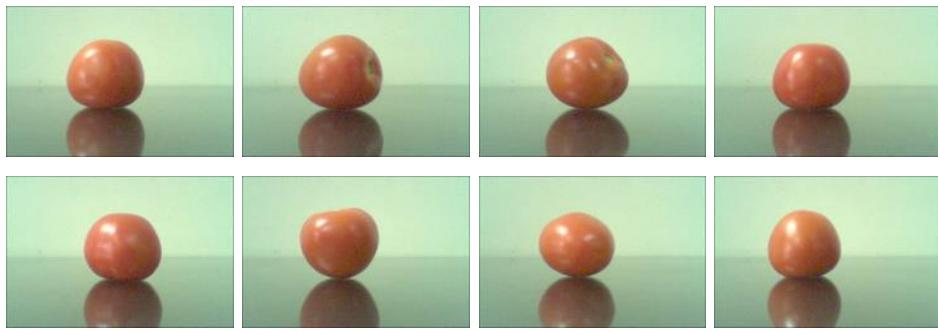
Data referensi yang digunakan adalah 8 gambar untuk masing-masing kematangan buah tomat; 8 gambar untuk buah tomat mentah (hijau), 8 gambar untuk buah tomat setengah matang/campur (jingga) dan 8 gambar untuk buah tomat matang (merah). Data referensi tersebut bisa diambil seperti gambar 9.1, 9.2 dan 9.3 berikut ini.



Gambar 9.1. Data gambar referensi untuk buah tomat mentah (hijau)



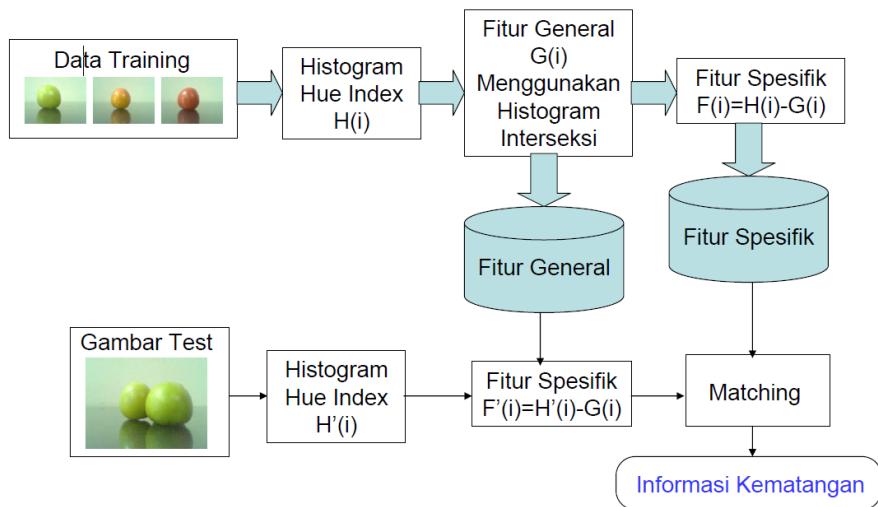
Gambar 9.2. Data gambar referensi untuk buah tomat setengah matang/campur (jingga)



Gambar 9.3. Data gambar referensi untuk buah tomat matang (merah)

9.1.2. Blok Diagram Sistem

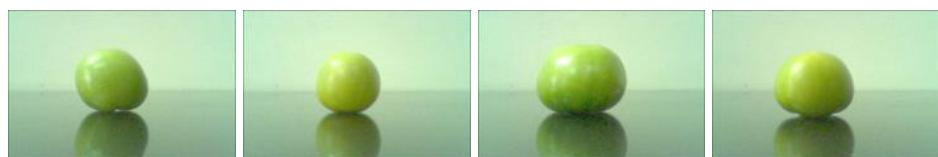
Blok diagram sistem pengenalan kematangan buah tomat adalah seperti terlihat pada gambar 9.4 berikut.

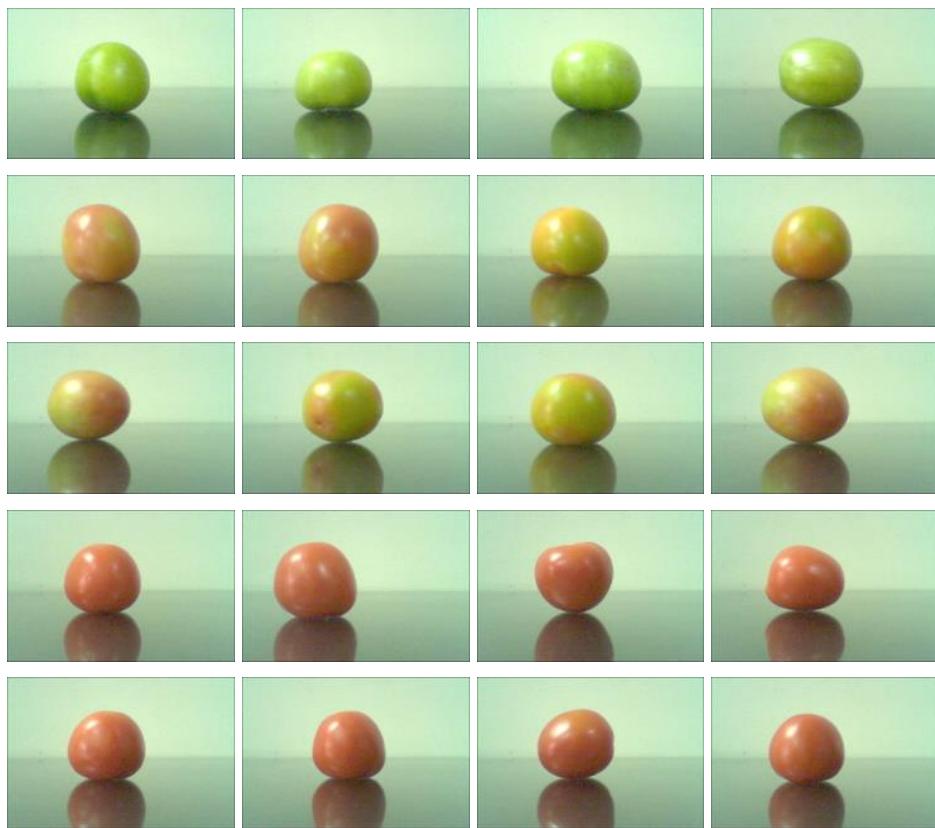


Gambar 9.4. Blok diagram sistem pengenalan kematangan buah tomat

Data training adalah data referensi, yang terdiri dari 24 gambar dengan tiga kategori kematangan.

Data test adalah data yang diambil dengan lingkungan yang sama, bukan dari 24 gambar yang menjadi data referensi. Untuk data test ini, disediakan 24 gambar seperti gambar 9.5 berikut ini.





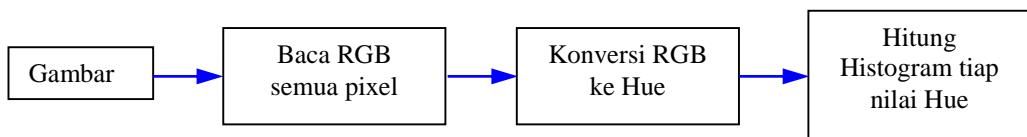
Gambar 9.5. Data test

Hasil pengenalan dari aplikasi pengenalan kematangan buah tomat adalah pemilahan kematangan buah tomat berdasarkan warnanya yaitu:

1. Mentah (Hijau)
2. Campur (Jingga)
3. Matang (Merah).

9.1.3. Ekstraksi Fitur

Di dalam blok diagram dituliskan bahwa fitur yang digunakan adalah Hue Index Histogram. Untuk mendapatkan Hue Index Histogram, maka langkah-langkah yang digunakan adalah seperti gambar 9.6 berikut.



