

Catatan Kuliah Pengantar Pengolahan Citra

Adang Suhendra

Date text

Daftar Isi

1	Pendahuluan	1
1.1	Pengertian Citra	1
1.2	Perspective Projection	3
1.3	Sistem Visual Manusia	4
1.4	Pemanfaatan Pengolahan Citra	8
1.5	Implementasi Pengolahan Citra	9
1.6	Proses Pengolahan Citra	9
1.7	Representasi Citra	10
1.8	QUIZ	11
1.9	LATIHAN PROGRAM	11
2	Warna	13
2.1	Warna pada Pengolahan Citra	13
2.2	Teori Tristimulus Persepsi Warna	14
2.3	Model Warna	17
2.3.1	Model RGB	18
2.3.2	Model CMY	19
2.3.3	Model HSI	19
2.4	Model YIQ	20
2.5	Applying Greyscale Transformations to Colour Images	21
3	Perspektif dan Transformasi Citra	25
3.1	Koordinat Kamera dan Koordinat World	25
3.2	Transformasi Geometrik	26
3.3	Sistem Koordinat Cartesian dan Homogeneous	28
3.4	Transformasi Image ke World	28
3.5	Distorsi Geometrik	29
3.6	Distorsi Radiometrik	31

DAFTAR ISI

3.7	Impulse	31
4	Transformasi Fourier	35
4.1	Transformasi Citra	35
4.2	Transformasi Fourier	36
4.2.1	Fourier 1 Dimensi	37
4.2.2	Rumus FT 2 Dimensi	39
5	Peningkatan Mutu Citra	41
5.1	Domain Spatial	42
5.1.1	Point Processing	43
5.1.2	Contrast Stretching dan Thresholding	43
5.1.3	Tiga Fungsi Transformasi Dasar	45
5.1.4	Histogram	48
5.1.5	Penggeseran dan Pelebaran Histogram	49
5.1.6	Perataan Histogram (Histogram Equalization)	49
5.1.7	Mean Filter	54
5.1.8	Order-Statistics Filters (Nonlinear Filters)	56
5.2	Filter Domain Frekuensi	59
6	Segmentasi Citra	61
6.1	Deteksi Diskontinuitas	61
6.1.1	Deteksi Titik	61
6.1.2	Deteksi Garis	62
6.1.3	Deteksi Sisi (<i>Edge Detection</i>)	63
6.2	Segmentation	67
6.2.1	Segmentasi Regular Block	68
6.2.2	Segmentasi Binary tree	69
6.2.3	Polygon segmentation	69

Daftar Gambar

1.1	Intensitas	3
1.2	Proyeksi	3
1.3	Weak Perspective	4
1.4	Sistem Visual Manusia	5
1.5	Ranah adaptasi warna pada sistem visual manusia	6
1.6	Weber Ratio	6
1.7	Garis seperti lipatan pada daerah batas tingkat intensitas . . .	7
1.8	Mach Band Efek	8
1.9	Tampilan kotak di tangan berkesan lebih gelap apabila back-ground lebih terang	8
1.10	Proses Pengolahan Citra dan Implementasinya	10
1.11	pallette	11
2.1	Visible Spectrum	14
2.2	Spectral response curves for each cone type. The peaks for each curve are at 440nm (blue), 545nm (green) and 580nm (red).	15
2.3	Diagram CIE Chromaticity menunjukkan semua visible colours. x dan y adalah jumlah normalisasi kemunculan X and Y primaries, $z = 1 - x - y$ menentukan jumlah Z primary yang dibutuhkan.	16
2.4	Warna Campuran pada Diagram Chromaticity	17
2.5	Koordinat warna RGB	18
2.6	Penambahan Campuran Warna Merah Hijau dan Biru	19
2.7	Model warna HSI	20

DAFTAR GAMBAR

2.8	Image (a) shows a colour test pattern, consisting of horizontal stripes of black, blue, green, cyan, red, magenta and yellow, a colour ramp with constant intensity, maximal saturation, and hue changing linearly from red through green to blue, and a greyscale ramp from black to white. Image (b) shows the intensity for image (a). Note how much detail is lost. Image (c) shows the luminance. This third image accurately reflects the brightness variations perceived in the original image. . . .	22
2.9	The top image is a very dark image of a forest scene. The middle image is the result of applying histogram equalisation to each of the red, green and blue components of the original image. The bottom image is the result of converting the image to YIQ format, and applying histogram equalisation to the luminance component only.	23
3.1	Visible Spectrum	25
3.2	Visible Spectrum	26
3.3	Visible Spectrum	30
3.4	Visible Spectrum	31
3.5	Distorsi Radiometrik dan Geometrik	32
3.6	Distorsi Radiometrik <i>blurring</i>	33
3.7	Fungsi Delta	33
3.8	Fungsi Konvolusi Impulse	34
3.9	Point Spread Function	34
4.1	Fungsi Kotak	37
4.2	Gambar Fungsi Fourier a) $n = 1$, b) $n = 3$, c) $n = 7$, d) $n = 99$	38
4.3	Fungsi Cos Sin	39
4.4	Fourier Transform	39
5.1	Proses Peningkatan Mutu Citra	41
5.2	Teknik Spatial merubah langsung pixel (x,y)	43
5.3	Masking pixel (x,y)	43
5.4	Contrast Stretching	44
5.5	Threshold	44
5.6	Fungsi Identitas dan Negatif	45
5.7	Transformasi Fungsi Negatif	46
5.8	Transformasi Fungsi Logaritmik	46

DAFTAR GAMBAR

5.9	Plot fungsi $c \cdot r^\gamma$	47
5.10	Penampilan citra di monitor	48
5.11	Histogram	50
5.12	Penggeseran Histogram	51
5.13	Pelebaran Histogram	52
5.14	Bentuk fungsi $T(r_k)$	53
5.15	Histogram Citra Dijital M	54
5.16	Histogram Setelah Proses Perataan dari Citra digital M	54
5.17	Mean Filter	55
5.18	(a) Citra yang diberi noise, (b) Mean Filter 3x3 (c)Median Filter 3x3	57
5.19	Transformasi Laplacian untuk penajaman detail pada gambar permukaan bulan bagian kutub utara	58
5.20	Langkah Proses Filter Domain Frekuensi	60
6.1	Deteksi Titik pada Citra	62
6.2	Deteksi Sisi Horizontal Vertikal	65
6.3	Deteksi Sisi secara Diagonal	66
6.4	Laplacian of Gaussin (LoG)	66
6.5	Proses Citra Zero Crossing dari Laplacian of Gaussin (LoG) .	67
6.6	Contoh Struktur Data Quadtree	70
6.7	Contoh Implementasi Segmentasi Quadtree	71
6.8	Contoh Pembagian Blok Polygon	71

Bab 1

Pendahuluan

Dengan digunakannya sistem satelit dalam teknik penginderaan jarak jauh maka dapat dikumpulkan data permukaan bumi dalam jumlah yang sangat besar, karena luas permukaan bumi yang sangat luas. Data citra yang sudah diperoleh tersebut kemudian dianalisis untuk kepentingan tertentu, misalkan diagnosa penyakit, pemetaan dsb. Teknik analisis yang konvensional, yaitu mengamati bentuk citra analog, membutuhkan waktu yang lama dan ketepatan yang kurang signifikan. Tetapi setelah ditemukannya citra digital, maka sudah dapat menggunakan alat bantu komputer yang dapat menganalisa citra tersebut dengan waktu yang lebih cepat dan ketepatan yang signifikan.

Sejarah perkembangan sistem pengolahan citra secara digital tidak hanya didukung oleh kebutuhan dalam aplikasi penginderaan jauh, tetapi juga dalam beberapa aplikasi lainnya seperti biomedik, astronomi, dan arkeologi yang umumnya membutuhkan suatu metode peningkatan kualitas citra.

1.1 Pengertian Citra

Suatu citra adalah fungsi intensitas 2 dimensi $f(x, y)$, dimana x dan y adalah koordinat spasial dan f pada titik (x, y) merupakan tingkat kecerahan (*brightness*) suatu citra pada suatu titik. Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek. Gambar 1.1 adalah gambar penangkapan / penerimaan citra oleh mata manusia.

Citra sebagai output alat perekaman, seperti kamera, dapat bersifat analog ataupun digital.

BAB 1. PENDAHULUAN

Citra Analog adalah citra yang masih dalam bentuk sinyal analog, seperti hasil pengambilan gambar oleh kamera atau citra tampilan di layar TV ataupun monitor (sinyal video).

Menurut presisi yang digunakan untuk menyatakan titik-titik koordinat pada domain spasial (bidang) dan untuk menyatakan nilai keabuan (warna suatu citra), maka secara teoritis citra dapat dikelompokkan menjadi empat kelas citra, yaitu: kontinu-kontinu, kontinu-diskrit, diskrit-kontinu, diskrit-diskrit. Parameter(label) pertama menyatakan presisi titik koordinat pada bidang, sedangkan label kedua menyatakan presisi nilai keabuan/warna. Label kontinu berarti nilai yang digunakan adalah tak terbatas dan tak terhingga, sedangkan diskrit menyatakan terbatas dan berhingga. Suatu citra digital merupakan representasi 2-D array sample diskrit suatu citra kontinu $f(x,y)$. Amplitudo setiap sample di kuantisasi untuk menyatakan bilangan hingga bit. Setiap elemen array 2-D sample disebut suatu pixel atau pel (dari istilah "picture element") Pengolahan citra digital adalah proses pengolahan citra digital dengan alat bantu komputer.

Tingkat ketajaman/resolusi warna pada citra digital tergantung pada jumlah "bit" yang digunakan oleh komputer untuk merepresentasikan setiap pixel tersebut. Tipe yang sering digunakan untuk merepresentasikan citra adalah "8-bit citra" (256 colors (0 untuk hitam - 255 untuk putih)), tetapi dengan kemajuan teknologi perangkat keras grafik, kemampuan tampilan citra di komputer hingga 32 bit (232 warna).

Ranah nilai intensitas dalam suatu citra juga ditentukan oleh alat digitasi yang digunakan untuk menangkap dan konversi citra analog ke citra digital (A/D).

Perolehan citra digital ini dapat dilakukan secara langsung oleh kamera digital ataupun melakukan proses konversi suatu citra analog ke citra digital. Untuk mengubah citra kontinu menjadi digital diperlukan proses pembuatan kisi-kisi arah horizontal dan vertikal, sehingga diperoleh gambar dalam bentuk array dua dimensi. Proses tersebut dikenal sebagai proses digitalisasi/sampling.

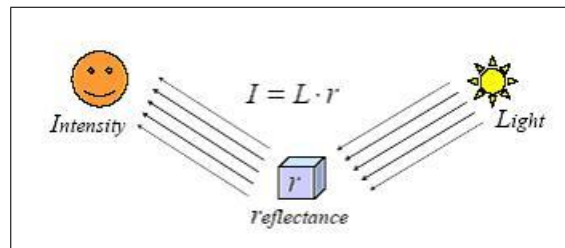
Citra monochrome atau citra hitam-putih merupakan citra satu kanal, di mana citra $f(x,y)$ merupakan fungsi tingkat keabuan dari hitam ke putih; x menyatakan variabel baris (garis jelajah) dan y menyatakan variabel kolom atau posisi di garis jelajah. Sebaliknya citra berwarna dikenal juga dengan citra multi-spectral, di mana warna citra biasanya

1.2. PERSPECTIVE PROJECTION

1.2 Perspective Projection

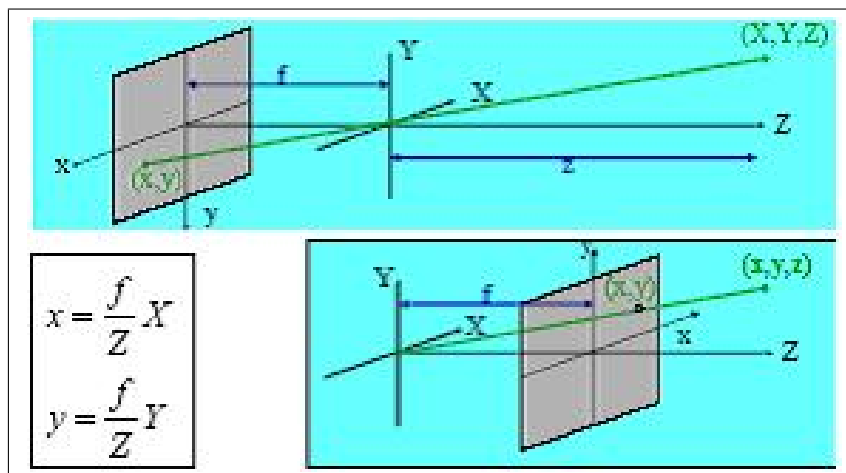
Perspective projection adalah proses transformasi dari koordinat 3Dworld ke 2D image, yang dilakukan berdasarkan:

- Continuous Perspective Projection
- optics



Gambar 1.1: Intensitas

Pada Gambar 1.2 proyeksi suatu koordinat 3 dimensi(X, Y, Z) ke koordinat 2 Dimensi (x, y) pada model perspective kontinu.



Gambar 1.2: Proyeksi

Weak perspective (scaled orthographic projection)

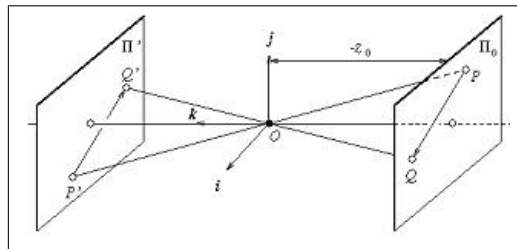
Merupakan efek perspektif, tetapi tidak berdasarkan skala objek secara individu, yaitu mengumpulkan titik-titik ke suatu kelompok pada sekitar

kedalaman yang sama dan kemudian membagi setiap titik dengan kedalaman kelompok titik-titik itu. Gambar 1.3 menunjukkan bentuk proyeksi weak perspective.

Bentuk persamaan untuk weak perspective adalah:

$$(x, y, z) \longrightarrow s(x, y)$$

Dimana s bernilai konstan untuk semua titik, garis parallel tidak konvergen tetapi terus parallel.



Gambar 1.3: Weak Perspective

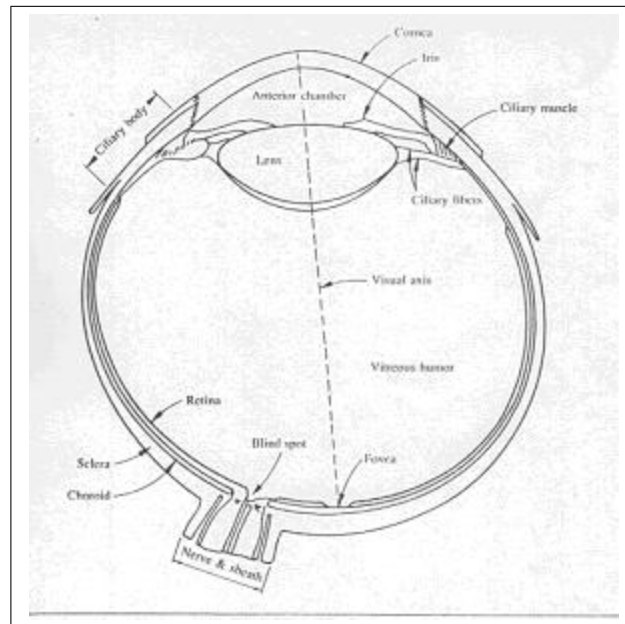
1.3 Sistem Visual Manusia

Kita perlu memperhatikan konsep system visual manusia dalam pengolahan citra, sebab proses tersebut sangat identik dengan proses pengolahan citra. Ketika manusia menangkap citra di mata, maka prinsip tersebut juga diterapkan oleh pengolahan citra untuk mengambil citra dari suatu alat sensor citra. Dari informasi citra yang ditangkap oleh mata, kemudian diproses untuk mengidentifikasi citra tersebut.

Beberapa pertanyaan sederhana yang muncul berkaitan dengan system vision manusia:

- Intensitas cahaya apa yang dapat kita bedakan?
- Apa resolusi spasial mata manusia ?
- Berapa akurat estimasi kita ketika membandingkan jarak dan luas area?
- Bagaimana sense manusia terhadap warna?

1.3. SISTEM VISUAL MANUSIA



Gambar 1.4: Sistem Visual Manusia

Pembentukan Citra oleh Sensor Mata

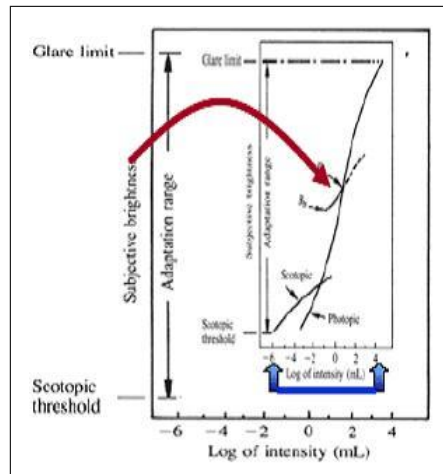
Gambar 1.4 menunjukkan system penangkapan citra pada mata manusia. Intensitas cahaya ditangkap oleh diagram iris dan diteruskan ke bagian retina mata. Kemudian bayangan obyek pada retina mata dibentuk dengan mengikuti konsep sistem optik dimana fokus lensa terletak antara retina dan lensa mata. Mata dan syaraf otak dapat menginterpretasi bayangan yang merupakan obyek pada posisi terbalik.

Pada system mata manusia terdapat bagian: Fovea di bagian retina terdiri dari dua jenis receptor: Sejumlah (6-7 juta) cone receptor, sensitif terhadap warna, vision cone disebut photopic vision atau bright light vision. Sejumlah rod receptor (75-150 juta) terletak pada permukaan retina, memberikan gambar keseluruhan pandangan dan sensitif terhadap iluminasi tingkat rendah, vision rod disebut scotopic vision atau dim-light vision.

Blind Spot adalah bagian retina yang tidak mengandung receptor sehingga tidak dapat menerima dan menginterpretasi informasi. Adaptasi terhadap kecerahan dan diskriminasi

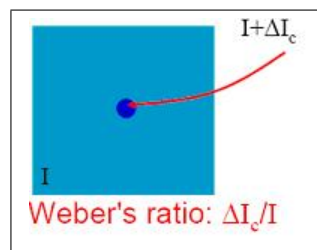
Ranah total tingkat intensitas yang dapat mendiskriminasikan secara simultan adalah lebih kecil dibandingkan ranah total adaptasi. Pada gam-

BAB 1. PENDAHULUAN



Gambar 1.5: Ranah adaptasi warna pada sistem visual manusia

bar 1.5, B_a adalah tingkat adaptasi kecerahan, Kurva Irisan yang pendek menyatakan ranah subjektif kecerahan yang dapat diterima oleh mata ketika mengadaptasikan tingkat kecerahan tersebut.

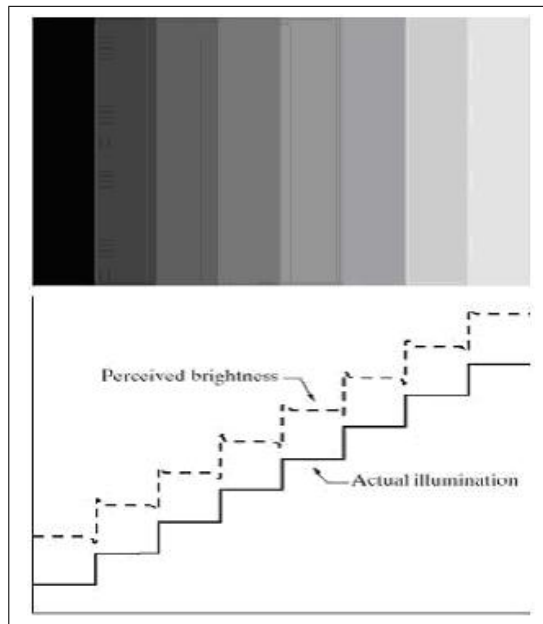


Gambar 1.6: Weber Ratio

Kemampuan mata untuk diskriminasi hitam/putih kecerahan pada level adaptasi tertentu. Pada gambar 1.6, I adalah nilai iluminasi uniform pada area datar dengan luasan yang cukup untuk keseluruhan field of view. ΔI_c adalah perubahan kecerahan objek yang diperlukan untuk membedakan objek dari warna latar belakangnya. Diskriminasi yang buruk apabila Weber Ratio ($\Delta I_c / I$) bernilai besar untuk tingkat iluminasi rendah dan akan meningkat signifikan (ketika ratio rendah) yaitu iluminasi background meningkat. Sangat sulit untuk membedakan iluminasi apabila background terang sebaliknya akan sangat mudah pada background yang lebih gelap.

Brightness (kecerahan) bukan merupakan suatu model fungsi intensitas

1.3. SISTEM VISUAL MANUSIA



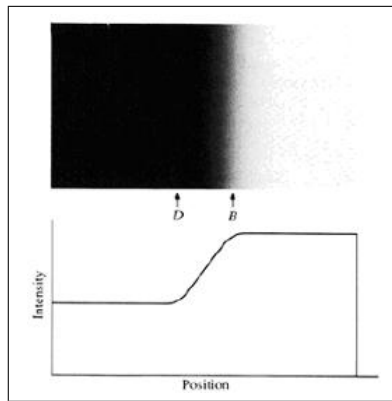
Gambar 1.7: Garis seperti lipatan pada daerah batas tingkat intensitas

yang sederhana, karena system visual sangat dipengaruhi oleh daerah batas intensitas yang berbeda. Intensitas dari masing-masing daerah pada gambar 1.7 adalah konstan tetapi pada kenyataan penglihatannya pattern brightness sangat kuat sehingga membentuk seperti garis lipatan di dekat batas masing-masing daerah tersebut.

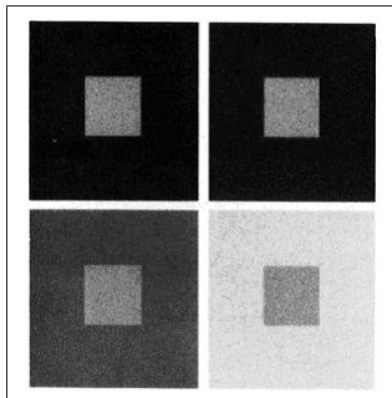
Pada tahun 1865, Ernst Mach menemukan fenomena mach band pattern . Pada gambar 1.8 di atas, brightness pattern yang diterima berupa suatu stripe yang lebih gelap di daerah D dan lebih terang di daerah B padahal kenyataanya daerah D dan B memiliki intensitas yang sama.

Contrast Simultan

Gambar 1.9. menyatakan suatu model contrast simultan, dimana kemunculan kotak kecil ditengah berkesan memiliki intensitas yang semakin lebih gelap apabila warna background menjadi lebih terang, padahal kenyataannya intensitas kotak kecil tersebut sama. Hal ini menyatakan bahwa daerah brightness yang diterima tidak bergantung sederhana pada intensitasnya.



Gambar 1.8: Mach Band Efek



Gambar 1.9: Tampilan kotak di tengah berkesan lebih gelap apabila background lebih terang

1.4 Pemanfaatan Pengolahan Citra

Manfaat pengolahan citra adalah menunjang kebutuhan kehidupan sehari-hari khususnya untuk :

- Memfasilitasi penyimpanan dan transmisi citra seperti menentukan metode penyimpanan citra yang efisien dalam suatu kamera digital sehingga mempercepat proses pengiriman citra dari jarak jauh misalkan dari planet Mars ke Bumi.
- Menyiapkan untuk ditampilkan di monitor atau di cetak. Proses yang dilakukan adalah melakukan merubah ukuran citra yang dharus di-

1.5. IMPLEMENTASI PENGOLAHAN CITRA

esuaikan dengan ukuran media tampilan serta proses halftoning untuk proses pencetakan.

- Meningkatkan dan memperbaiki citra dengan menghilangkan goresan-goresan pada ataupun meningkatkan visibilitas citra
- Ekstraksi informasi citra misalkan character recognizing, pengukuran plusi air dari citra aerial

1.5 Implementasi Pengolahan Citra

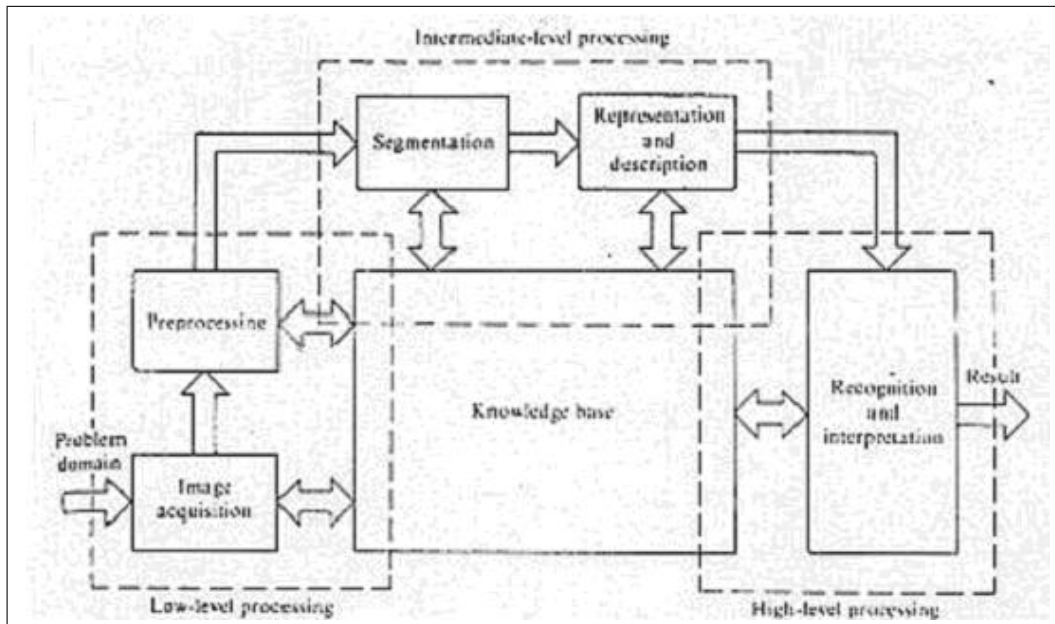
Aplikasi pengolahan citra digital sudah diimplementasikan secara luas, khususnya di bidang keilmuan maupun industri seperti kedokteran, keamanan, pertanian, geologi, biology, system kontrol fabrikasi, dll. Bentuk dari model implementasi citra:

- Pengembangan Sistem Aplikasi Biomedik
- Pengembangan Sistem Optical Character Recognition (OCR)
- Pengembangan Sistem Aplikasi Inderaja
- Pengembangan Sistem Multitemporal Multisensor Image Classification and Fusion

1.6 Proses Pengolahan Citra

Gambar 1.10 di atas adalah langkah dalam pengolahan citra secara umum yang terdiri dari:

- Pembentukan Citra (Data Acquisition): Menentukan data yang diperlukan dan memilih metode perekaman citra digital.
- Pengolahan Citra Tingkat Awal (Image Preprocessing): Meningkatkan kontras, menghilangkan gangguan geometrik / radiometrik, menentukan bagian citra yang akan diobservasi. Segmentasi Citra (Image Segmentation) dan Deteksi Sisi (Edge Detection): Melakukan partisi citra menjadi wilayah-wilayah obyek (internal properties) atau menentukan garis batas wilayah obyek (external shape characteristics).



Gambar 1.10: Proses Pengolahan Citra dan Implementasinya

- Seleksi dan Ekstraksi Ciri (Feature Extraction and Selection): Seleksi ciri memilih informasi kuantitatif dari ciri yang ada, yang dapat membedakan kelas-kelas obyek secara baik. Ekstraksi ciri mengukur besaran kuantitatif ciri setiap piksel

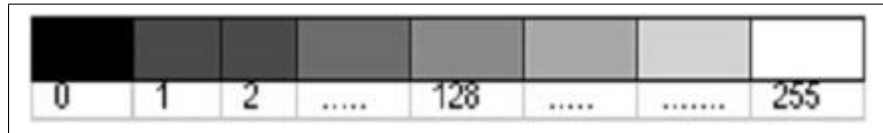
1.7 Representasi Citra

Bagaimana sebuah citra direpresentasikan dalam file? Pertama-tama seperti halnya jika kita ingin melukis sebuah gambar, kita harus memiliki palet dan kanvas Palette:

- kumpulan warna yang dapat membentuk citra, sama halnya seperti ketika hendak melukis dengan cat warna, kita memiliki palet yang bisa kita isikan berbagai warna cat air
- Setiap warna yang berbeda dalam palet tersebut diberi nomor (berupa angka)

1.8. QUIZ

- Contoh untuk citra monokrom (warnanya hanya putih-abuabu-hitam), berarti kita memiliki palet seperti pada gambar 1.11



Gambar 1.11: pallete

1.8 QUIZ

1. Projection : Titik fokus suatu kamera ditetapkan pada koordinat (10, 0, 1). Suatu plane citra ditempatkan pada koordinat $z = 2$. Terdapat dua titik pada koordinate world: $p1 = (7, 3, 7)$, $p2 = (8, 3, 13)$.
 - Tentukan koordinat world titik dari proyeksi $p1$ dan $p2$ dengan menggunakan perspective projection.
 - Tentukan pula nilai proyeksi $p1$ and $p2$ apabila menggunakan weak perspective.
2. Sebutkan efek kecerahan (brightness) iluminasi pada system visual manusia.
3. Jelaskan efek visual manusia apa yang terjadi apabila manusia yang sebelumnya berada ditempat terang kemudian tiba-tiba masuk ke ruangan yang gelap.

1.9 LATIHAN PROGRAM

- Buatlah suatu matlab script untuk membaca suatu input file citra serta tampilkan informasi format citra tersebut.

BAB 1. PENDAHULUAN

Bab 2

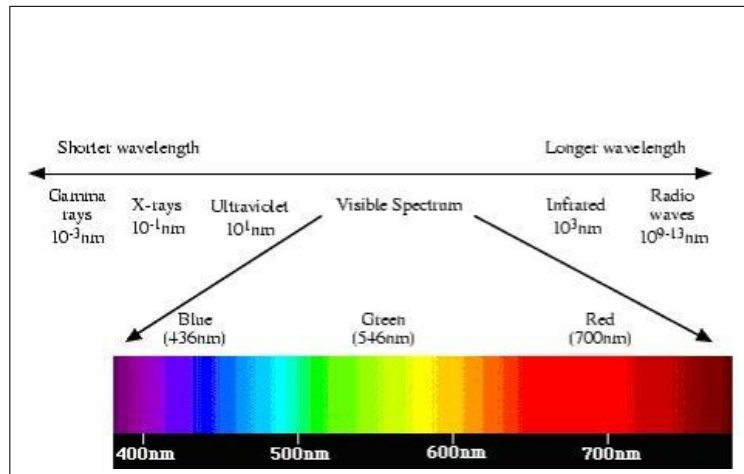
Warna

2.1 Warna pada Pengolahan Citra

Sistem visual manusia dapat membedakan ratusan ribu shade warna dan intensitas, tetapi hanya 100 shade keabun. Oleh sebab itu, dalam suatu citra, masih banyak informasi lainnya yang ada pada warna, dan informais tersebut juga dapat digunakan untuk menyederhanakan analisis citra, misalkan identifikasi objek dan ekstraksi warna.

Tiga kuatisasi yang dapat digunakan untuk menggambarkan warna:

- *hue* ditentukan oleh dominan panjang gelombang. Warna yang dapat dilihat oleh mata memiliki panjang gelombang antara 400 nm (violet) - 700 nm (red) pada spektrum electromagnetic seperti pada gambar 2.1
- (*Saturation*) ditentukan oleh tingkat kemurnian, dan tergantung pada jumlah sinar putih yang tercampur dengan hue. Suatu warna hue murni adalah secara penuh tersaturasi, yaitu tidak ada sinar putih yang tercampur. Hue dan saturation digabungkan menentukan chromaticity suatu warna. Intensitas ditentukan oleh jumlah sinar yang diserap. Semakin banyak sinar yang diserap semakin banyak tinggi intensitas warnanya.
- Sinar *Achromatic* tidak memiliki warna, tetapi hanya ditentukan oleh atribut intensitas. Tingkat keabuan (Greylevel) adalah ukuran intensitas yang ditentukan oleh energi, sehingga merupakan suatu kuantitas fisik. Dalam hal lain, brightness atau luminance ditentukan oleh persepsi warna (sehingga dapat merupakan efek *psychology*). Apabila



Gambar 2.1: Visible Spectrum

diberikan sinar biru dan hijau dengan intensitas yang sama, sinar biru diterima (perceived) lebih gelap dibandingkan sinar hijau. Sehingga dapat dikatakan bahwa persepsi intensitas manusia adalah non-linear, misalkan perubahan intensitas yang dinormalisasi dari 0.1 ke 0.11 dan 0.5 ke 0.55 akan diterima dengan perubahan tingkat kecerahan (brightness) yang sama.

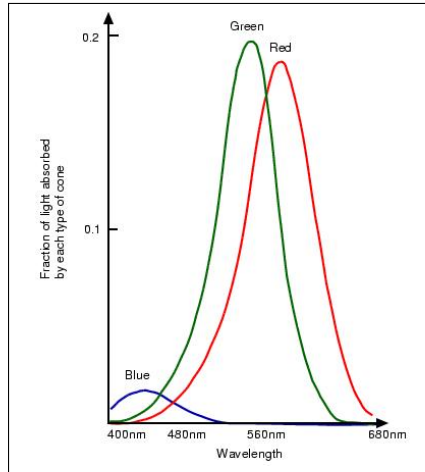
Warna secara utuh bergantung pada sifat pantulan (*reflectance*) suatu objek. Warna yang dilihat merupakan yang dipantulkan, sedangkan yang lainnya diserap. Sehingga sumber sinar perlu diperhitungkan begitu pula sifat alami system visual manusia ketika menangkap suatu warna. Sebagai contoh, suatu objek yang memantulkan sinar merah dan hijau akan tampak berwarna hijau apabila benda tersebut disinari oleh sinar hijau (tanpa adanya sinar merah). Demikian juga sebaliknya, objek akan tampak berwarna merah apabila tidak terdapat sinar hijau. Apabila benda tersebut disinari oleh sinar putih, maka objek tersebut berwarna kuning (merupakan gabungan warna hijau + merah).

2.2 Teori Tristimulus Persepsi Warna

Seperti yang dijelaskan pada Bab 1, retina manusia memiliki 3 jenis cones. Respon setiap jenis cone sebagai suatu fungsi panjang gelombang (gambar

2.2. TEORI TRISTIMULUS PERSEPSI WARNA

2.2). Pada Gambar 2.2, puncak setiap kurva warna adalah [ada 440 nm (biru), 545 nm (hijau) dan 580 nm (merah).