Gambar 9.6. Proses mendapatkan histogram dari hue-index

Konversi RGB ke Hue merupakan bagian dari konversi RGB ke HSV hanya diambil untuk nilai H saja.

$$H = \begin{cases} 60 \cdot \left(\frac{G' - B'}{D} \bmod 6 \right) \\ 60 \cdot \left\{ \left(\frac{B' - R'}{D} + 2 \right) \bmod 6 \right\} \\ 60 \cdot \left\{ \left(\frac{R' - G'}{D} + 4 \right) \bmod 6 \right\} \end{cases}$$

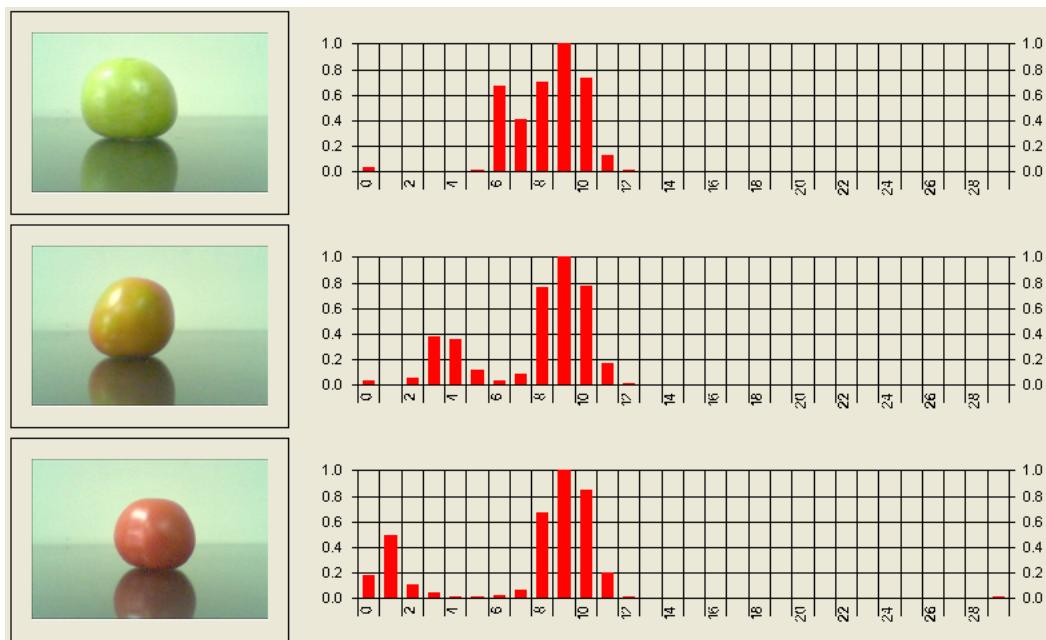
Dimana:

$$R' = \frac{R}{255}, G' = \frac{G}{255}, B' = \frac{B}{255}$$

Hasil dari konversi di atas, nilai H berada dalam range 0-360. Untuk menjadikan lebih simpel, bisa dinormalisasi menjadi range 0-30 dengan cara dibagi 6 dan dibulatkan, sehingga nilai H selalu bulat.

$$H = \left\lceil \frac{H}{6} \right\rceil$$

Setelah itu baru dihitung histogram atau kemunculan setiap nilai H. Contoh hasil histogram dari tiga gambar buah tomat dengan kematangan yang berbeda seperti terlihat pada gambar 9.7 berikut ini.



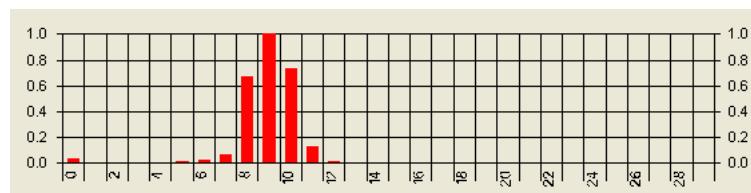
Gambar 9.7. Histogram hue index dari tiga buah tomat dengan kematangan berbeda.

Kalau dilihat hasil histogram dari ketiga gambar di atas, hasilnya mirip. Hal ini disebabkan oleh kesamaan latar belakang. Nilai histogram yang dihitung di atas adalah nilai histogram obyek dan latar belakangnya. Namun di sini latar belakang tidak dihapus pada gambar karena posisi dan besar buah tomat bisa tidak sama. Cara yang dimanfaatkan adalah dengan fitur umum (*general feature*) dan fitur khusus (*specific feature*).

Fitur umum adalah fitur/histogram yang menyatakan informasi yang sama untuk semua gambar. Fitur umum adalah fitur yang ada di semua gambar (seperti latar belakang). Untuk mendapatkan fitur umum ini, semua histogram dicari nilai minimal yang beririsan, atau ditulis dengan:

$$H_{umum} = \min(H)$$

Hasil dari fitur umum untuk tiga gambar di atas adalah seperti gambar 9.8 berikut.

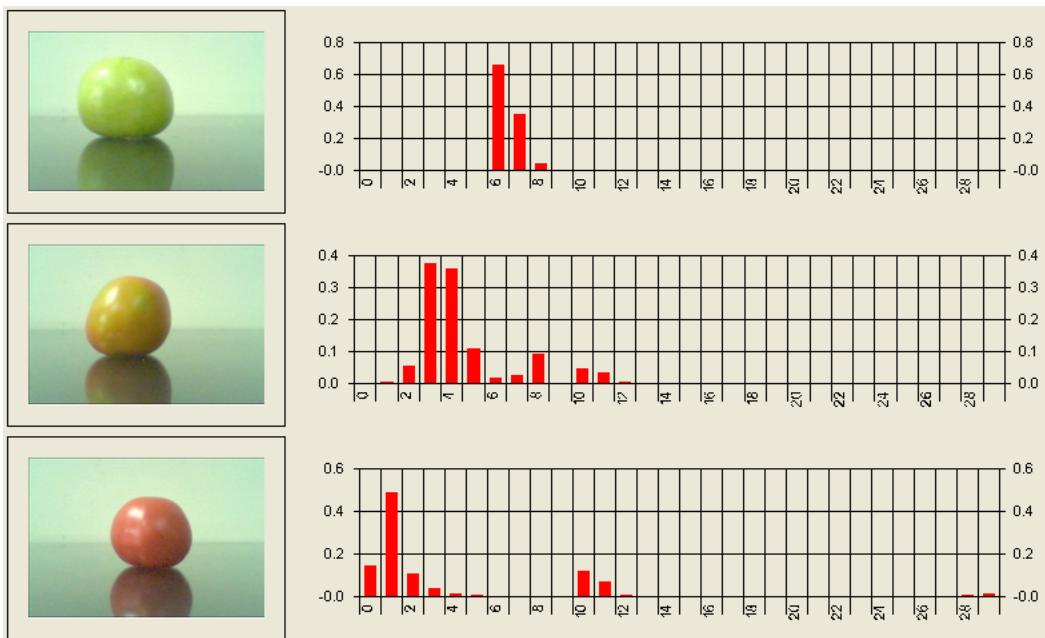


Gambar 9.8. Fitur umum

Hasil tersebut menunjukkan kesamaan yang ada pada semua gambar termasuk latar belakang. Artinya untuk mendapat fitur khusus atau fitur yang berlaku untuk obyek yang berbeda seperti buah tomatnya, maka histogram dari setiap gambar dikurangi dengan fitur umum.

$$H_{obyek} = H - H_{umum}$$

Hasil dari fitur khusus untuk obyek tomat dari tiga gambar contoh di atas adalah seperti gambar 9.9 berikut.