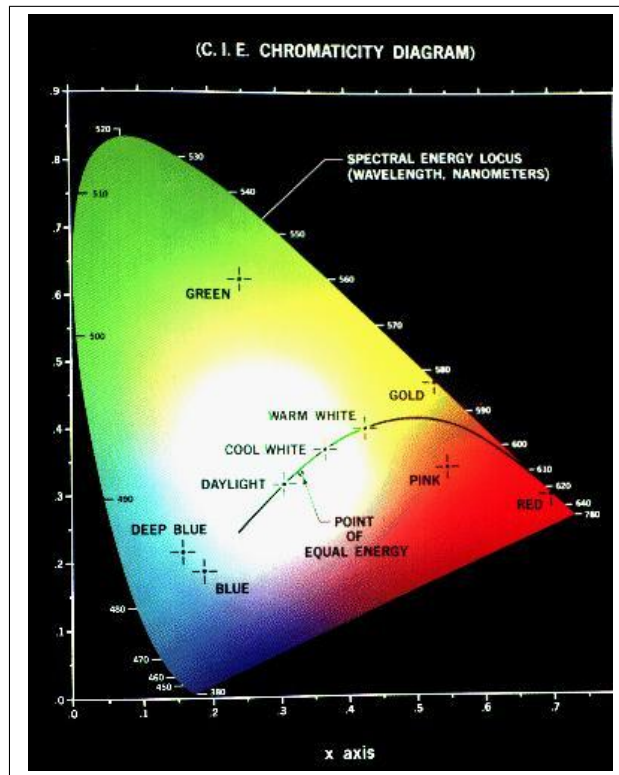
Gambar 2.2: Spectral response curves for each cone type. The peaks for each curve are at 440nm (blue), 545nm (green) and 580nm (red).

CIE primaries

Teori tritstimulus persepsi warna dapat dilihat bahwa suatu warna dapat diperoleh dari suatu campuran tiga warna utama: merah, hijau dan biru (Red Green Blue). Meskipun hampir setiap warna yang tampak dapat ditentukan sesuai dengan tiga komponen diatas, tetapi masih terdapat beberapa warna yang tidak dapat diuraikan sebagai kombinasi dari ketiga warna dasar tersebut. Bagaimanapun juga apabila salah satu dari ketiga komponen warna dasar tersebut ditambahkan ke warna yang tidak dapat dicocokkan tadi, maka warna yang tidak dapat dicocokkan tersebut dapat dicocokkan dengan campuran dari dua warna dasar lain. Hal ini menunjukkan bahwa warna dapat memiliki nilai bobot negatif dari ketiga komponen warna dasar tersebut. Pada tahun 1931, Commission Internationale de l'Éclairage (CIE) mendefinisikan tiga standar komponen warna utama : X, Y dan Z. yang dapat ditambahkan untuk membentuk semua kemungkinan warna. Warna utama Y dipilih sedemikian rupa sehingga fungsi kecocokan warnanya secara tepat mencocokkan fungsi luminous efisiensi mata manusia berdasarkan penjumlahan ketiga warna seperti pada gamabar 2.3.

Diagram Chromaticity (Gambar 2.3) menunjukkan semua visible colours. Sumbu x dan y merupakan nilai normalisasi warna utama X dan Y untuk

BAB 2. WARNA



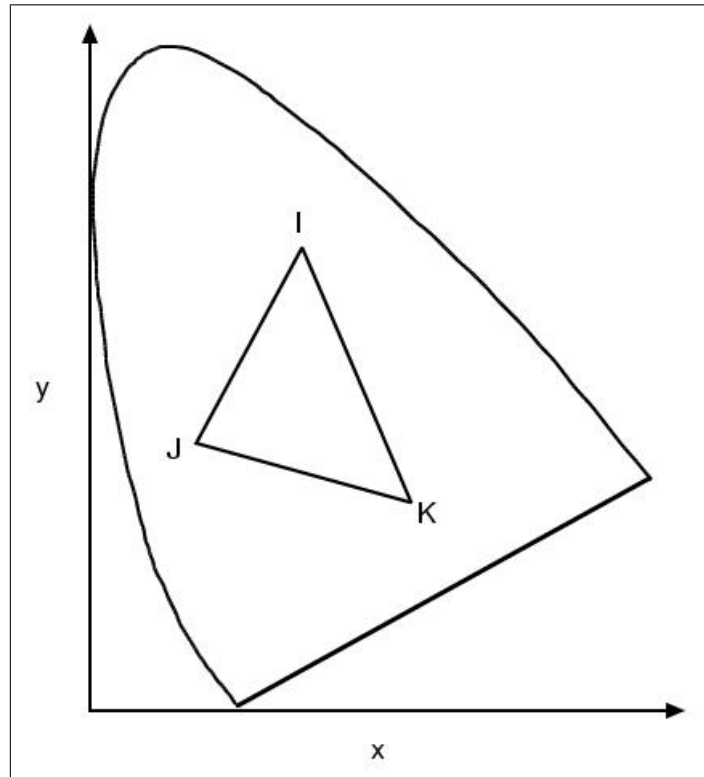
Gambar 2.3: Diagram CIE Chromaticity menunjukkan semua visible colours. x dan y adalah jumlah normalisasi kemunculan X and Y primaries, $z = 1 - x - y$ menentukan jumlah Z primary yang dibutuhkan.

suatu warna, dan $z = 1 - x - y$ menyatakan jumlah Z utama yang diperlukan. Chromaticity bergantung pada panjang gelombang dan saturation dominan, dan tidak bergantung pada energi luminan. Warna dengan nilai chromaticity yang sama tetapi dengan luminan berbeda akan terpetakan pada titik yang sama di region tersebut.

Warna spectrum utama murni berada pada bagian kurva batas daerah, dan suatu sinar putih standar memiliki warna yang didefinisikan berada dekat (tetapi tidak di) titik dengan persamaan energi $x = y = z = 1/3$. Warna komplementer, yaitu warna yang ditambahkan ke warna putih, berada di titik akhir suatu garis yang melewati titik tersebut. Sebagai ilustrasi pada gambar 2.4, semua warna yang berada di dalam segitiga dapat dibentuk dari campuran warna yang berada pada verteks (garis) segitiga. Dari ilustrasi

2.3. MODEL WARNA

grafik tersebut, semua warna visible tidak dapat diperoleh dari campuran warna utama R, G dan B (atau dari tiga visible warna lainnya), karena bentuk diagramnya bukan segitiga.



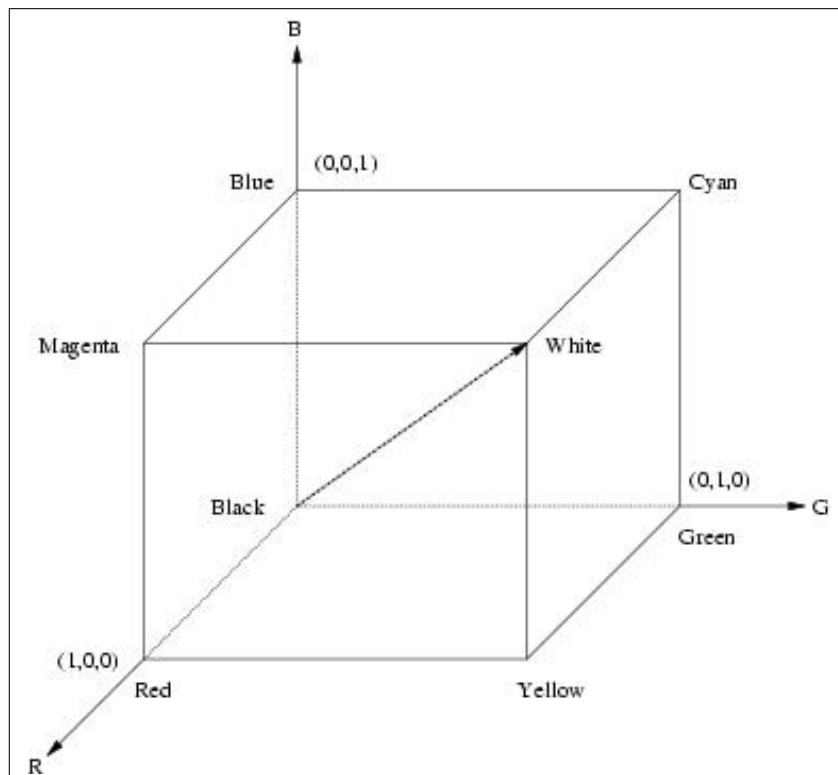
Gambar 2.4: Warna Campuran pada Diagram Chromaticity

2.3 Model Warna

Model warna merupakan cara standar untuk menspesifikasikan suatu warna tertentu, dengan mendefinisikan suatu sistem koordinat 3D, dan suatu ruang bagian yang mengandung semua warna yang dapat dibentuk ke dalam suatu model tertentu. Suatu warna yang dapat dispesifikasikan menggunakan suatu model akan berhubungan ke suatu titik tunggal dalam suatu ruang bagian yang didefinisikannya. Masing-masing warna diarahkan ke salah satu standard hardware tertentu (RGB, CMY, YIQ), atau aplikasi pengolahan citra (HSI).

2.3.1 Model RGB

Suatu citra dalam model RGB terdiri dari tiga bidang citra yang saling lepas, masing-masing terdiri dari warna utama: merah, hijau dan biru. (Standar panjang gelombang untuk tiga warna utama ditunjukkan pada gambar 2.2). Suatu warna dispesifikasikan sebagai campuran sejumlah komponen warna utama. Gambar 2.5 menunjukkan bentuk geometri dari model warna RGB untuk menspesifikasikan warna menggunakan sistem koordinat Cartesian. Spektrum greyscale (tingkat keabuan) yaitu warna yang dibentuk dari gabungan tiga warna utama dengan jumlah yang sama, berada pada garis yang menghubungkan titik hitam dan putih.

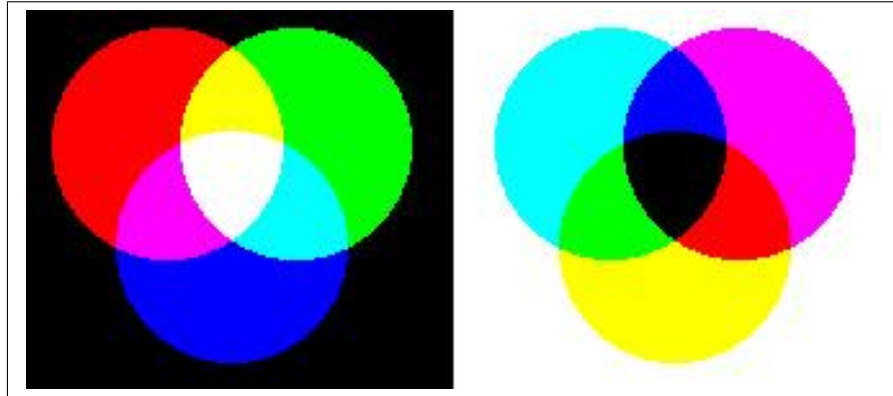


Gambar 2.5: Koordinat warna RGB

Warna dipresentasikan dalam suatu sinar tambahan untuk membentuk warna baru, dan berhubungan untuk membentuk sinar campuran. Citra pada gambar 2.6 sebelah kiri menunjukkan campuran dengan menambahkan warna utama merah, hijau, dan biru untuk membentuk warna sekunder

2.3. MODEL WARNA

kuning (merah+hijau), cyan (biru+hijau), magenta (merah+biru) dan putih (merah+hijau+biru). Model warna RGB banyak digunakan untuk monitor komputer dan video kamera.



Gambar 2.6: Penambahan Campuran Warna Merah Hijau dan Biru

2.3.2 Model CMY

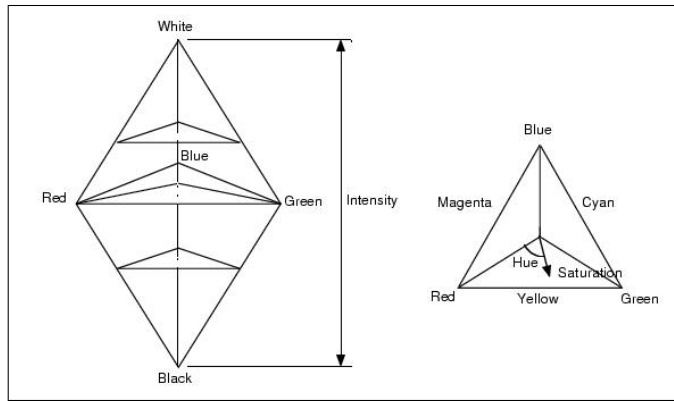
Model CMY (Cyan, Magenta dan Yellow) adalah suatu model subtractive yang berhubungan dengan penyerapan warna, sebagai contoh pigment warna cat (seperti pada Gambar 2.6 sebelah kanan). Suatu permukaan yang dicat warna cyan kemudian diiluminasi sinar putih, maka tidak ada sinar merah yang dipantulkan, dan similar untuk warna magenta dengan hijau, dan kuning dengan biru. Relasi model CMY adalah sebagai berikut:

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

2.3.3 Model HSI

The HSI model, showing the HSI solid on the left, and the HSI triangle on the right, formed by taking a horizontal slice through the HSI solid at a particular intensity. Hue is measured from red, and saturation is given by distance from the axis. Colours on the surface of the solid are fully saturated, i.e. pure colours, and the greyscale spectrum is on the axis of the solid. For these colours, hue is undefined.

Sebagaimana yang sudah dijelaskan, warna juga dapat dispesifikasikan oleh tiga kuantisasi *hue*, *saturation*, *intensity* (disebut model HSI) seperti pada gambar 2.7. Pada gambar sebelah kiri merupakan bentuk solid HSI dan sebelah kanan adalah model segitiga HSI yang merupakan bidang datar dari pemotongan model solid HSI secara horisontal pada tingkat intensitas tertentu. *Hue* ditentukan dari warna merah, saturation ditentukan berdasarkan jarak dari sumbu. Warna pada permukaan model solid dibentuk dari saturasi penuh, yaitu warna murni, dan spektrum tingkat keabuan,



Gambar 2.7: Model warna HSI

Konversi nilai antar model RGB dan HSI adalah sebagai berikut:

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

dimana kuantitas R, G, dan B adalah jumlah komponen warna merah, hijau, biru dan dinormilisasi ke [0,1]. Intensitas adalah nilai rata-rata komponen merah, hijau dan biru. Nilai saturation ditentukan sebagai:

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

2.4 Model YIQ

The YIQ (luminance-inphase-quadrature) model is a recoding of RGB for colour television, and is a very important model for colour image processing. The importance of luminance was discussed in 1.

The conversion from RGB to YIQ is given by:

2.5. APPLYING GREYSCALE TRANSFORMATIONS TO COLOUR IMAGES

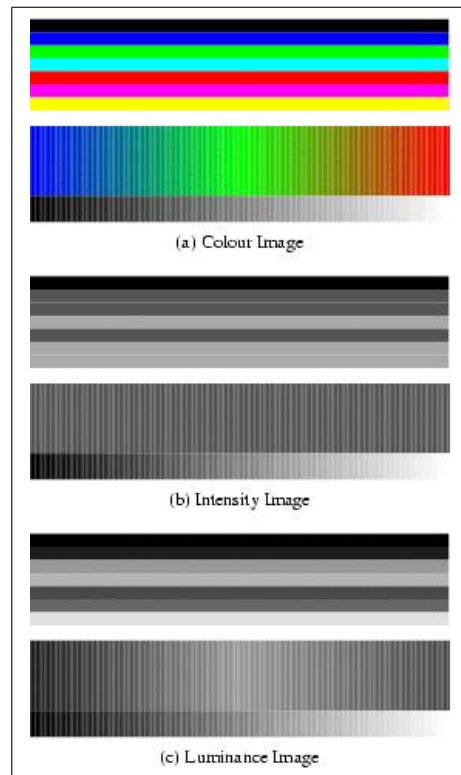
$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

The luminance (Y) component contains all the information required for black and white television, and captures our perception of the relative brightness of particular colours. That we perceive green as much lighter than red, and red lighter than blue, is indicated by their respective weights of 0.587, 0.299 and 0.114 in the first row of the conversion matrix above. These weights should be used when converting a colour image to greyscale if you want the perception of brightness to remain the same. This is not the case for the intensity component in an HSI image, as shown in figure 8. The Y component is the same as the CIE primary Y (see bagian 3.2).

2.5 Applying Greyscale Transformations to Colour Images

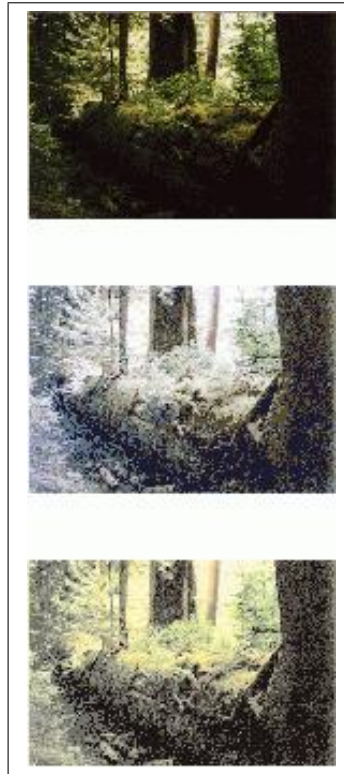
Given all these different representations of colour, and hence colour images, the question arises as to what is the best way to apply the image processing techniques we have covered so far to these images? One possibility is to apply the transformations to each colour plane in an RGB image, but what exactly does this mean? If we want to increase the contrast in a dark image by histogram equalisation, can we just equalise each colour independently? This will result in quite different colours in our transformed image. In general it is better to apply the transformation to just the intensity component of an HSI image, or the luminance component of a YIQ image, thus leaving the chromaticity unaltered.

An example is shown in figure 2.9. When histogram equalisation is applied to each colour plane of the RGB image, the final image is lighter, but also quite differently coloured to the original. When histogram equalisation is only applied to the luminance component of the image in YIQ format, the result is more like a lighter version of the original image, as required.



Gambar 2.8: Image (a) shows a colour test pattern, consisting of horizontal stripes of black, blue, green, cyan, red, magenta and yellow, a colour ramp with constant intensity, maximal saturation, and hue changing linearly from red through green to blue, and a greyscale ramp from black to white. Image (b) shows the intensity for image (a). Note how much detail is lost. Image (c) shows the luminance. This third image accurately reflects the brightness variations perceived in the original image.

2.5. APPLYING GREYSCALE TRANSFORMATIONS TO COLOUR IMAGES



Gambar 2.9: The top image is a very dark image of a forest scene. The middle image is the result of applying histogram equalisation to each of the red, green and blue components of the original image. The bottom image is the result of converting the image to YIQ format, and applying histogram equalisation to the luminance component only.

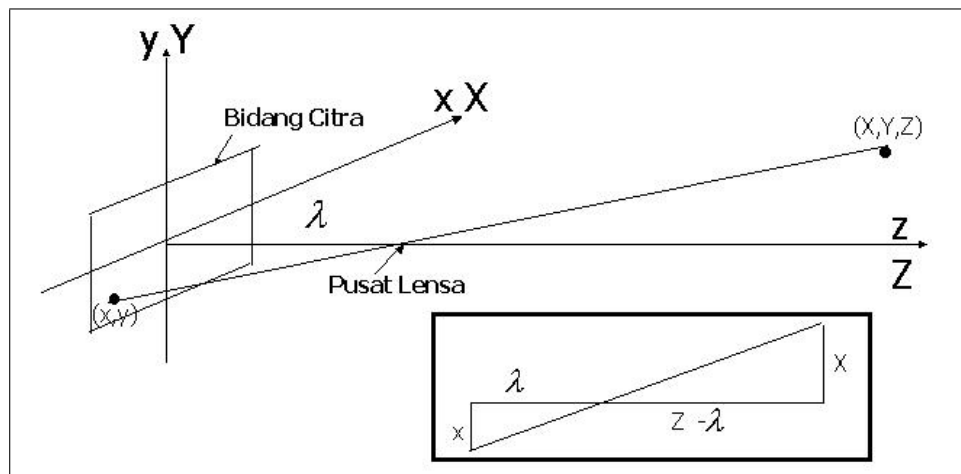
BAB 2. WARNA

Bab 3

Perspektif dan Transformasi Citra

3.1 Koordinat Kamera dan Koordinat World

Suatu citra diperoleh dari obyek suatu Sistem koordinat dapat dinyatakan ke dalam dua bentuk



Gambar 3.1: Visible Spectrum

Bila kedua sistem sumbu (camera dan world) pada gambar 3.1 dihimpitkan, maka obyek (pada ruang world) dan bayangan (pada bidang citra) akan membentuk segitiga sama dan sebangun, sehingga bentuk transformasi dari

koordinat 3D world (X, Y, Z) ke koordinat kamera (x, y, z) adalah sbb:

$$\frac{x}{\lambda} = \frac{X}{Z-\lambda}$$

dan

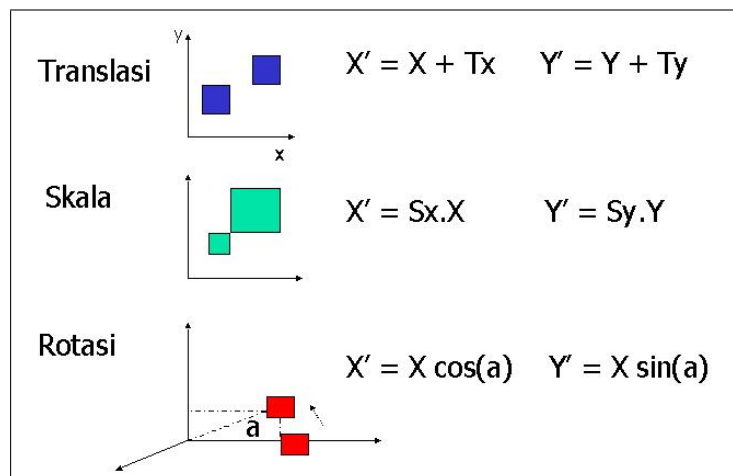
$$x = \frac{\lambda X}{\lambda - Z}$$

$$y = \frac{\lambda Y}{\lambda - Z}$$

$$z = \frac{\lambda Z}{\lambda - Z}$$

3.2 Transformasi Geometrik

Transformasi Geometrik adalah transformasi berdasarkan perpindahan geometrik suatu titik. Transformasi ini terdiri dari translasi, skala dan rotasi. Gambar 3.2 adalah contoh dari masing-masing transformasi geometrik untuk model 2 dimensi.



Gambar 3.2: Visible Spectrum

Diperlukan suatu representasi yang seragam (homogeneous representation) agar memungkinkan dilakukannya transformasi komposit secara efisien dan dapat menyimpan faktor normalisasi koordinat akibat transformasi yang dilakukan berturut-turut. Representasi hogeneous ini biasanya dalam bentuk matriks.

3.2. TRANSFORMASI GEOMETRIK

Secara umum, transformasi geometrik ini dilakukan melalui proses perkalian dengan matriks transformasi. Bentuk matriks transformasi tersebut adalah sebagai berikut:

- Translasi sejauh (T_x, T_y, T_z)

$$\begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Rotasi dengan sumbu pusat sebesar a derajat.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(a) & \sin(a) & 0 \\ 0 & -\sin(a) & \cos(a) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Skala dengan faktor S

$$\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Matrix transformasi perspektif

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

Tanda minus artinya gambar obyek terbalik, λ adalah jarak pusat lensa, dan $\frac{1}{\lambda}$ merupakan faktor skala. Koordinat obyek pada camera system dapat diturunkan dari koordinat obyek pada world system dengan menggunakan transformasi perspektif.

3.3 Sistem Koordinat Cartesian dan Homogeneous

Koordinat obyek pada *world system* dalam bentuk sistem koordinat Cartesian (W_c) dan homogeneous coordinate system (W_h). Bentuk ke dua sistem tersebut adalah:

$$W_c = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad W_h = \begin{pmatrix} kX \\ kY \\ kZ \\ k \end{pmatrix}$$

k merupakan konstanta *non-zero* dan biasanya diambil nilai $k = 1$.

Sistem koordinat obyek pada camera adalah C_c dan C_h masing-masing untuk sistem koordinat Cartesian dan homogeneous coordinate adalah sebagai berikut:

$$C_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{\lambda} & 1 \end{bmatrix} \begin{pmatrix} kX \\ kY \\ kZ \\ k \end{pmatrix} = \begin{bmatrix} kX \\ kY \\ kZ \\ -\frac{kZ}{\lambda} + k \end{bmatrix}$$

Koordinat Cartesian $C_c(x, y, z)$ diperoleh dengan membagi koordinat $C_h(xh, yh, zh)$ dengan faktor koordinat ke empat, dalam hal ini yaitu $-\frac{kZ}{\lambda} + k$ sehingga bentuk sistem koordinat cartesian camera adalah:

$$C_c = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{bmatrix} \frac{kX}{-\frac{kZ}{\lambda} + k} \\ \frac{kY}{-\frac{kZ}{\lambda} + k} \\ \frac{kZ}{-\frac{kZ}{\lambda} + k} \end{bmatrix} = \begin{bmatrix} X(\lambda - Z) \\ Y(\lambda - Z) \\ Z(\lambda - Z) \end{bmatrix}$$

Hubungan antara (x, y, z) dan (X, Y, Z) diatas disebut sebagai Camera Basic Mathematical Model.

3.4 Transformasi Image ke World

Suatu titik obyek $(X_0, Y_0, 0)$ terletak di bidang citra, dengan camera system dan world system berhimpit dan bidang citra terletak pada $Z = 0$, maka koordinat homogeneous dari obyek tersebut pada world system adalah:

$$W_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{1}{\lambda} & 1 \end{bmatrix} \begin{pmatrix} kX_0 \\ kY_0 \\ 0 \\ k \end{pmatrix} = \begin{bmatrix} kX_0 \\ kY_0 \\ 0 \\ k \end{bmatrix}$$

Titik (X_0, Y_0) merupakan titik proyeksi seluruh titik-titik 3-D yang terletak pada garis yang melalui $(X_0, Y_0, 0)$ dan $(0, 0, \lambda)$.

Persamaan garis yang melalui titik $(X_0, Y_0, 0)$ dan $(0, 0, \lambda)$ adalah: (lihat penurunan dari rumus segitiga sebangun yang menghasilkan hubungan antara camera dan world system)

$$X = \frac{X_0}{\lambda}(\lambda - Z)$$

$$Y = \frac{Y_0}{\lambda}(\lambda - Z)$$

Dengan demikian kita tidak dapat menentukan titik 3-D hanya dari proyeksi titik tersebut pada bidang citra tanpa diketahuinya koordinat Z pada ruang 3-D tersebut.

Jika terdapat suatu titik pada citra (X_0, Y_0, z) dimana z adalah variabel bebas yang menyatakan kedalaman atau jarak, maka:

$$C_h = \begin{pmatrix} kX_0 \\ kY_0 \\ kz \\ k \end{pmatrix} W_h = \begin{pmatrix} kX_0 \\ kY_0 \\ kz \\ \frac{kz}{\lambda} + k \end{pmatrix}$$

Titik 3-D nya adalah:

$$X = \frac{\lambda X_0}{(\lambda + z)}$$

$$Y = \frac{\lambda Y_0}{(\lambda + z)}$$

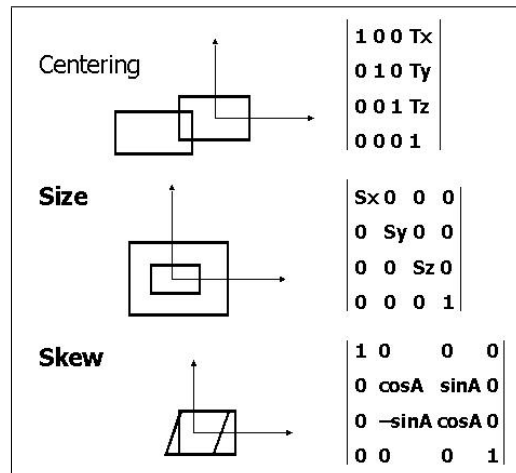
$$Z = \frac{\lambda z}{(\lambda + z)}$$

3.5 Distorsi Geometrik

Distorsi geometrik merupakan distorsi spasial, yaitu terjadi pergeseran posisi spasial citra yang ditangkap. Distorsi geometrik ini disebabkan oleh kesalahan yang terjadi seperti kerusakan sensor (internal), platform (external) dan

BAB 3. PERSPEKTIF DAN TRANSFORMASI CITRA

gerakan bumi. Koreksi yang dilakukan bila terjadi distorsi bersifat sederhana, seperti centering (translasi), size (skala), skew (rotasi). Gambar 3.3 menggambarkan matriks transformasi. Koreksi geometrik bila terdapat distorsi yang bersifat kompleks adalah image registration/rectification, misal dengan bilinear transformation dan least square method, seperti berikut:

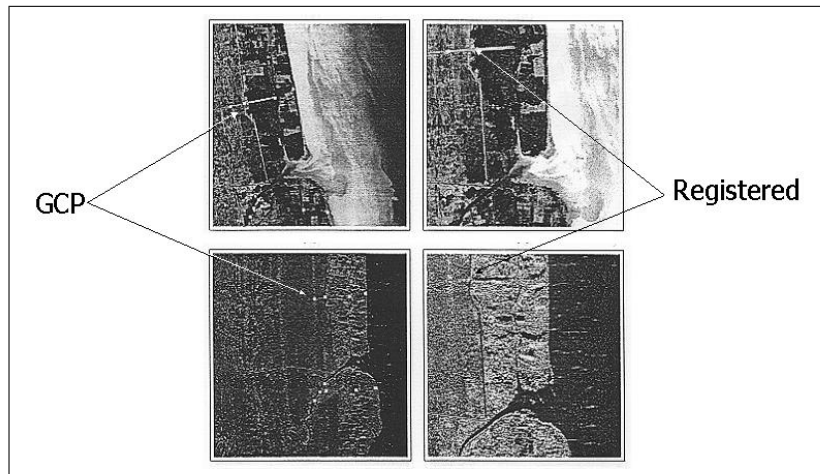


Gambar 3.3: Visible Spectrum

$$\begin{aligned} X &= aX + bY + cXY + d \\ Y &= eX + fY + gXY + h \end{aligned} \quad (3.1)$$

Gambar 3.4 menunjukkan suatu contoh adanya geometri distorsi, dan akan direstorasi menggunakan interpolasi berdasarkan titik kontrol daratan (Ground Control Point (GCP)) yang diambil langsung dengan menggunakan teknologi seperti Global Position System (GPS). Titik-titik tersebut dibandingkan dengan posisi titik tersebut di citra.

Persamaan yang digunakan untuk mendapatkan koreksi posisi secara pergeseran geometrik adalah dengan menggunakan metode transformasi bilinear dan least square seperti pada persamaan 3.1. Jumlah pasangan persamaan diatas adalah sebanyak ground control points yang digunakan. Salah satu citra dijadikan acuan (koordinat piksel (X, Y)), maka koordinat piksel citra yang diregistrasi (X, Y) dapat dihitung dari persamaan diatas dengan menyelesaikan koefisien a, b, c , dan d .



Gambar 3.4: Visible Spectrum

3.6 Distorsi Radiometrik

Distorsi Radiometrik muncul dalam bentuk distribusi intensitas yang tidak tepat. Sumber distorsi ini adalah kamera (internal) dalam bentuk shading effect, atmosfer (external) dalam bentuk besarnya intensitas yang tidak sama walaupun untuk obyek yang kategorinya sama, akibat adanya kabut, posisi matahari atau substansi atmosfer lainnya. Koreksi yang dilakukan untuk jenis distorsi ini adalah dengan teknik filtering.

Gambar 3.5 adalah contoh adanya distorsi dalam bentuk skew (geometrik external - rotasi) dan adanya striping (radiometrik internal low pass filter). Sedangkan gambar 3.6 menunjukkan adanya distorsi radiometrik dengan frekuensi rendah. Restorasinya dilakukan dengan menggunakan high pass filter. Gambar 3.4 menunjukkan adanya distorsi radiometrik dengan frekuensi tinggi yang ditunjukkan dengan adanya semacam garis-garis (*stripping*). Restorasi dapat dilakukan dengan menggunakan low pass filter untuk menghilangkan *stripping* tersebut.

3.7 Impulse

Salah satu bentuk contoh dari satu fungsi impulse adalah fungsi Delta Dirac pada domain kontinue dan Fungsi Delta Kronecker pada domain diskrit $d(x)$ yang mempunyai nilai 1 pada suatu x dan mempunyai nilai 0 pada x lainnya.



Gambar 3.5: Distorsi Radiometrik dan Geometrik

Bentuk dari Fungsi Delta tersebut adalah seperti Gambar 3.7.

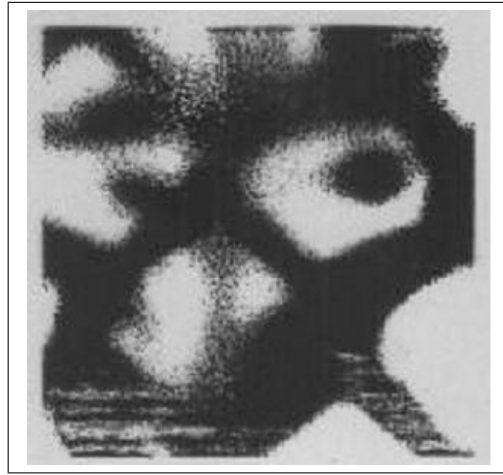
Menurut teori filtering, pada sistem yang ideal, sinyal yang masuk (impulse) sama dengan sinyal yang keluar (impulse response). Hal tersebut dapat digambarkan dengan transfer function dalam bentuk fungsi Delta Dirac, seperti pada gambar 3.8.

Pada sistem yang tidak ideal, sinyal yang masuk mengalami degradasi atau penurunan kualitas. Misalkan adanya low frekuensi seperti *blurring* sebagai akibat adanya point spread function (PSF) yang ditunjukkan pada gambar 3.9.

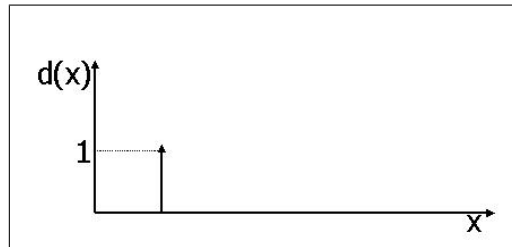
Adanya Impulse tersebut mengakibatkan perlunya memperbaiki citra agar sesuai dengan yang diinginkan. Untuk mengetahui sebaran spectrum suatu citra agar dapat mengetahui/memprediksi dampak impuls yang terjadi maka digunakan transformasi Fourier yang mengubah representasi citra dari domain spasial ke domain frekwensi. Sebaliknya Inverse Fourier Transform (IFT) akan mengubah representasi citra dari domain frekwensi ke domain spasial ($g(x,y) \rightarrow FT \rightarrow G(u,v) \rightarrow IFT \rightarrow g(x,y)$). FT ini akan memudahkan proses konvolusi dari bentuk integral menjadi bentuk perkalian biasa.

Pada sistem yang linier, Fourier transform dari suatu impulse menghasilkan $G(u,v)$ dan disebut sebagai system transfer function. Sebaliknya $g(x,y)$ sebagai inverse transform dari system transfer function disebut sebagai impulse response. Pada sistem optik, $g(x,y)$ merupakan inverse transform dari optical transfer function $G(u,v)$, dan disebut point spread function.

3.7. IMPULSE

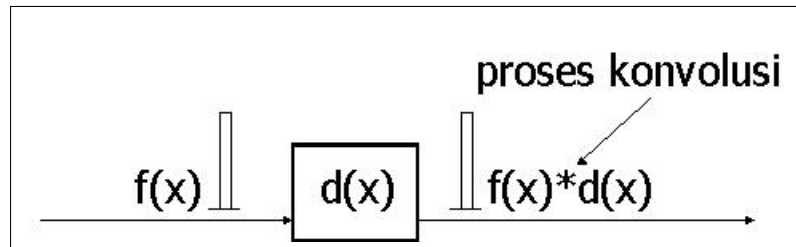


Gambar 3.6: Distorsi Radiometrik *blurring*

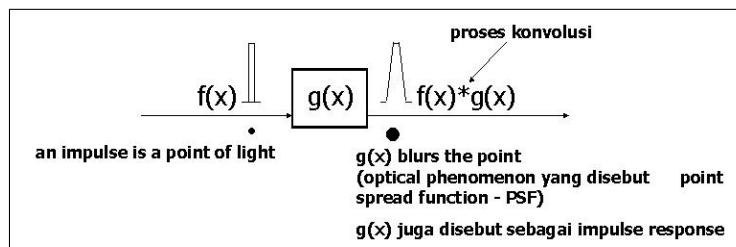


Gambar 3.7: Fungsi Delta

Optical transfer function dan point spread function masing-masing adalah Fourier transform dari yang lainnya.