Gambar 9.9. Hasil fitur khusus

Hasil fitur berupa histogram di atas terlihat bisa berbeda untuk setiap kematangan buah tomat. Histogram ini yang bisa digunakan sebagai fitur untuk proses *matching* untuk pengenalan kematangan buah tomat.

9.1.4. Proses Matching

Proses matching digunakan untuk menentukan apakah buah tomat yang tertangkap oleh kamera termasuk kategori mentah, campur atau matang. Bila data fitur khusus pada database disimpan dalam H^k dan fitur buah tomat yang masuk adalah F , maka dicari k yang menghasilkan selisih F dan H^k paling kecil. Selisih dari F dan H^k dirumuskan dengan:

$$d^k = \|F - H^k\|$$

atau

$$d^k = \sqrt{\sum_i (F_i - H_i^k)^2}$$

9.1.4. Hasil Percobaan

Percobaan dilakukan dengan menguji semua data test, dan menampilkan hasilnya sesuai dengan gambarnya.



Gambar 9.10. Hasil percobaan

Dari hasil percobaan di atas, terlihat hasil pengenalannya sangat bagus. Aplikasi bisa mengenali kematangan semua buah yang masuk. Hasil ini bisa menjadi bagus, karena dalam aplikasi ini menggunakan fitur yang terpisah antara fitur umum dan fitur khusus. Pada beberapa hasil yang lain, bisa terjadi aplikasi ini mengalami kesalahan dalam mengenali kematangan buah seperti terlihat pada gambar 9.11 berikut. Kesalahan pengenalan sering terjadi pada warna campur. Kesalahan tersebut bisa disebabkan karena jumlah data yang digunakan sebagai data training tidak mencukupi atau jumlah data vektor yang kurang (dalam percobaan ini menggunakan jumlah data vektor 30 warna). Hasil ini bisa diperbaiki dengan menambah data training.



Gambar 9.11. Salah satu hasil percobaan dengan data uji yang berbeda

9.2. Pengenalan Nilai Nominal Uang Kertas Rupiah

Sistem ini berawal dari ide membuat aplikasi yang bisa membantu orang buta untuk mengenali nilai nominal uang kertas rupiah. Hal ini disebabkan uang kertas rupiah belum semuanya dirancang untuk orang buta dengan kode-kode yang dikenali dengan cara meraba. Uniknya lagi, uang kertas rupiah mempunyai warna yang berbeda pada setiap nominalnya. Sehingga sistem ini menggunakan fitur warna untuk mengenali uang kertas rupiah.

Nominal uang kertas rupiah terdapat 7 macam yaitu 1000 (seribu), 2000 (dua ribu), 5000 (lima ribu), 10000 (sepuluh ribu), 20000 (dua puluh ribu), 50000 (lima puluh ribu) dan 100000 (seratus ribu). Gambar 9.10 berikut menunjukkan contoh gambar dari masing-masing nominal uang kertas tersebut.

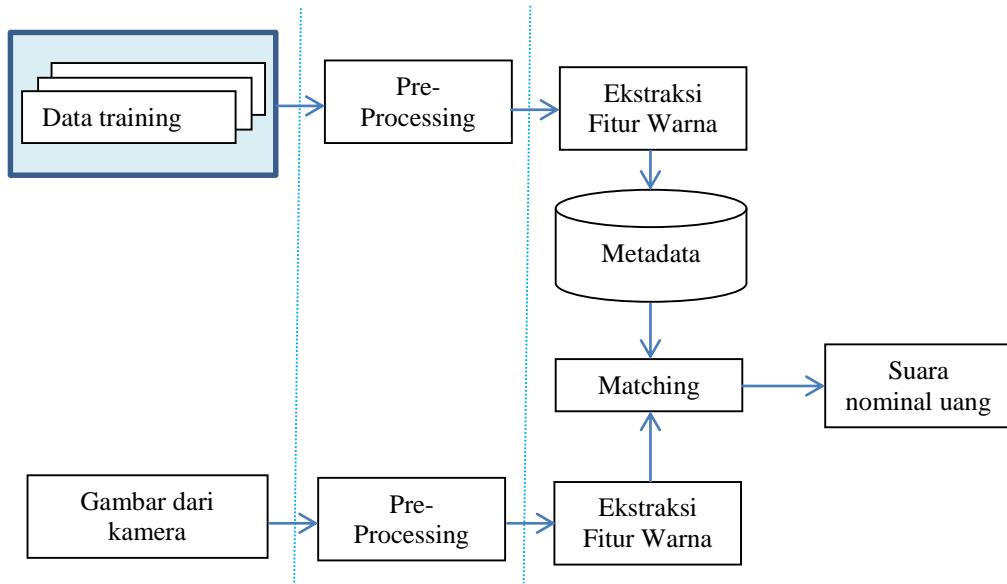
Gambar 9.12 menunjukkan bahwa warna pada setiap uang kertas ternyata berbeda. Warna dominan pada setiap gambar uang kertas memiliki perbedaan, sehingga warna bisa menjadi acuan dalam menentukan nominal uang kertas Rupiah. Ini menjadi dasar berpikir untuk membuat aplikasi untuk mengenali nominal uang kertas menggunakan fitur warna. Dalam hal ini fitur warna yang digunakan adalah histogram RGB-index.



Gambar 9.12. Contoh uang kertas rupiah

9.2.1. Blok Diagram

Blok diagram dari aplikasi untuk mengenali nominal uang kertas menggunakan fitur warna adalah seperti pada gambar 9.11 berikut.



Gambar 9.13. Blok diagram aplikasi mengenali nominal uang kertas

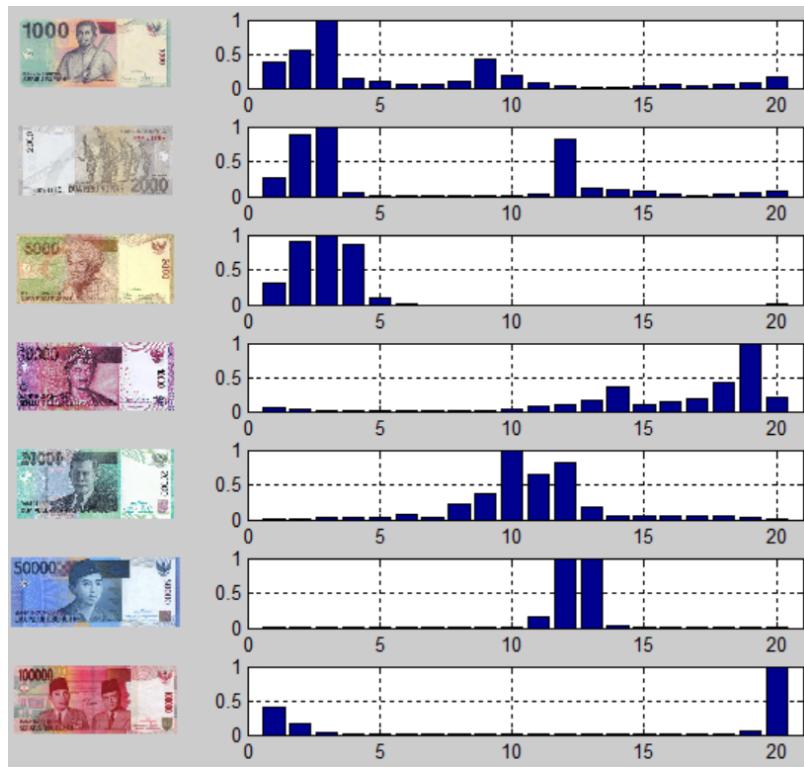
Ada beberapa proses yang ada di dalam blok diagram tersebut: data training, pre-processing, ekstraksi fitur dan proses matching untuk mengenali nilai nominalnya.