Gambar 3.8: Fungsi Konvolusi Impulse



Gambar 3.9: Point Spread Function

Bab 4

Transformasi Fourier

Mengapa perlu transformasi ?

Setiap orang pada suatu saat pernah menggunakan suatu teknik analisis dengan transformasi untuk menyederhanakan penyelesaian suatu masalah [Brigham,1974] misalkan penyelesaian fungsi $y = x/z$. Solusi penyelesaian persamaan tersebut dapat dilakukan secara analisis konvensional yaitu dengan pembagian secara manual. Sedangkan secara analisis transformasi yaitu melakukan transformasi:

$$\log(y) = \log(x) - \log(z)$$

look-up table \rightarrow pengurangan \rightarrow look-up table.

Transformasi juga diperlukan ketika ingin mengetahui suatu informasi tertentu yang tidak tersedia sebelumnya. Misalkan jika ingin mengetahui informasi frekuensi kita memerlukan transformasi Fourier. Sedangkan jika ingin mengetahui informasi tentang kombinasi skala dan frekuensi kita memerlukan transformasi wavelet.

4.1 Transformasi Citra

Transformasi citra, sesuai namanya, merupakan proses perubahan bentuk citra untuk mendapatkan suatu informasi tertentu. Transformasi bisa dibagi menjadi dua, yaitu Transformasi piksel/transformasi geometris dan transformasi ruang/domain/space. Transformasi piksel masih bermain di ruang/domain yang sama (domain spasial), hanya posisi piksel yang kadang diubah, misalkan rotasi, translasi, scaling, invers, shear, dll. Transformasi

jenis ini relatif mudah diimplementasikan dan banyak aplikasi yang dapat melakukannya (Paint, ACDSee, dll). Transformasi ruang merupakan proses perubahan citra dari suatu ruang/domain ke ruang/domain lainnya, sebagai contoh dari ruang spasial ke ruang frekuensi. Seperti halnya istilah ruang pada Aljabar Linier menjelaskan tentang Basis dan Ruang, misalkan Ruang vektor. Salah satu basis yang merentang ruang vektor 2 dimensi adalah $[10]$ dan $[01]$. Artinya, semua vektor yang mungkin ada di ruang vektor 2 dimensi selalu dapat direpresentasikan sebagai kombinasi linier dari basis tersebut.

Ada beberapa transformasi ruang yang akan dipelajari, yaitu :

- Transformasi Fourier (basis: cos-sin)
- Transformasi Hadamard/Walsh (basis: kolom dan baris yang ortogonal)
- Transformasi DCT (basis: cos) Transformasi Wavelet (basis: scaling function dan mother wavelet)

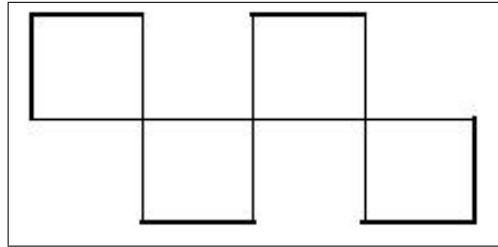
4.2 Transformasi Fourier

Pada tahun 1822, Joseph Fourier, ahli matematika dari Prancis menemukan bahwa: setiap fungsi periodik (sinyal) dapat dibentuk dari penjumlahan gelombang-gelombang sinus/cosinus. Contoh : Sinyal kotak (gambar 4.1) dapat didekatkan dengan penjumlahan dari fungsi-fungsi sinus (gambar 4.2 dengan pendekatan fungsi $f(x) = \sin(x) + \sin(3x)/3 + \sin(5x)/5 + \sin(7x)/7 + \sin(9x)/9 \dots$). Gambar pendekatan fungsi Fourier tersebut menggunakan sejumlah iterasi masing-masing 1, 3, 7, 99 yang dihasilkan dari algoritma matlab 1

Algorithm 1 Program Matlab

```
function kotak(n)
t = 0:pi/200:8*pi;
kot = sin(t);
for i = 3 : 2: n
kot = kot + (sin(i*t))/i;
end
plot(kot)
```

4.2. TRANSFORMASI FOURIER



Gambar 4.1: Fungsi Kotak

Jika semua sinyal periodik dapat dinyatakan dalam penjumlahan fungsi-fungsi sinus-cosinus, pertanyaan berikutnya yang muncul adalah: Jika terdapat sebuah sinyal sembarang, bagaimana dapat diketahui bentuk fungsi-fungsi \cos \sin apa yang membentuknya? Atau dengan kata lain Berapakah frekuensi yang dominan di sinyal tersebut? Pertanyaan di atas dapat dijawab dengan menghitung nilai $F(u)$ dari sinyal tersebut. Dari nilai $F(u)$ kemudian dapat diperoleh kembali sinyal awal dengan menghitung $f(x)$, menggunakan Inverse Fourier.

4.2.1 Fourier 1 Dimensi

Bentuk persamaan Fourier 1 Dimensi terbagi menjadi dua, yaitu:

- Rumus FT kontinu 1 dimensi

$$\begin{aligned} F(u) &= \int_{-\infty}^{\infty} f(x) \exp[-2j\pi ux] dx \\ f(x) &= \int_{-\infty}^{\infty} F(u) \exp[2j\pi ux] du \end{aligned}$$

$$\text{Rumus Euler: } \exp[-2j\pi ux] = \cos 2\pi ux - j \sin 2\pi ux$$

- Rumus Fourier Transform diskret 1 dimensi

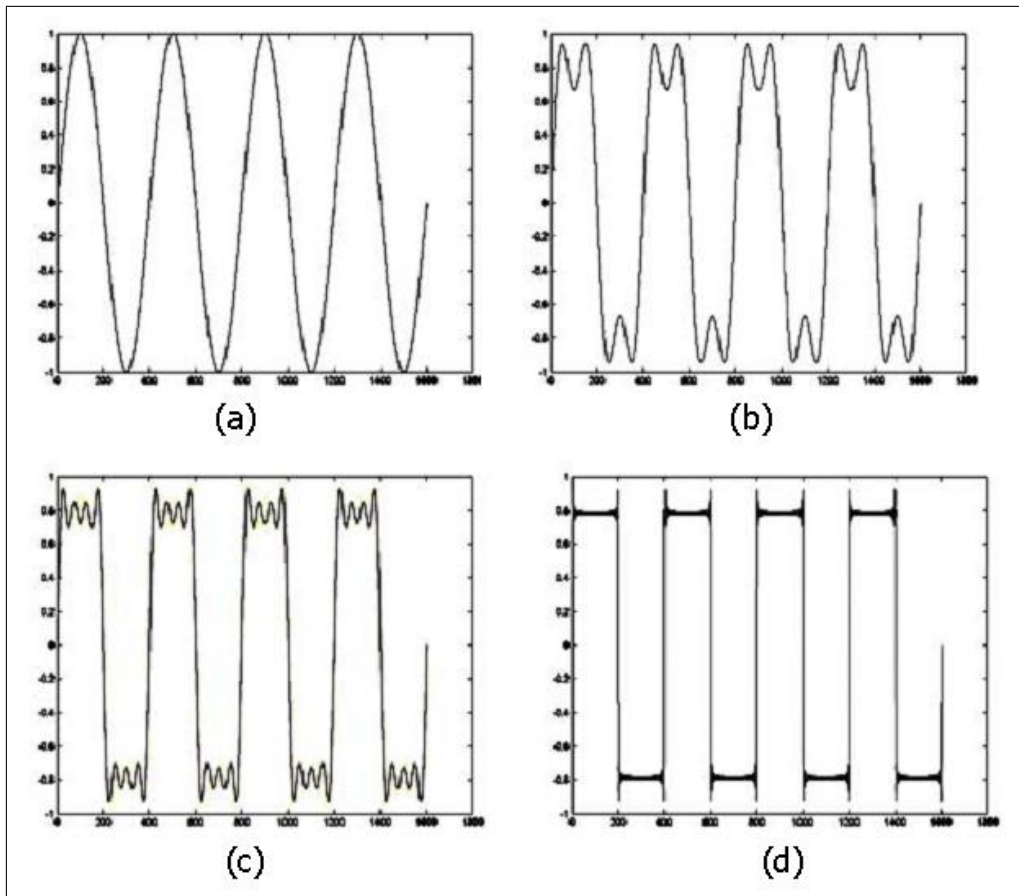
$$\begin{aligned} F(u) &= \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-2j\pi ux/N] \\ f(x) &= \frac{1}{N} \sum_{u=0}^{N-1} F(u) \exp[2j\pi ux/N] \end{aligned}$$

Misalkan kita memiliki sinyal $x(t)$ dengan rumus sbb:

$$x(t) = \cos(2\pi 5t) + \cos(2\pi 10t) + \cos(2\pi 20t) + \cos(2\pi 50t)$$

Sinyal ini memiliki empat komponen frekuensi yaitu 5,10,20,50 (Gambar 4.3).

FT dari sinyal tersebut (Gambar 4.4). Terlihat bahwa FT dapat menangkap frekuensi-frekuensi yang dominan dalam sinyal tersebut, yaitu 5,10, 20, 50. (nilai maksimum $F(u)$ berada pada angka 5,10, 20, 50).



Gambar 4.2: Gambar Fungsi Fourier a) $n = 1$, b) $n = 3$, c) $n = 7$, d) $n = 99$

Contoh Penghitungan FT 1 dimensi:

$$\begin{aligned} F(u) &= \frac{1}{N} \sum_{x=0}^{N-1} f(x) \exp[-2j\pi ux/N] \\ &= \frac{1}{N} \sum_{x=0}^{N-1} f(x) (\cos(2\pi ux/N) - j \sin(2\pi ux/N)) \end{aligned}$$

$$\text{contoh : } f(0) = 2; f(1) = 3; f(2) = 4; f(3) = 4$$

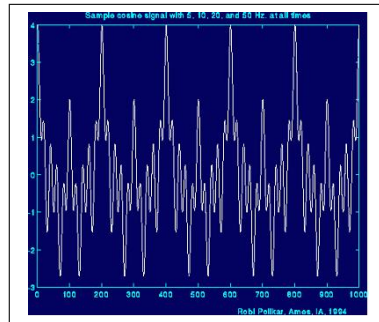
$$\begin{aligned} F(0) &= \frac{1}{N} \sum_{x=0}^{N-1} f(x) (\cos(2\pi 0x/N) - j \sin(2\pi 0x/N)) \\ &= \frac{1}{4} [f(0) + f(1) + f(2) + f(3)] = 3.25 \end{aligned}$$

$$\begin{aligned} F(1) &= \frac{1}{N} \sum_{x=0}^{N-1} f(x) (\cos(2\pi x/N) - j \sin(2\pi x/N)) \\ &= \frac{1}{4} [2(1 - 0) + 3(0 - j) + 4(-1) + 4(j)] = -0.5 + 0.25j \end{aligned}$$

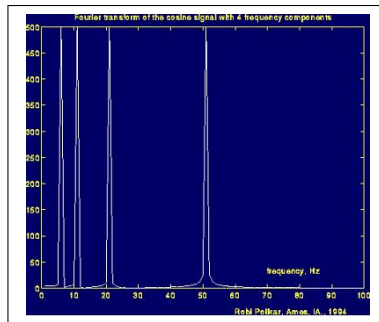
$$F(2) = -0.25$$

$$F(3) = -0.5 - 0.25j$$

4.2. TRANSFORMASI FOURIER



Gambar 4.3: Fungsi Cos Sin



Gambar 4.4: Fourier Transform

Hasil penghitungan FT biasanya mengandung bilangan real dan imajiner, sehingga Fourier Spectrum didapatkan dari magnitude kedua bilangan tersebut shg $|F(u)| = [R^2(u) + I^2(u)]^{1/2}$. Untuk contoh sebelumnya, Fourier Spectrumnya adalah sebagai berikut:

$$\begin{aligned} |F(0)| &= 3.25 \\ |F(1)| &= 1/2 = 0.5590 \\ |F(2)| &= 0.25 \\ |F(3)| &= [(0.5)^2 + (0.25)^2]^{1/2} = 0.5590 \end{aligned}$$

4.2.2 Rumus FT 2 Dimensi

Secara umum rumus Fourier 2 Dimensi adalah sebagai berikut:

- Fourier Transform

- Kontinu

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp[-2j\pi(ux + vy)] dx dy$$

BAB 4. TRANSFORMASI FOURIER

- Diskrete

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \exp[-2j\pi(ux/M + vy/N)] dx dy$$

- Inverse Fourier

- Kontinu

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp[2j\pi(ux + vy)] du dv$$

- Diskrete

$$f(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u, v) \exp[2j\pi(ux/M + vy/N)]$$

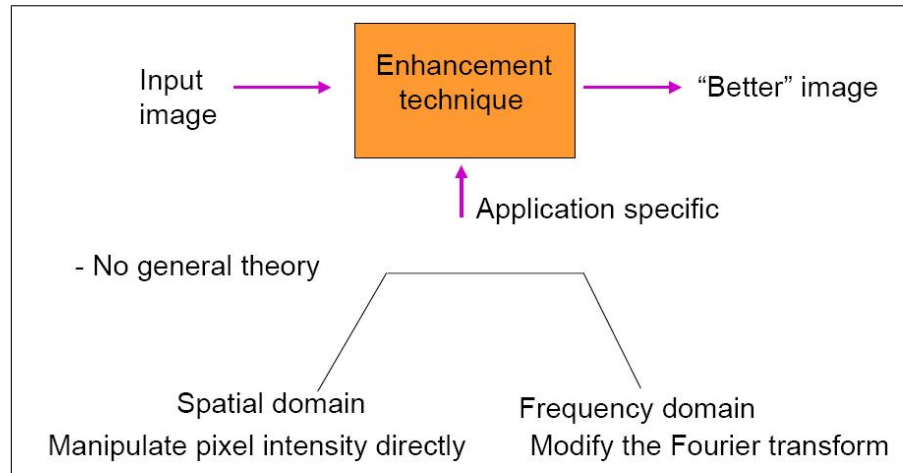
$M = \text{TinggiCitra}(\text{jumlahbaris})$

$N = \text{LebarCitra}(\text{jumlahkolom})$

Bab 5

Peningkatan Mutu Citra

Proses peningkatan mutu citra bertujuan untuk memperoleh citra yang dapat memberikan informasi sesuai dengan tujuan/kepentingan pengolahan citra. Proses peningkatan mutu citra ini termasuk memperbaiki citra yang ketika proses akuisisi mengalami gangguan yang signifikan seperti noise, gangguan geometris, radiometrik dan beberapa gangguan faktor alam lainnya.



Gambar 5.1: Proses Peningkatan Mutu Citra

Suatu metode pendekatan peningkatan mutu citra yang terbaik untuk satu implementasi belum tentu baik untuk implementasi lainnya, sebab karakteristik citra dapat saling berbeda. Gambar 5.1 menunjukkan proses peningkatan citra. Secara umum domain dalam peningkatan mutu citra ini

dapat dilakukan secara spatial dan frekuensi. Domain Spatial melakukan manipulasi nilai pixel secara langsung dengan dipengaruhi oleh nilai pixel lainnya secara spatial sedangkan domain frekuensi berdasarkan frekuensi spektrum citra. Terdapat beberapa teknik peningkatan mutu citra yang merupakan kombinasi dari dua kategori ini.

Peningkatan mutu citra dapat dikaitkan dengan metode Filtering, dimana citra tersebut di-*filter* untuk mendapatkan citra yang lebih baik. Jenis Filter dapat dikategorikan menjadi dua jenis, yaitu linear dan non-linear filtering. Linear filtering adalah mengubah nilai pixel berdasarkan kombinasi linear nilai pixel disekelilingnya.

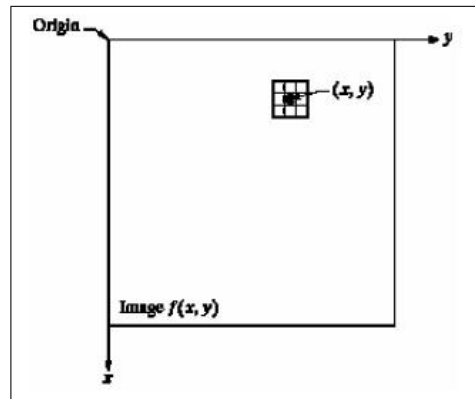
- Basic gray level transformations.
- Histogram Modification.
- Average and Median Filtering.
- Frequency domain operations.
- Homomorphic Filtering.
- Edge enhancement.

5.1 Domain Spatial

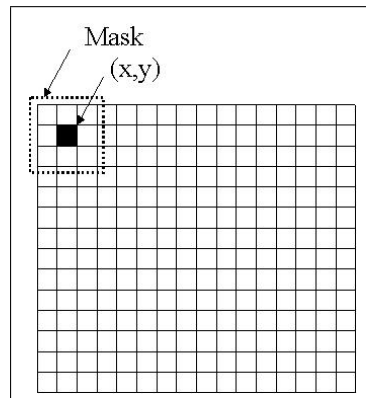
Merupakan teknik peningkatan mutu citra yang melakukan manipulasi langsung pixel (x,y) suatu citra dengan menggunakan fungsi transformasi: $g(x,y) = T[f(x,y)]$, dimana $f(x,y)$ sebagai citra input, $g(x,y)$ hasil citra yang sudah diproses dan T adalah operator pada f yang didefinisikan berdasarkan beberapa lingkungan di (x,y) . Teknik ini ditunjukkan pada gambar 5.2

Masking/Filter suatu pixel (x,y) ditentukan berdasarkan pixel tetangganya yang didefinisikan sebagai bentuk bujur sangkar (sering digunakan) ataupun circular sebagai sub-citra yang berpusat di titik (x,y) dengan ukuran lebih dari 1×1 (gambar 5.3 menggunakan masking 3×3). Pusat sub-citra berpindah dari satu pixel ke pixel lainnya dimulai dari pojok atas. Nilai koefisien masking ditentukan berdasarkan prosesnya. Teknik masking digunakan untuk penajaman citra dan penghalusan citra.

5.1. DOMAIN SPATIAL



Gambar 5.2: Teknik Spatial merubah langsung pixel (x,y)



Gambar 5.3: Masking pixel (x,y)

5.1.1 Point Processing

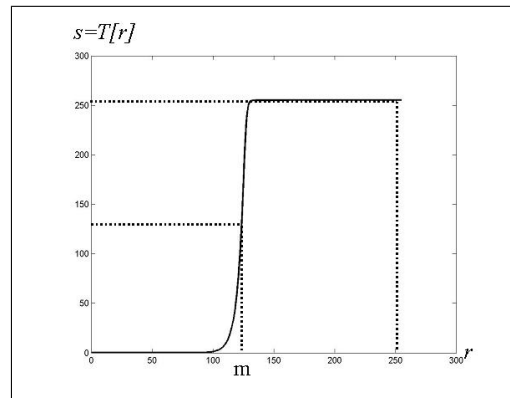
Pada proses titik, ukuran masking tetangga adalah 1×1 , dimana g bergantung hanya pada nilai f di titik (x, y) (satu titik/pixel). T yang merupakan fungsi transformasi memetakan f ke suatu nilai tingkat keabuan tertentu s , dimana $s = T(r)$. r adalah tingkat keabuan $f(x, y)$ dan s adalah tingkat keabuan $g(x, y)$.

5.1.2 Contrast Stretching dan Thresholding

Mengubah nilai kontras pixel dari pixel aslinya dengan ketentuan (gambar 5.4):

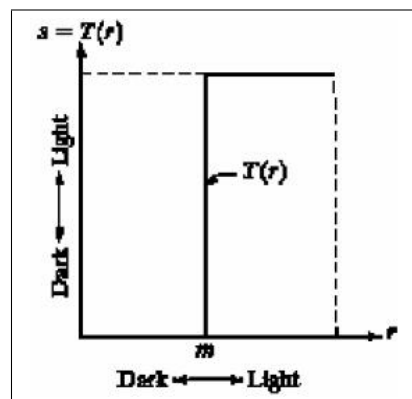
BAB 5. PENINGKATAN MUTU CITRA

- Menggelapkan pixel dengan nilai tingkat keabuan yang lebih rendah dari m
- Meningkatkan kontras apabila nilai tingkat keabuan pixel lebih dari m



Gambar 5.4: Contrast Stretching

Thresholding adalah mentransformasikan citra menjadi citra biner, yaitu citra dengan dua warna. Gambar 5.7 menunjukkan bahwa jika nilai tingkat keabuan suatu pixel $0 \leq r < m$ maka akan ditransformasi ke nilai 0, sebaliknya untuk nilai pixel $m \leq r \leq (L - 1)$ akan ditransformasikan ke nilai $L - 1$.

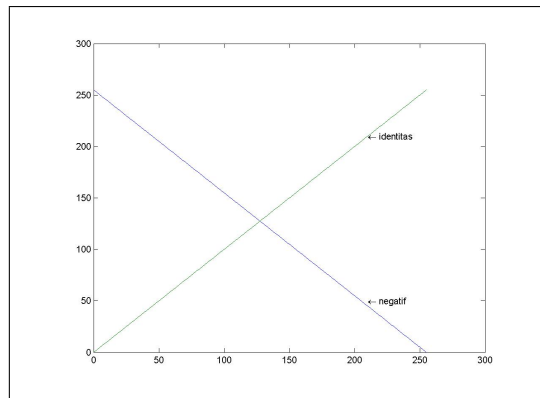


Gambar 5.5: Threshold

5.1.3 Tiga Fungsi Transformasi Dasar

Fungsi Linear

Fungsi linear adalah berbasiskan pada model $s = mr + c$, yaitu membentuk korelasi garis lurus antara input dan output. Transformasi linear yang sering digunakan adalah fungsi identitas dan negatif. Input dan output citra untuk transformasi fungsi identitas adalah sama, biasanya digunakan untuk melengkapi atau kombinasi dengan teknik transformasi lain ($s = r$). Kebalikannya untuk transformasi negatif, nilai tingkat keabuan citra output merupakan negatif dari nilai keabuan citra input ($s = (l - 1) - r$) (seperti gambar 5.6)

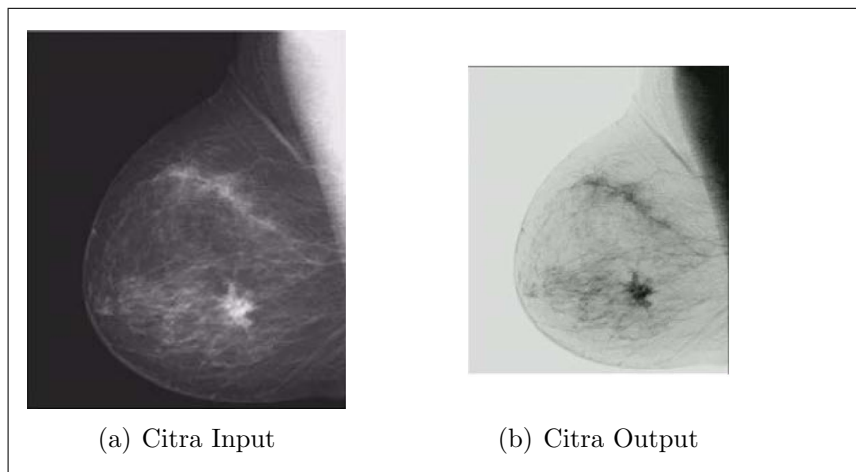


Gambar 5.6: Fungs Identitas dan Negatif

Fungsi Log

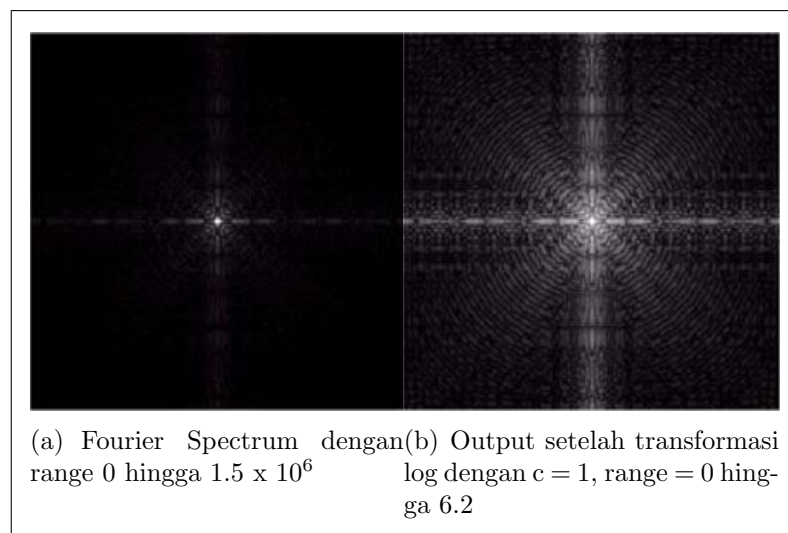
Menggunakan fungsi transformasi $s = c \cdot \log(1 + r)$, dimana c adalah konstanta. Transformasi **Log** memetakan suatu citra dengan range warna sempit menjadi lebih lebar pada citra outputnya. Digunakan untuk meningkatkan nilai pixel yang gelap dalam suatu citra ketika proses kompresi nilai yang lebih tinggi.

Transformasi log dapat memapatakan range citra yang dinamis dengan nilai warna pixel yang sangat bervariasi. Contoh suatu citra dengan range warna yang dinamis adalah spektrum warna hasil transformasi Fourier yang dapat memiliki range intensitas dari 0 hingga 10^6 atau lebih tinggi lagi. Kita tidak dapat melihat derajat ketelitian yang signifikan pada citra dengan



Gambar 5.7: Transformasi Fungsi Negatif

range image yang tinggi, sehingga perlu diturunkan sehingga dapat dilihat (seperti pada gambar 5.8).

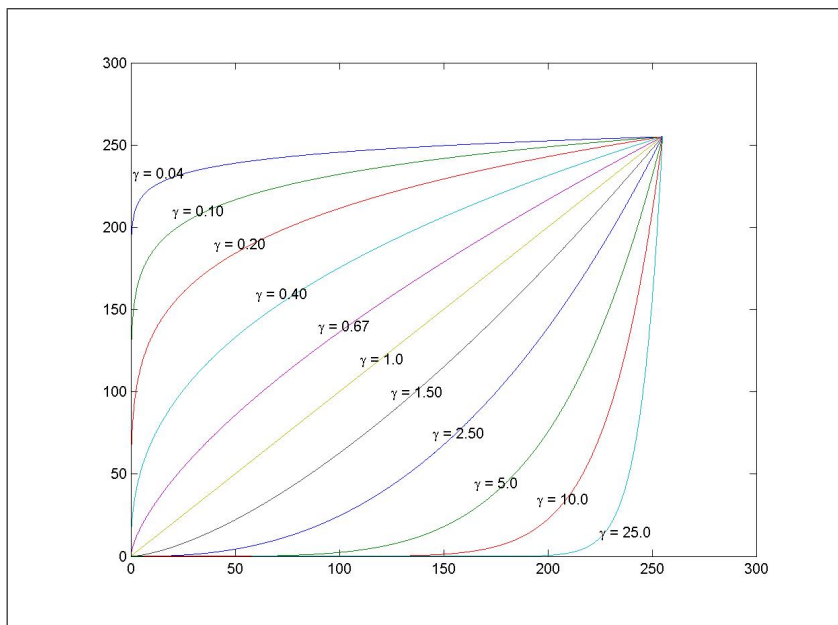


Gambar 5.8: Transformasi Fungsi Logaritmic

5.1. DOMAIN SPATIAL

Perpangkatan

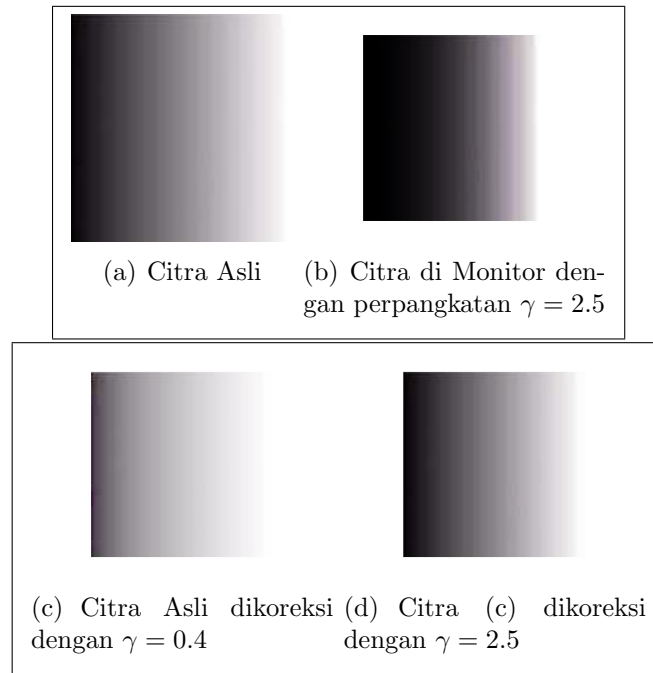
Memiliki fungsi transformasi dengan bentuk $s = cr^\gamma$, dimana c dan γ adalah konstanta positif. Transformasi perpangkatan dengan nilai fractional γ memetakan suatu range citra gelap yang sempit ke range yang lebih lebar, atau sebaliknya untuk nilai input dengan tingkat kecerahan yang tinggi. Gambar 5.9 adalah bentuk fungsi perpangkatan dengan beberapa nilai gamma.



Gambar 5.9: Plot fungsi $c \cdot r^\gamma$

Sebagai contoh pada proses menampilkan citra di monitor CRT (Cathode Ray Monitor) dengan ranah fungsi perpangkatan 1.5 hingga 2.5. Gambar 5.10(a) adalah citra asli, dan gambar 5.10(b) adalah citra setelah koreksi dengan nilai $\gamma = 2.5$ yang mengakibatkan gambar menjadi lebih gelap. Gambar 5.10(c) adalah koreksi gamma gambar 5.10(a) dengan nilai $\gamma = 0.4$ dan gambar 5.10(d) adalah citra pada gambar 5.10(c) dikoreksi dengan fungsi pangkat dengan nilai $\gamma = 2.5$.

Jika nilai γ dikurangi dalam jumlah besar, citra akan mengurangi kontras citra.



Gambar 5.10: Penampilan citra di monitor

5.1.4 Histogram

Dark image Bright image Components of histogram are concentrated on the low side of the gray scale. Components of histogram are concentrated on the high side of the gray scale.

Histogram suatu citra digital dengan suatu tingkat keabuan $[0, L - 1]$ adalah suatu fungsi dikrit:

$$h(r_k) = n_k \quad (5.1)$$

dimana r_k = tingkat keabuan ke-k
 n_k = jumlah total pixel dengan tingkat keabuan r_k pada citra
 $h(r_k)$ = histogram citra digital dengan tingkat keabuan r_k

Gambar 5.11 menunjukkan bentuk histogram citra digital. Citra gelap (gambar a) dimana komponen citra terkonsentrasi pada sisi sebelah kiri dari skala tingkat keabuan, sedangkan pada gambar (b) dengan tingkat kecerahan yang tinggi memberikan bentuk histogram dengan konsentrasi di sebelah

kanan skala. Gambar (c) adalah citra dengan kontras rendah membentuk histogram sempit di tengah skala, sedangkan untuk citra dengan kontras tinggi (gambar c) memberikan citra yang lebar sesuai garis vertikal yang lebih tinggi dari yang lainnya.

5.1.5 Penggeseran dan Pelebaran Histogram

Proses perbaikan kontras suatu citra dapat dilakukan dengan teknik penggeseran histogram dan atau pelebaran histogram. Hal ini dilakukan dengan memetakan nilai intensitas setiap pixel menjadi suatu nilai intensitas yhang menurut rumus/nilai tertentu. Misalakan ketika akan meningkatkan tingkat intensitas suatu gambar, maka dapat ditambahkan suatu faktor/nilai tertentu sehingga warna akan bergeser ke arah yang lebih terang (ke kanan) dan akan lebih gelap (ke kiri). Gambar 5.12 menjelaskan proses perubahan citra melalui penggeseran histogram. Pada gambar (b), adalah pergeseran histogram dengan menambahkan tingkat keabuan gambar (a) sebanyak 130 (kontras meningkat), sedangkan gambar (d) adalah pergeseran histogram dengan mengurangi angka

Pelebaran Histogram dilakukan dengan mengalikan citra asli dengan suatu bilangan. Pada gambar 5.13 menunjukkan pelebaran histogram. Gambar (c) adalah mengalikan tingkat keabuan citra (a) dengan 2 dan gambar (e) mengalikan tingkat keabuan dengan 0.5. Pada gambar (d) terlihat bahwa histogram menjadi lebih lebar dibandingkan gambar (b), sedangkan gambar (f) menunjukkan histogram lebih sempit dibandingkan gambar (a).

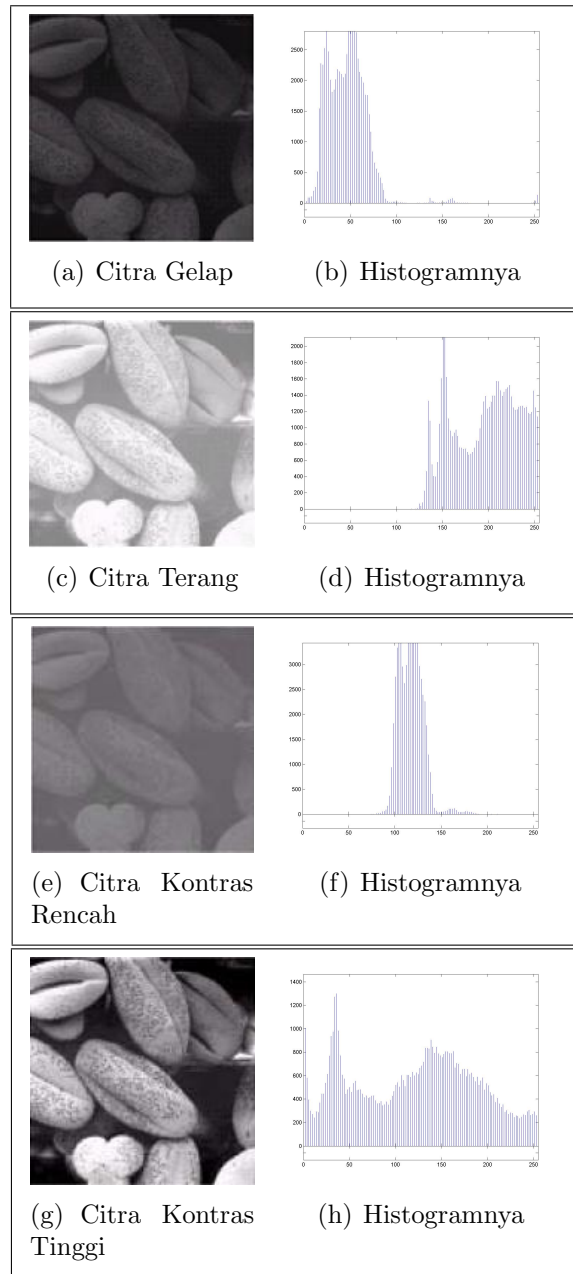
5.1.6 Perataan Histogram (Histogram Equalization)

Teknik perataan histogram merupakan gabuang anantara penggeseran dan pelebaran histogram. Tujuan yang akan dicapai pada teknik ini adalah untuk mendapatkan citra dengan daerah tingkat keabuan yang penuh dan dengan distribusi pixel pada setiap tingkat keabuan yang merata.