9.2.2. Data Training

Data training yang digunakan adalah data seperti pada gambar 9.10. Untuk aplikasi ini, setiap nominal mata uang menggunakan sebuah gambar data training. Hal ini disebabkan oleh perbedaan warna yang cukup jauh pada masing-masing nominal mata uang. Memang ada beberapa nominal mata uang yang warnanya agak dekat seperti uang 10000 (sepuluh ribu) dan 100000 (seratus ribu). Namun itu masih bisa dibedakan dengan cukup jelas.

9.2.3. Fitur Warna

Fitur warna yang digunakan untuk mewakili masing-masing pola warna uang kertas adalah histogram Hue-index. Histogram hue-index adalah sebuah histogram yang menyatakan kemunculan setiap warna yang dinyarakan dalam indeks hue. Beberapa contoh histogram hue-index dari data training adalah:

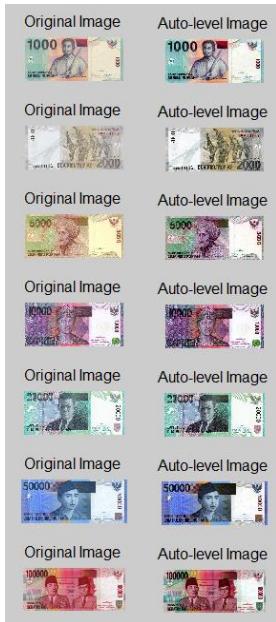


Gambar 9.14. Histogram dari semua data training

Gambar 9.14 menunjukkan bahwa ada perbedaan nilai-nilai maksimum pada masing-masing gambar uang yang menunjukkan perbedaan dominasi warna. Hal ini menjadi salah satu harapan untuk mendapatkan hasil pengenalan yang maksimal.

9.2.4. Preprocessing

Proses *preprocessing* dilakukan untuk menghasilkan warna yang lebih akurat dengan level yang memenuhi semua ruang histogram. Proses *processing* yang dilakukan adalah “auto level”. Proses auto-level yang digunakan dalam aplikasi ini adalah proses auto level yang tidak mempengaruhi hue dan saturation terlalu kuat, hanya berpengaruh pada value.



Gambar 9.15. Hasil dari auto level

Proses auto-level adalah sebagai berikut:

- (1) Baca R-G-B dari setiap titik
- (2) Hitung nilai grayscale X dengan rumus berikut:

$$x = \frac{r + g + b}{3}$$

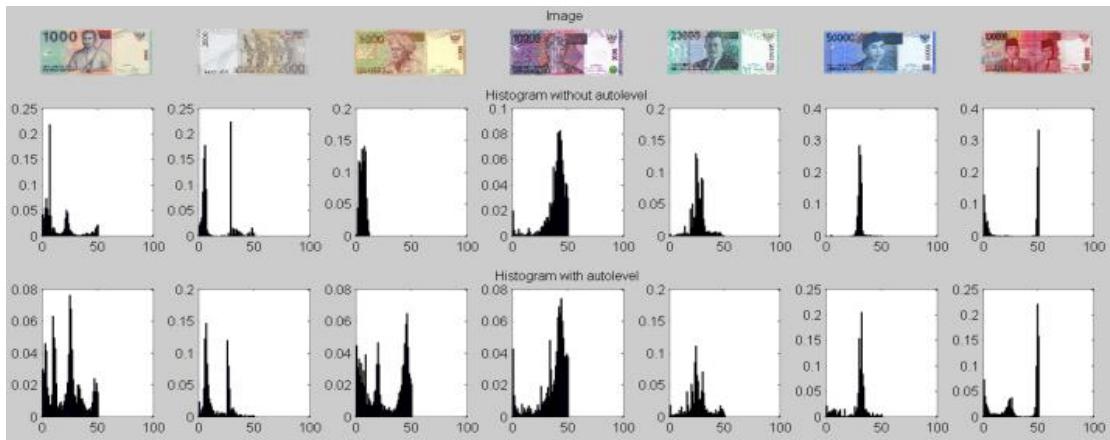
- (3) Hitung nilai minimum x_{min} dan maksimum x_{max} dari X
- (4) Update nilai level dari masing-masing R, G dan B.

$$r' = \frac{255}{(x_{max} - x_{min})}(r - x_{min})$$

$$g' = \frac{255}{(x_{max} - x_{min})}(g - x_{min})$$

$$b' = \frac{255}{(x_{max} - x_{min})}(b - x_{min})$$

Hasil dari proses auto-level dari gambar-gambar training terlihat pada gambar 9.15. Gambar 9.16 menunjukkan perubahan nilai histogram dari gambar-gambar sebelum proses auto-level dan setelah proses auto level. Terlihat tidak terlalu berbeda, tetapi diperhatikan pada gambar nilai uang 5ribu-an, 20ribu-an dan 100ribu-an, akan terlihat perbedaannya.



Gambar 9.15. Perubahan nilai histogram sebelum dan sesudah auto level

9.2.5. Proses Template Matching

Template matching adalah sebuah sistem untuk membandingkan data masukan (data query) q_i dan data training p_i menggunakan jarak Euclidian D yang dituliskan sebagai berikut:

$$D = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

Proses template matching ini mencari data training yang mempunyai jarak yang paling kecil (minimum). Bila ada data training yang mempunyai jarak minimum dengan data query, maka perlu ditentukan/dipilih apakah data yang pertama, terakhir dan dipilih acak.