Pada teknik perataan histogram ini mentransformasi tingkat keabuan r_k menjadi s_k dengan suatu fungsi transformasi $T(r_k)$ (gambar 5.14). Fungsi transformasi ini memiliki syarat sebagai berikut:

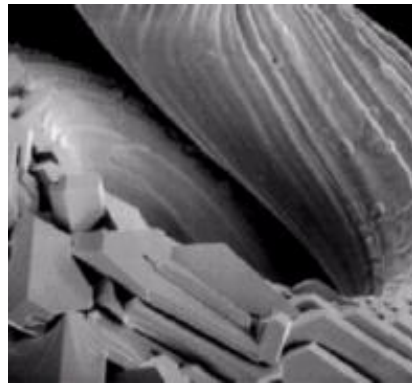
- $T(r)$ memberikan nilai tunggal (one-one-onto) sehingga memiliki inverse serta monoton naik untuk interval $0 \leq r \leq 1$.
- $0 \leq T(r) \leq 1$ untuk $0 \leq r \leq 1$.

BAB 5. PENINGKATAN MUTU CITRA

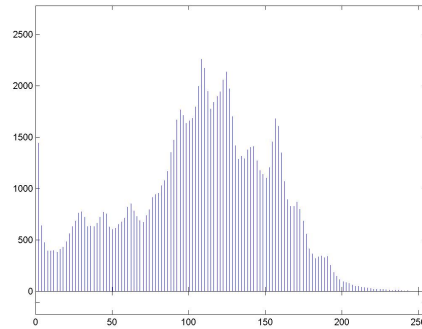


Gambar 5.11: Histogram

5.1. DOMAIN SPATIAL



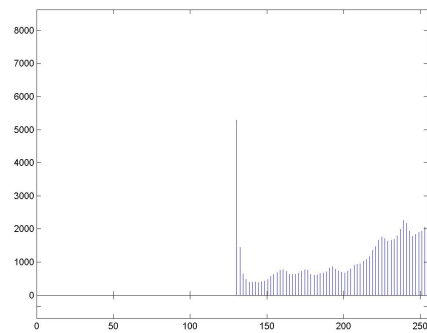
(a) Citra Asli



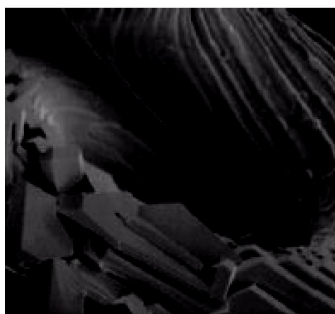
(b) Histogram Citra Asli



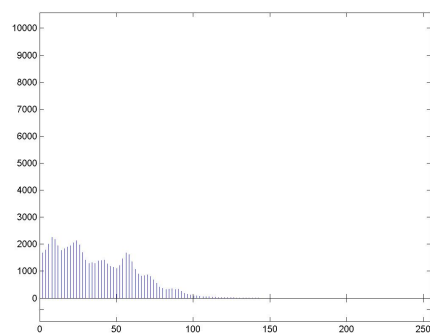
(c) Citra Asli ditambah tingkat keabuan
130



(d) Histogramnya



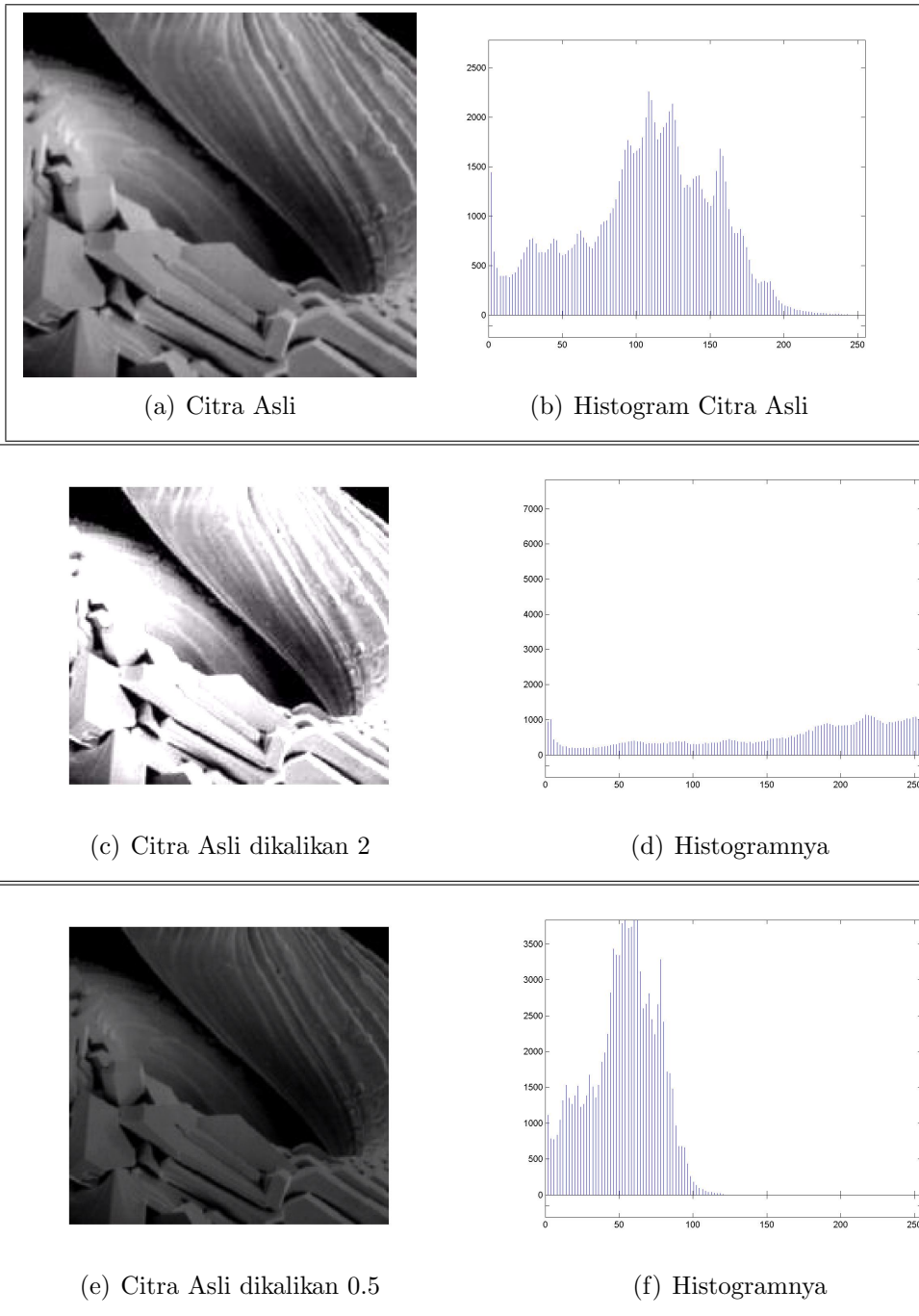
(e) Citra Asli dikurangi tingkat keabuan
100



(f) Histogramnya

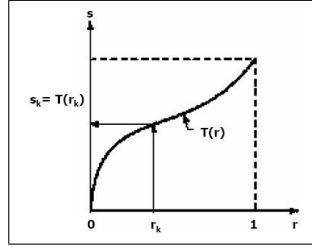
Gambar 5.12: Penggeseran Histogram

BAB 5. PENINGKATAN MUTU CITRA



Gambar 5.13: Pelebaran Histogram

5.1. DOMAIN SPATIAL



Gambar 5.14: Bentuk fungsi $T(r_k)$

- Memiliki inverse $T^{-1}(s_k) = r_k$ untuk $0 \leq s \leq 1$ (kembali ke nilai r_k semula)

Untuk menghitung fungsi $T(r_k)$, perlu dilakukan normalisasi nilai histogram kedalam ranah $[0.,1.]$, yang dinyatakan sebagai $r_k = n_k/n$, dimana n_k adalah jumlah total pixel dengan tingkat keabuan ke -k, dan n jumlah total pixel. Kemudian dihitung nilai probabilitas $p_r = \frac{n_k}{n}$ dimana $0 \leq k \leq L - 1$. Sehingga fungsi transformasinya adalah:

$$\begin{aligned} s_k &= T(r_k) = \sum_{j=0}^k p_r(r_j) \\ &= \sum_{j=0}^k \frac{n_j}{n} \text{ dimana } 0 \leq k \leq L - 1 \end{aligned}$$

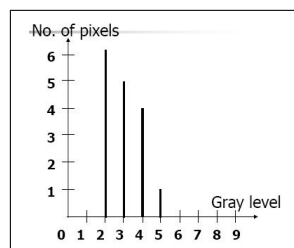
Sebagai ilustrasi, diberikan suatu input citra digital M(4x4) dengan tingkat keabuan 0 hingga 9 sebagai berikut:

$$\begin{bmatrix} 2 & 3 & 3 & 2 \\ 4 & 2 & 4 & 3 \\ 3 & 2 & 3 & 5 \\ 2 & 4 & 2 & 4 \end{bmatrix}$$

dengan bentuk histogram seperti gambar 5.15
maka:

(j)	0	1	2	3	4	5	6	7	8	9
n_j	0	0	6	5	4	1	0	0	0	0
$\sum_{j=0}^{k-1} n_j$	0	0	6	11	15	16	16	16	16	16
$s = \sum_{j=0}^{k-1} n_j/n$	0	0	6/16	11/16	15/16	1	1	1	1	1
$s \cdot 9$	0	0	3	6	8	9	9	9	9	9

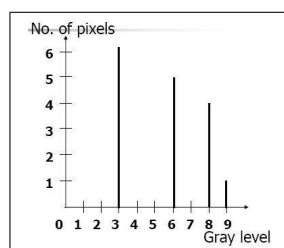
Dari tabel di atas, maka bentuk citra digital hasil perataan histogram citra digital M di atas adalah sebagai berikut:



Gambar 5.15: Histogram Citra Dijital M

$$\begin{bmatrix} 3 & 6 & 6 & 3 \\ 8 & 3 & 8 & 6 \\ 6 & 3 & 6 & 9 \\ 3 & 8 & 3 & 8 \end{bmatrix}$$

Bentuk histogram citra tersebut adalah seperti gambar 5.16:



Gambar 5.16: Histogram Setelah Proses Perataan dari Citra digital M

5.1.7 Mean Filter

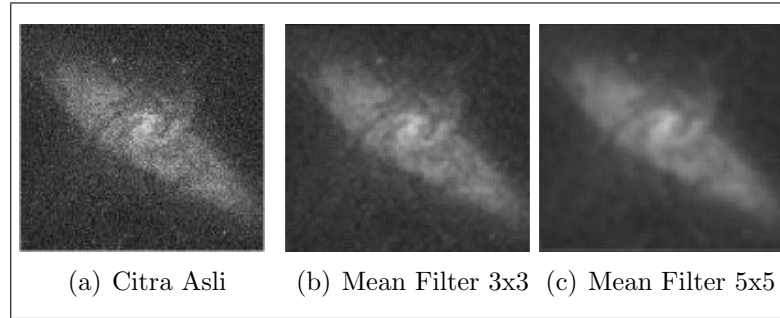
Mean filter digunakan untuk penghalusan (*smoothing*) citra yang memiliki gangguan (*noise*). Mean Filter adalah mengganti nilai pixel pada posisi (x,y) dengan nilai rata-rata pixel yang berada tetangga disekitarnya. Luasan jumlah pixel tetangga ditentukan sebagai masking/kernel/window yang berukuran misalkan 2x2, 3x3, 4x4, dan seterusnya. Penggantian nilai pixel tersebut adalah sebagai berikut:

$$s(x, y) = \frac{1}{MN} \sum_{i=-a}^a \sum_{j=-b}^b p(x + i, y + j)$$

dimana $a = (M - 1)/2$ dan $b = (N - 1)/2$.

5.1. DOMAIN SPATIAL

Gambar 5.17 adalah contoh proses filter dengan menggunakan teknik mean filter. Gambar (a) adalah citra yang diberi noise, gambar (b) adalah mean filter citra (a) dengan ukuran kernel 3x3, sedangkan gambar (c) adalah mean filter citra (a) dengan ukuran kernel 5x5.



Gambar 5.17: Mean Filter

Berikut adalah contoh matriks kernel/masking yang berukuran 3x3 sebagai berikut:

$$\frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sebagai contoh, diberikan suatu citra M (5x5) sebagai berikut:

$$\begin{bmatrix} 4 & 5 & 8 & 5 & 8 \\ 7 & 3 & 5 & 6 & 9 \\ 2 & 2 & 9 & 4 & 8 \\ 6 & 5 & 5 & 6 & 9 \\ 5 & 3 & 8 & 9 & 8 \end{bmatrix}$$

Kemudian akan dilakukan mean filter untuk citra M dengan menggunakan matriks kernel (3x3). Pixel $m(2,2) = 3$, akan diubah menjadi:

Selain mean filtering yang merupakan proses filter linier, terdapat pula pendekatan filter pembobotan (*weighted filter*). Salah satu bentuk matriks kernel filter pembobotan adalah sebagai berikut:

$$\frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\begin{aligned}
 m(2,2) &= 1/9 * (m(1,1) + m(1,2) + m(1,3) + m(2,1) + m(2,2) + m(2,3) + \\
 &\quad m(3,1) + m(3,2) + m(3,3)) \\
 &= 1/9 * (4 + 5 + 8 + 7 + 3 + 5 + 2 + 2 + 9) \\
 &= 1/9 * (45) \\
 &= 5
 \end{aligned}$$

Bentuk umum perubahan nilai pixel (x,y) dengan weighted filter adalah:

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s,t) f(x+s, y+t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s,t)}$$

Perintah Matlab untuk menghasilkan matriks kernel mean filter adalah: `fspecial('average',3)`, sehingga untuk mendapatkan citra yang dihasilkan dari mean filter adalah: `filter2(fspecial('average',3),image)`.

5.1.8 Order-Statistics Filters (Nonlinear Filters)

Filter secara statistik merupakan suatu model filter non linear, karena tidak menggunakan model kombinasi linier dari piksel tetangga. Respon berdasarkan pada urutan (ranking) pixel pada area citra yang ditentukan oleh model filter. Contoh filter non-linear adalah:

- Median Filter. $R = \text{median} \{ z_k | k = 1, 2, \dots, nxn \}$
- Max Filter. $R = \max \{ z_k | k = 1, 2, \dots, nxn \}$
- Min Filter. $R = \min \{ z_k | k = 1, 2, \dots, nxn \}$

dimana $n \times n$ adalah ukuran matriks mask.

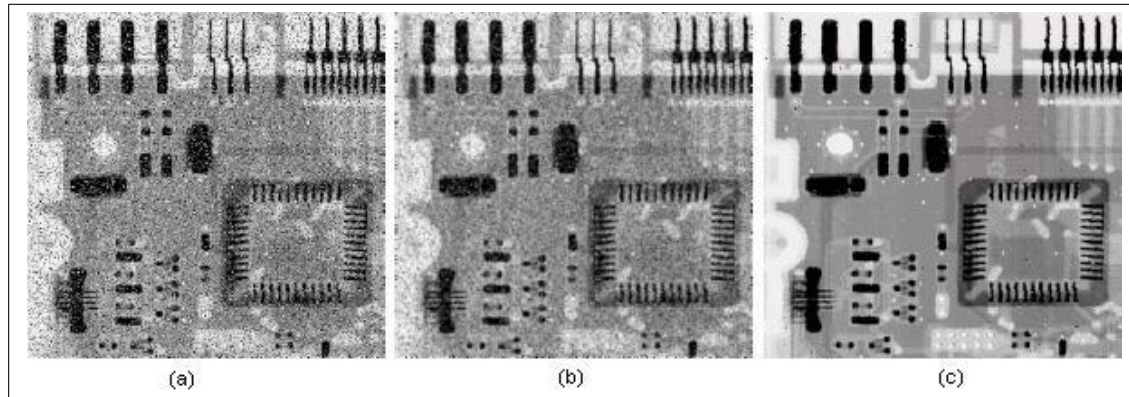
Median filter mengganti nilai suatu piksel dengan median nilai tingkat keabuan dari piksel tetangga (nilai asli piksel digunakan juga pada saat perhitungan nilai median tersebut). Media filter ini cukup populer karena beberapa tipe gangguan acak (seperti *salt noise*, *pepper noise*. Teknik ini mampu mengurangi gangguan yang lebih baik dibandingkan dengan model linear smooting dengan ukuran yang sama.

Median filter mengubah suatu titik dengan tingkat keabuan yang berbeda menjadi lebih mirip dengan tetangganya. Selain itu juga median filter mengganti nilai cluster piksel terisolasi, yang lebih terang atau gelap dibandingkan dengan piksel tetangganya serta luasannya kurang dari $n^2/2$, dengan

5.1. DOMAIN SPATIAL

nilai median dari masking $n \times n$. Sehingga dapat dikatakan bahwa noise yang dihilangkan akan memiliki nilai sama dengan intensitas median tetangganya.

Gambar 5.18 adalah contoh perbandingan hasil antara mean filter dan median filter dengan menggunakan ukuran window yang sama (3×3).



Gambar 5.18: (a) Citra yang diberi noise, (b) Mean Filter 3×3 (c) Median Filter 3×3

Sharpening Spatial Filter

Digunakan untuk mendapatkan detail suatu citra atau meningkatkan detail citra yang sudah di-blurkan karena kesalahan ataupun dampak metode akuisisi. Seperti yang sudah diketahui bahwa bluring dapat dilakukan dalam domain spasial dengan menggunakan mean filter, dimana analogis dengan operasi integrasi. Sehingga untuk proses penajaman ini dianalogikan dengan operasi differential.