9.2.6. Hasil Percobaan

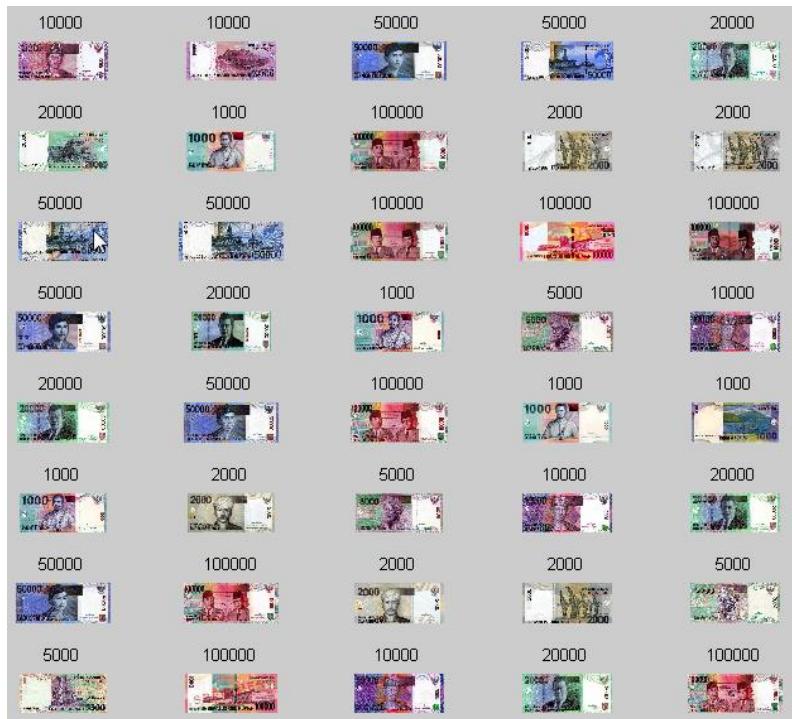
Ada dua percobaan yang dilakukan: (1) Percobaan tanpa proses auto-level, dan (2) percobaan dengan auto-level. Kemudian hasilnya dibandingkan dari jumlah gambar yang dikenali benar.

Hasil percobaan tanpa auto level menggunakan 40 gambar yang berbeda seperti terlihat pada gambar 9.16. Terdapat 9 gambar yang dikenali salah. Kesalahan yang paling banyak ada di uang 2000 dan 5000. Hal ini disebabkan adanya kemiripan level pada gambar-gambar tersebut. Hal ini akan bisa diperbaiki dengan proses auto level.



Gambar 9.16. Hasil percobaan tanpa autolevel

Hasil percobaan dengan auto level menggunakan 40 gambar yang sama dengan percobaan tanpa auto level untuk menunjukkan perbedaan. Hasil percobaan dengan auto level ditunjukkan pada gambar 9.17. Hasil menunjukkan bahwa tidak ada kesalahan pengenalan. Semua gambar yang dicobakan bisa dikenali dengan nominal yang benar.



Gambar 9.17. Hasil percobaan dengan auto level

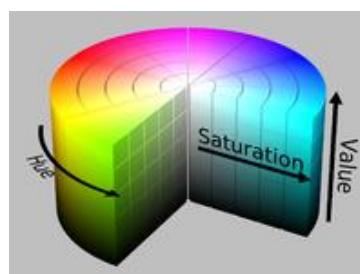
9.3. Pengenalan Bunga indonesia

Indonesia adalah negara yang memiliki kekayaan alam yang cukup besar di dunia termasuk kekayaan alam berupa flora endemik Indonesia. Namun masyarakat Indonesia sendiri kurang mengenal bunga-bunga tersebut. Hal ini dapat dibuktikan dengan adanya kuisioner yang telah dibuat. Misalkan seseorang melihat gambar bunga ditepi jalan, seringkali orang tersebut tidak mengetahui nama bunga tersebut. Bahkan untuk mengetahui nama bunga Indonesia melalui gambarnya, sistem pencarian seperti Google pun tidak banyak membantu. Oleh karena itu, untuk memudahkan mengetahui informasi tentang bunga yang ada di Indonesia, dibuat aplikasi pengenalan flora Indonesia.

Pada aplikasi ini terdapat dua cara identifikasi objek bunga, yaitu dengan cara menginputkan teks (nama bunga) yang akan dicari dan hasilnya berupa tampilan gambar dari objek (bunga) beserta informasi pendukung, serta dengan cara menginputkan gambar objek baik melalui kamera maupun file gambar yang sudah ada dan hasilnya berupa nama bunga tersebut beserta informasi lain mengenai bunga tersebut. Informasi dari bunga meliputi nama latin, nama lokal, deskripsi, penyebaran dan ekologi bunga tersebut. Selain itu, aplikasi ini juga memberikan daftar data flora yang dapat dibaca oleh pengguna.

9.3.1. HSV (Hue, Saturation, Value) Color Space

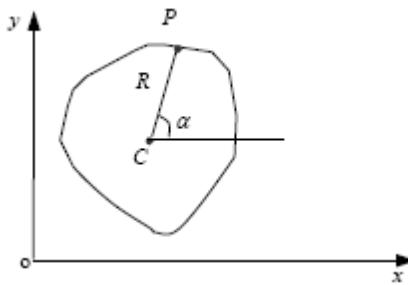
HSV merupakan model warna yang terdiri dari tiga komponen. Yang pertama adalah *Hue*, yang merepresentasikan warna sesungguhnya dari masing-masing piksel dan mempunyai nilai 0 sampai 2π . Komponen kedua adalah *Saturation* yang menyatakan tingkat kemurnian suatu warna, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna. Nilai *Saturation* berkisar antara 0 sampai 1. Kemudian komponen ketiga adalah *Value* atau juga disebut *Intensity* merupakan atribut yang menyatakan banyaknya cahaya. Dengan nilai *Intensity* yang rendah, meskipun dengan nilai *Saturation* yang tinggi, warna akan mendekati nilai abu-abu. *Saturation* memberikan nilai kemurnian warna tetapi mata manusia kurang sensitif dengan variasinya jika dibandingkan dengan variasi pada *Hue* dan *Intensity*. Untuk mendapatkan model warna HSV perlu dilakukan konversi dari model warna RGB karena gambar inputan dari layar perangkat keras atau kamera memiliki default model warna RGB. Gambar 9.18 adalah ilustrasi model warna HSV.



Gambar 9.18 Model warna HSV

9.3.2. CCD (Centroid Contour Distance)

Centroid-Contour Distance (CCD) merupakan suatu metode yang digunakan untuk melakukan ekstraksi fitur bentuk dari suatu gambar. Metode ini dapat merepresentasikan ciri bentuk secara umum dari suatu gambar. Satu kelemahan dari metode ini adalah CCD tidak dapat mengatasi perubahan rotasi. Metode ini menghitung jarak R dari centroid suatu bentuk dengan batasan tepinya sesuai dengan kelipatan dari sudut α . Pada Gambar 9.19, jarak R adalah jarak antara titik C (centroid) dengan titik P (bagian tepi objek) dan sudut α adalah sudut yang dibentuk antara 2 garis R yang berdekatan dan memiliki kelipatan yang sama untuk setiap sudut yang dibentuk. Sebagai contoh bila nilai kelipatan α misalkan 10° , maka akan terdapat 36 jarak R (R_1, R_2, \dots, R_{36}).



Gambar 9.19. Pendekatan Centroid-Contour Distance

9.3.3. Sobel Edge Detection

Deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek gambar. Suatu titik (x,y) dikatakan sebagai tepi (edge) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangga. Tujuannya untuk menandai bagian yang menjadi detail citra, memperbaiki detail dari citra yang kabur yang terjadi karena error atau adanya efek dari proses akuisisi citra dan bagaimana mendapatkan garis-garis tepi suatu obyek gambar.

Metode Sobel merupakan pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyanga. Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan Hight Pass Filter. Kelebihan dari metode sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi.

$$\text{Kernel vertikal} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\text{Kernel horisontal} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

9.3.4. Pengumpulan Data

Aplikasi identifikasi bunga ini memerlukan data-data pendukung sebelum dilakukan perancangan sistem. Data-data tersebut mencakup gambar-gambar bunga Indonesia dan informasi yang meliputi nama lokal bunga, nama latin bunga, deskripsi umum bunga, persebaran bunga, dan ekologi bunga tersebut. Data gambar bunga didapat dari situs ensiklopedia Wikipedia, buku ensiklopedia flora khas Indonesia dan sumber-sumber lain dari internet. Bunga-bunga dapat dikategorikan berdasarkan tempat

persebarannya atau tempat dimana bunga tersebut banyak dijumpai di Indonesia. Contoh data gambar bunga terlihat pada tabel 9.1. Tabel 9.1 menunjukkan 10 contoh gambar bunga, tetapi data sebenarnya adalah 43 jenis bunga yang bisa dimanfaatkan.

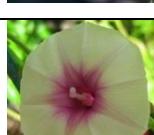
9.3.5. Blok Diagram

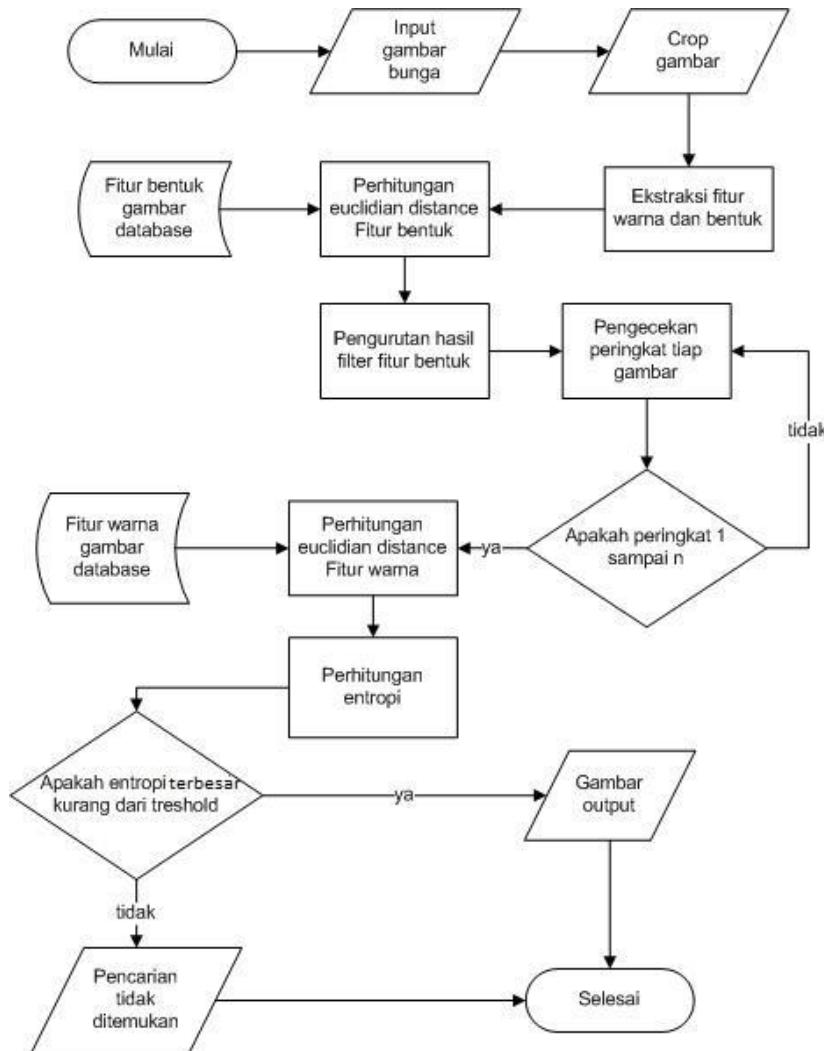
Diagram alir dari aplikasi identifikasi bunga adalah seperti terlihat pada gambar 9.20 berikut. Diagram alir di atas menjelaskan alur proses penggunaan aplikasi dari awal hingga akhir.

1. User memasukkan gambar bunga yang akan dicari informasinya.
2. User melakukan *crop* pada gambar bila diperlukan.
3. Sistem aplikasi melakukan ekstraksi fitur-fitur gambar yang dimasukkan oleh user.
4. Sistem aplikasi menghitung jarak fitur gambar masukan dengan Euclidian Distance terhadap tiap fitur bentuk gambar yang ada dalam database.
5. Untuk gambar yang memiliki jarak terkecil di peringkat 1 sampai n akan dihitung jarak fitur warna terhadap gambar input dan dihitung .
6. Bunga yang memiliki nilai entropi terbesar dan kurang dari treshold yang ditentukan merupakan output hasil pencarian.
7. Jika bunga dengan entropi terbesar namun melebihi treshold yang ditentukan, maka pencarian dinyatakan tidak menemukan hasil atau bunga yang dicari tidak termasuk flora Indonesia.

Tabel 9.1. Contoh data gambar bunga di Indonesia

Indeks	Gambar				
1					
2					

3					
4					
5					
6					
7					
8					
9					



Gambar 9.20 Diagram alir aplikasi

9.3.6. Ekstraksi Fitur

(1) Ekstraksi Fitur Warna

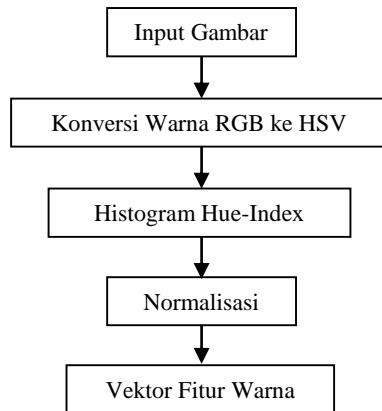
Secara garis besar, proses ekstraksi fitur warna digambarkan dalam diagram blok pada Gambar 9.21. Proses konversi warna RGB ke HSV dilakukan tentunya mengambil nilai R, G, B masing-masing piksel dari gambar, kemudian setelah didapat nilai R, G, B dari tiap piksel tersebut baru kemudian dilakukan konversi warna RGB menjadi HSV. Berikut adalah rumus yang digunakan untuk mengkonversi nilai-nilai RGB menjadi HSV:

$$H = \arccos \left(\frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right)$$

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

$$V = \frac{1}{3}(R+G+B)$$

Dimana R adalah nilai *red*, G adalah nilai *green* dan B adalah nilai *blue* dari model warna RGB. Sedangkan H adalah nilai *Hue* , S adalah nilai *Saturation* dan V adalah *Value* atau nilai *Intensity* pada model warna HSV.

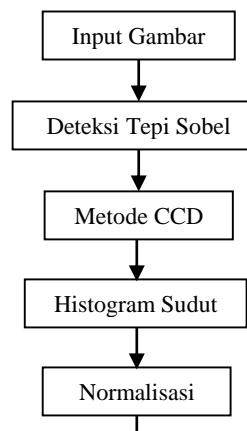


Gambar 9.21. Diagram Blok Ekstraksi Fitur Warna

Setelah proses konversi berikutnya menentukan histogram Hue-index. Untuk aplikasi ini, nilai *Saturation* dan *Value* tidak digunakan karena keduanya tidak dianggap merepresentasikan warna dari gambar. Hanya elemen *Hue* saja yang dihitung sebagai histogram warna. Karena nilai *Hue* adalah nilai yang merepresentasikan warna di tiap piksel gambar. Nilai terdiri dari 0 hingga 360 sehingga histogram warna yang didapat adalah array satu dimensi sepanjang 361 data. Histogram warna berdasarkan nilai *Hue* dari gambar biasa disebut dengan *Hue Index*. Histogram Hue-index menyatakan berapa pixel yang mempunyai indeks hue tertentu dari 0 s/d 360. Pada aplikasi *ini* diambil 40 nilai hue index, artinya dari 360° diambil 40 satuan nilai indeks, dengan demikian setiap nilai indexnya mewakili 9° nilai Hue.

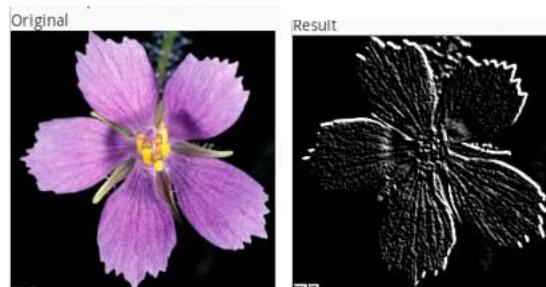
(2) Ekstraksi Fitur Bentuk Menggunakan CCD

Secara garis besar, proses ekstraksi fitur bentuk digambarkan dalam diagram pada Gambar 9.22.



Gambar 9.22. Diagram Blok Ekstraksi Fitur Bentuk

Langkah pertama adalah deteksi tepi (*Edge Detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek gambar. Suatu titik (x,y) dikatakan sebagai tepi dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangga. Pada aplikasi ini, proses deteksi tepi menggunakan metode Sobel. Gambar 9.23 merupakan hasil deteksi tepi metode Sobel yang dilakukan pada gambar bunga.



Gambar 9.23. Hasil deteksi tepi metode sobel

Setelah dilakukan deteksi tepi pada gambar, selanjutnya dilakukan perhitungan algoritma CCD. Gambar 9.24 merupakan ilustrasi penerapan rumus CCD pada gambar bunga yang sudah dideteksi tepi. Perhitungan algoritma CCD sebagai berikut:

- 1) Menentukan koordinat polar untuk titik P dengan (r, θ) . Dimana r dimulai dari satu dan jaraknya terus bertambah sebesar 1 piksel, $r = r + 1$. Nilai awal θ diset sama dengan 0° . Kemudian digunakan kelipatan sudut θ sebesar 10° hingga 360° .
- 2) Hitung posisi koordinat titik P setiap kelipatan sudut θ . Sehingga di setiap jarak r terdapat 36 titik dari setiap sudut θ . Untuk menentukan posisi titik P dihitung dengan rumus:

$$X = X_0 + r * \cos\theta$$

$$Y = Y_0 + r * \sin\theta$$

Keterangan:

X = Posisi titik P pada sumbu X

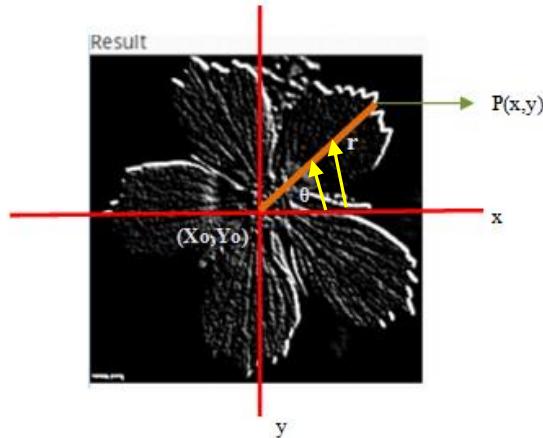
Y = Posisi titik P pada sumbu Y

X_o = Posisi titik tengah pada sumbu X

Y_o = Posisi titik tengah pada sumbu Y

r = Kelipatan jarak ($r = 1, 2, 3, \dots, \text{dst}$)

θ = Kelipatan sudut ($\theta = 10^\circ, 20^\circ, \dots, 360^\circ$)



Gambar 9.24. Ilustrasi penerapan rumus CCD pada gambar bunga

Baru kemudian setiap posisi koordinat titik P tersebut dilakukan pengecekan terhadap piksel tetangga terdekatnya. Apabila titik P dan tetangganya berwarna putih, maka titik P tersebut merupakan bagian tepi. Ketika titik P merupakan bagian tepi objek, maka nilai r dari titik tersebut akan disimpan sebagai fitur bentuk. Tabel 9.2 merupakan ilustrasi metode CCD saat diterapkan pada gambar bunga.

Tabel 9.2. Contoh perhitungan CCD

	0°	10°	20°	30°	40°	50°	60°	70°	80°	90°	350°
r=1	0	0	0	0	0	0	0	0	0	0	0
r=2	1	0	0	0	1	0	0	1	0	0	0
r=3	1	0	1	0	0	0	0	0	0	1	0
r=4	1	0	1	0	0	0	1	0	0	0	0
r=5	0	0	0	0	0	1	0	0	0	0	0
r=6	1	0	0	0	0	0	0	0	0	0	0
r=7	0	1	0	0	0	0	0	0	0	0	0

	0°	10°	20°	30°	40°	50°	60°	70°	80°	90°	350°
....	0
r=N	0	0	0	0	0	0	0	0	0	0	0

Pada ilustrasi diatas, nilai 1 berarti titik P berwarna putih, sedangkan nilai 0 berarti titik P berwarna hitam. Pada $\theta=10^\circ$ titik P saat $r=2$ bernilai 1, kemudian dilakukan pengecekan dengan titik P di $r+1$, jika titik P pada $r+1$ juga bernilai 1, maka titik P pada $r=2$ adalah bagian tepi objek. Jika nilai 1 yang berdekatan lebih dari 2, maka yang diambil adalah r yang terkecil. Untuk setiap sudut θ , diambil masing-masing 3 titik yang merupakan tepi. Jika titik tepi kurang dari 3, maka sisanya bernilai 0. Jika titik tepi lebih dari 3, maka yang diambil adalah 3 titik pertama, sehingga terdapat 108 data.

- 3) Data sebanyak 108 tersebut disimpan dalam bentuk array 1 dimensi untuk kemudian dilakukan normalisasi dengan nilai pembagi r maksimum.

(3) Penggabungan dan Normalisasi

Hasil vektor fitur dari gambar merupakan gabungan dari hasil histogram Hue-index (fitur warna) dan algoritma CCD (fitur bentuk). Hasil penggabungan adalah:

$$\nu = [\nu^{\text{warna}} \quad \nu^{\text{bentuk}}]$$

Hasil penggabungan merupakan sebuah vektor baru. Bila masing-masing fitur mempunya 40 indeks, maka vektor fitur yang dibentuk ada 80 indeks.

Nilai dari setiap vektor tidak seragam, sehingga diperlukan proses normalisasi agar bobot setiap nilainya bisa ditentukan. Normalisasi histogram dilakukan dengan membagi tiap nilai histogram dengan nilai terbesar dari data set histogram tersebut.

$$\nu_i = \frac{\nu_i}{\max(\nu_i)}$$

9.3.7. Perhitungan Jarak Kemiripan Gambar

Proses ini menghitung jarak kemiripan gambar input dengan gambar-gambar yang pada pada database. Perhitungan jarak ini dilakukan antara vektor fitur dari gambar input dan vektor fitur dari gambar-gambar yang ada dalam database. Terdapat dua macam perhitungan jarak yaitu Euclidian dan Entropi.

(1) Jarak Euclidian

Perhitungan jarak pada aplikasi ini menggunakan metode Euclidian Distance. Metode ini diimplementasikan dengan rumus sebagai berikut.

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Dimana,

- d = jarak dua gambar dengan perhitungan Euclidian Distance
 x_i = data ke-i pada histogram gambar input
 y_i = data ke-i pada histogram gambar yang ada di database
 n = panjang data pada histogram

(2) Entropi

Perhitungan entropi dilakukan dengan mengambil data peringkat 1 sampai peringkat tertentu dari data euclidian distance yang sudah diurutkan. Misalkan diambil gambar peringkat 1 sampai 6. Gambar 6.9 adalah contoh hasil pencarian gambar input bunga yang menampilkan gambar dengan nilai euclidean distance antara peringkat 1 sampai 6.

Dari semua gambar yang dianggap memiliki potensi sebagai output ini, dihitung nilai entropinya dengan rumus berikut

$$\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Dimana (p_+) merupakan jumlah rata-rata nilai euclidean distance kelompok gambar yang bersolusi positif atau mendukung pada data input. Sedangkan (p_-) adalah jumlah rata-rata nilai euclidean distance kelompok gambar yang bersolusi negatif atau tidak mendukung pada data input.

Yang pertama dilakukan adalah mengambil data peringkat teratas sesuai yang ditentukan. Dari kesekian data yang diambil tersebut, dilakukan perhitungan jumlah nilai euclidian distance untuk data dengan id_jenis yang sama.

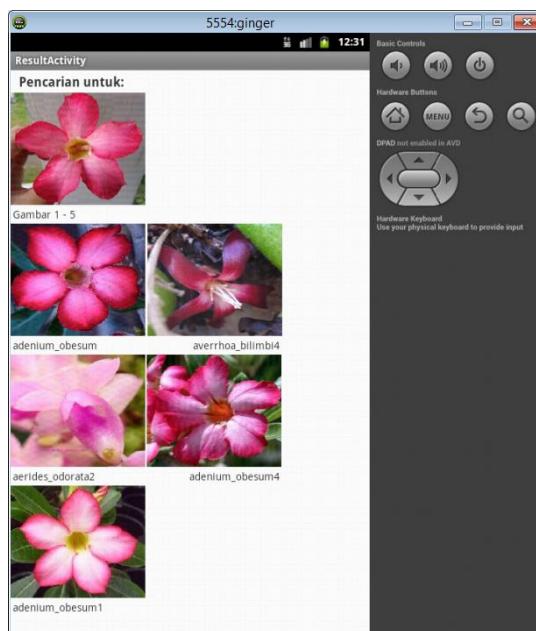
Selanjutnya bisa dilakukan perhitungan variable yang diperlukan untuk mendapatkan nilai entropi. Nilai (p_+) didapatkan dengan membagi jumlah nilai euclidian dengan jumlah gambar dengan id_jenis yang sama. Sedangkan untuk nilai (p_-) didapatkan

dengan membagi jumlah nilai euclidian distance dengan jumlah gambar yang memiliki id_jenis berbeda dengan id_jenis gambar pada perhitungan (p_+).

Nilai entropi dihitung menggunakan nilai (p_+) dan nilai (p_-) yang telah didapat dengan rumus yang sudah disebutkan. Perhitungan entropi ini dilakukan untuk masing-masing id_jenis. Perhitungan entropi disimpan dalam array dua dimensi dan diurutkan secara descending. Yang berarti entropi terbesar berada di peringkat atas. Output yang terpilih adalah jenis bunga dengan entropi terbesar dan kurang dari threshold. Bila nilai tersebut melebihi threshold maka dianggap gambar input tidak menemukan hasil pencarian atau bukan termasuk flora Indonesia.

9.3.8. Pengujian Pengukuran Kemiripan Gambar

Pengujian ini dilakukan untuk menentukan cara pengukuran kemiripan antara gambar input dan gambar output. Perhitungan kemiripan gambar dilakukan berdasarkan fitur yang dimiliki dari setiap gambar. Karena fitur warna dan fitur bentuk memiliki model nilai yang sama yaitu antara 0 hingga 1, hanya panjang datanya saja yang berbeda, maka pengujian ini dilakukan pada satu fitur saja. Fitur yang dipilih adalah fitur warna. Gambar 9.25 merupakan hasil pencarian 5 gambar dengan nilai euclidean distance terkecil berdasarkan gambar input. Sedangkan Tabel 6.4 menunjukkan hasil pengujian pengukuran kemiripan gambar yang telah dilakukan.



Gambar 9.25 Hasil pencarian 5 gambar peringkat atas

Tabel 9.3. Hasil pengujian pengukuran kemiripan gambar

Metode Uji Coba	Prosentase Kemiripan Hasil
Euclidean Distance	50%
Euclidean Distance + Entopi	93.18%

Jika melihat hasil pengujian dari fitur bentuk bahwa gambar input masih berada di antara peringkat 1 sampai 9. Hal ini dapat digunakan untuk memperbaiki kekurangan pada identifikasi berdasarkan fitur warna. Jadi, terlebih dahulu dilakukan identifikasi menggunakan fitur bentuk. Hasil perhitungan euclidian distance nya diambil dari peringkat 1 hingga 9. Hanya id_jenis bunga yang terdapat pada kesembilan data inilah dilakukan identifikasi berdasarkan fitur warna. Perhitungan entropi dari identifikasi yang terakhir diurutkan lagi. Entropi terbesar dan kurang dari treshold diambil sebagai output.

Dari hasil semua pengujian terlihat bahwa data output yang benar adalah 43 dari 44 percobaan. Jika dibandingkan dengan pengujian berdasarkan fitur warna saja, pengujian dengan kombinasi fitur warna dan fitur bentuk menghasilkan tingkat kemiripan yang lebih tinggi. Pada bunga id 4,6, dan 35 sebelumnya menghasilkan output yang salah pada identifikasi berdasarkan fitur warna saja. Namun setelah dikombinasikan dengan fitur bentuk, bunga-bunga dengan id tersebut menghasilkan output yang sesuai. Namun, hal lain yang terjadi adalah bunga dengan id 21 yang tidak teridentifikasi, padahal pada identifikasi berdasarkan warna bunga id 21 menghasilkan output yang sesuai. Hal ini disebabkan karena bunga id 21 tidak masuk peringkat 1 sampai 9 pada proses identifikasi dengan fitur bentuk, sehingga bunga id 21 tidak masuk dalam proses identifikasi berdasarkan fitur warna. Meski demikian, identifikasi bunga dengan menggunakan fitur warna dan fitur bentuk telah meningkatkan nilai kemiripan dari 93.18% menjadi 97.72%.

DAFTAR PUSTAKA

1. Rafael C. Gonzales, Richard E.Woods,"*Digital Image Processing 3rd edition* ", Prentice Hall,2010
2. Wanasanan Thongsongkrit, "*Lecture Notes Digital Image Processing Chapter 1,2,9*", Department of Computer Engineering Faculty of Engineering Chiang Mai University
3. Richard Alan Peter, "*Lecture Notes 2007*", [http://www.archive.org
/details/Lectures_on_Image_Processing](http://www.archive.org/details/Lectures_on_Image_Processing)
4. Achmad Basuki, Nana Ramadijanti, Fadilah Fahrul, "*Modul Praktikum Pengolahan Citra Dengan C#2012*", PENS-2013