Kekuatan respon operator derivatif adalah proportional dengan derajat diskontinu suatu citra pada suatu titik dimana operasi diterapkan. Sehingga, citra diferensiasi adalah berkaitan dengan meningkatkan sisi dan diskontinu lainnya (noise) serta penurunan area dengan nilai variasi tingkat keabuan yang rendah.

Laplacian merupakan sharpening spatial filter dengan matriks marking sebagai berikut:

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

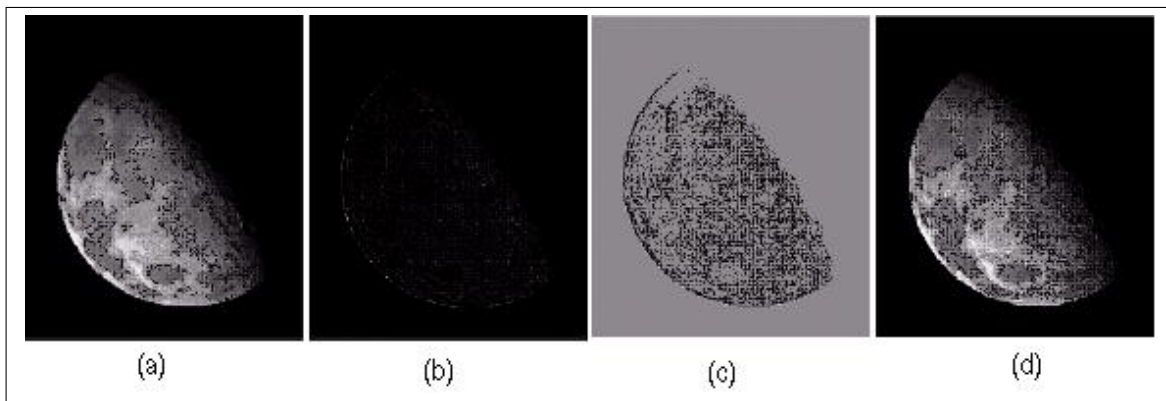
BAB 5. PENINGKATAN MUTU CITRA

Sedangkan bentuk laplacian mask untuk perluasan terhadap diagonal neighbors adalah sebagai berikut:

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

atau bentuk implementasi lainnya yang menghasilkan citra yang sama, yaitu:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \text{ dan } \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



Gambar 5.19: Transformasi Laplacian untuk penajaman detail pada gambar permukaan bulan bagian kutub utara

Gambar 5.19 menunjukkan proses transformasi Laplacian digunakan untuk memperjelas detail. Gambar (a) adalah citra asli, sedangkan gambar (b) setelah dilakukan filter laplacian, (c) skala dari hasil filter laplacian untuk memperjelas tampilan dan (d) adalah penjumlahan antara gambar asli (a) ditambah dengan gambar (b). Sehingga untuk menyederhanakan proses komputasi, maka penjumlahan hasil filter laplacian dengan citra asli dapat digunakan matriks masking sebagai berikut:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

5.2. FILTER DOMAIN FREKUENSI

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Dampak operator Laplace

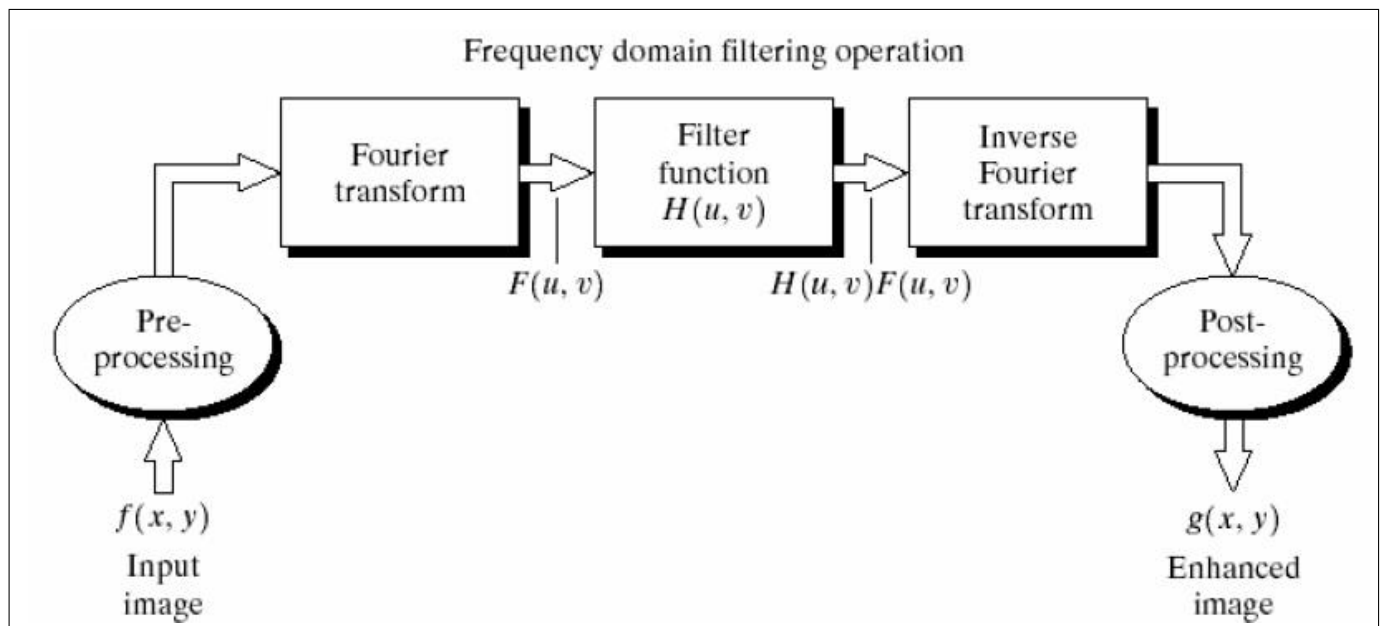
Dampak operator laplace dapat ditinjau dari sisi:

- Dampak operator derivative akan menjelaskan diskontinu tingkat keabuan pada suatu citra sehingga mengurangi perubahan pada suatu daerah dengan tingkat keabuan yang sedikit bervariasi.
- Cenderung menghasilkan citra yang garis sisi yang agak keabuan dan dikontinuitas lainnya serta superimposed pada suatu kegelapan. Selain itu juga menghasilkan latar belakang yang tidak memiliki fitur.

5.2 Filter Domain Frekuensi

Setelah pembahasan tentang filter domain spatial, filter domain frekuensi adalah melakukan peningkatan mutu citra melalui spectrum citra. Gambar 5.20 memperlihatkan alur proses filter dengan domain frekuensi:

1. mengalikan citra input dengan -1^{x+y} untuk menggeser transformasi ke tengah yaitu $u = M/2$ dan $v = N/2$ (jika M dan N adalah bilangan bulat, maka koordinat yang digeser akan menjadi integer).
2. Hitung transformasi diskrit Fourier $F(u, v)$ dari citra diatas.
3. Kalikan $F(u, v)$ dengan fungsi filter $H(u, v)$
4. Hitung inverse DFT dari hasil (3).
5. Ambil bagian real dari hasil (4).
6. Kalikan hasil (5) dengan $-1^{(x+y)}$ untuk mengembalikan ke bentuk seperti citra input.



Gambar 5.20: Langkah Proses Filter Domain Frekuensi

Bab 6

Segmentasi Citra

Segmentasi adalah membagi suatu citra ke dalam beberapa daerah berdasarkan kesesuaian bentuk/objek. Proses segmentasi akan selesai apabila objek yang diperhatikan dalam aplikasi sudah terisolasi.

Algoritma segmentasi secara umum berbasiskan pada salah satu dari dua sifat dasar nilai intensitas:

1. diskontinu: membagi suatu citra berdasarkan perubahan besar nilai intensitas (seperti sisi)
2. similaritas: membagi suatu citra berdasarkan similaritas sesuai kriteria tertentu yang sudah didefinisikan.

6.1 Deteksi Diskontinuitas

detect the three basic types of graylevel discontinuities .. points , lines , edges
.. the common way is to run a mask through the image

Mendeteksi tiga tipe dasar diskontinuitas tingkat keabuan, yaitu: titik, garis dan sisi. Teknik umum yang digunakan adalah dengan menggunakan mask pada keseluruhan image.

6.1.1 Deteksi Titik

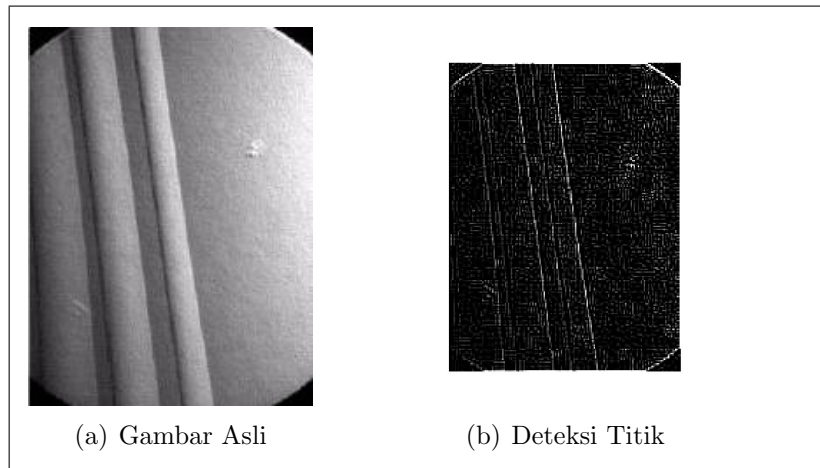
Penggunaan marking untuk melakukan deteksi titik, misalkan penggunaan masking laplace sebagai berikut

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

Suatu titik dideteksi pada lokasi dimana mark terpusat apabila:

$$|R| \geq T$$

dimana, T adalah threshold non-negatif, R adalah jumlah perkalian koefisien dengan tingkat keabuan yang ada pada daerah yang sesuai dengan mark. Gambar 6.1 adalah contoh deteksi titik dengan menggunakan matrik mask laplace seperti di atas.



Gambar 6.1: Deteksi Titik pada Citra

6.1.2 Deteksi Garis

Proses deteksi garis dilakukan melalui beberapa sisi, seperti horisontal, vertikal, dan sudut $(+/-)45^0$

$$\begin{array}{cccc} \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} & \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} & \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} & \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix} \\ \text{Horizontal} & +45^0 & \text{Vertical} & -45^0 \end{array}$$

6.1. DETEKSI DISKONTINUITAS

Mask horizontal akan menghasilkan respon yang maksimal apabila suatu garis berada pada baris tengah mask dengan latar belakang konstan. Hal ini juga similiar untuk mask yang lainnya. Arah yang diinginkan dari setiap mask dibobotkan dengan suatu koefisien yang lebih besar (yaitu 2) daripada kemungkinan arah yang lainnya.

Tahap melakukan deteksi garis adalah sebagai berikut:

1. Kalikan setiap masking ke citra
2. Misalkan R_1, R_2, R_3, R_4 masing-masing menyatakan response terhadap masking horizontal, $+45^\circ$ vertical, dan -45° .
3. Jika pada suatu titik di citra berlaku $|R_i| > |R_j|$ untuk semua $j \neq i$, maka titik tersebut dikatakan lebih cenderung berhubungan dengan suatu garis pada arah mask i .

Sebagai alternatif, jika kita akan mendeteksi semua garis pada suatu citra yang sesuai dengan arah mask yang diberikan dengan menjalankan mask tersebut ke keseluruhan citra dan threshold hasil sesuai dengan nilai absolut tertentu. Titik-titik yang tersisa memiliki respon yang terkuat, yaitu, untuk garis-garis pada pixel tebal, berhubungan dekat dengan arah yang didefinisikan oleh mask.

6.1.3 Deteksi Sisi (*Edge Detection*)

Operasi pendeteksi diskontinuitas yang paling penting adalah deteksi sisi. Sisi didefinisikan sebagai batas antara dua daerah yang memiliki perbedaan tingkat keabuan yang signifikan, sehingga dapat ditentukan berdasarkan nilai turunan pertama. Hal ini dilakukan dengan menghitung nilai gradien suatu piksel relatif terhadap piksel disekitarnya. Contoh aproksimasi turunan pertama yang baik adalah operator Sobel dengan keuntungan efek smoothing. Karena dampak derivasi adalah meningkatkan noise, efek smoothing merupakan suatu fitur operator Sobel yang menguntungkan untuk mengurangi noise tersebut.

Bentuk operator sobel adalah sebagai berikut:

$$\begin{array}{cc} \text{Horizontal} & \text{Vertical} \\ \left[\begin{array}{ccc} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{array} \right] & \left[\begin{array}{ccc} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{array} \right] \end{array}$$

Nilai gradien dihitung berdasarkan dua arah: horizontal dan vertical, yaitu:

$$\begin{aligned} G_x &= \frac{\partial f}{\partial x} = (x_7 + 2x_8 + x_9) - (x_1 + 2x_2 + x_3) \\ G_y &= \frac{\partial f}{\partial y} = (x_3 + 2x_6 + x_9) - (x_1 + 2x_4 + x_7) \end{aligned}$$

Aproksimasi nilai gradien yang umum adalah menggunakan penjumlahan nilai absolute kedua gradien tersebut, yaitu:

$$\nabla f \approx |G_x| + |G_y|$$

dan arah dari gradien tersebut adalah:

$$\alpha = \tan^{-1} \frac{G_x}{G_y}$$

Derivatif kedua dapat didekatkan dengan menggunakan making Laplacian. Kekurangan dari Laplacian adalah sensitif terhadap noise dan tidak mampu mendeteksi arah sisi. Di lain hal, karena merupakan derivatif ke dua, laplacian menghasilkan dua puncak (impuls positif dan negatif). Dengan Laplacian, kita dapat mendeteksi apakah pixel berada pada bagian gelap atau terang suatu sisi (edge). Sifat ini dapat digunakan dalam segmentasi citra.

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & -4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Selain itu terdapat pula operator lain untuk deteksi sisi, yaitu: prewit dan robert. Bentuk operator prewit adalah sebagai berikut:

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

Sedangkan operator Robert adalah sebagai berikut:

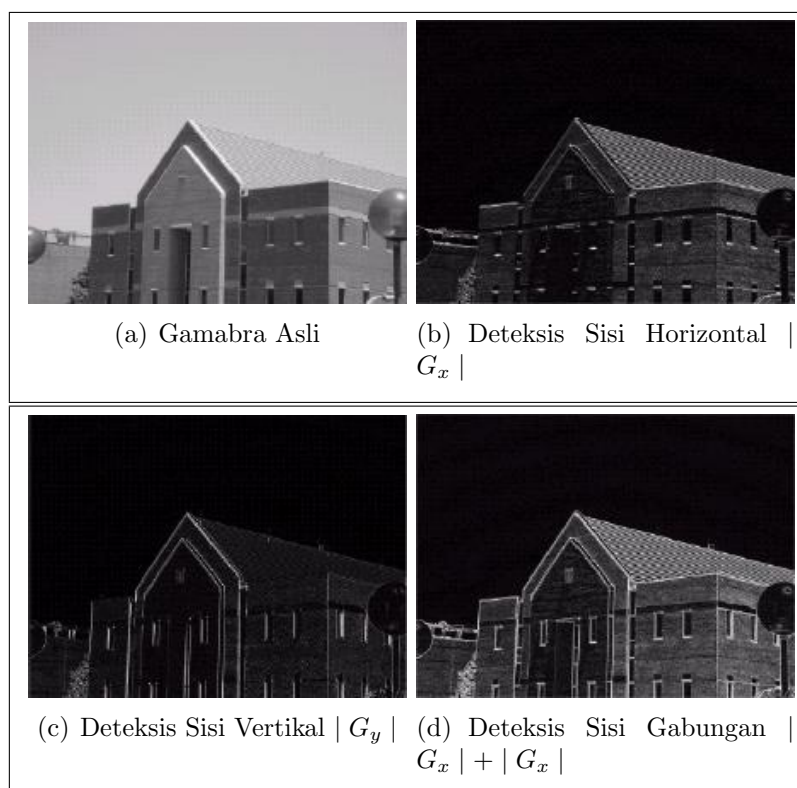
Horizontal	Vertical
$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

6.1. DETEKSI DISKONTINUITAS

Selain model deteksi sisi horizontal dan vertical, terdapat pula deteksi sisi untuk model diagonal. Berikut ini adalah operator masking Prewit dan Sobel.

$$\begin{array}{c} \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \\ \text{Prewit} \\ \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} \\ \text{Sobel} \end{array}$$

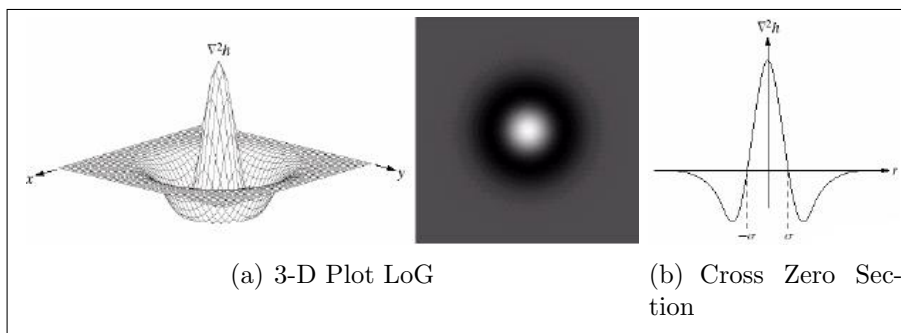
Gambar 6.2 adalah contoh pembentukan deteksi sisi horizontal dan vertical.



Gambar 6.2: Deteksi Sisi Horizontal Vertikal



Gambar 6.3: Deteksi Sisi secara Diagonal



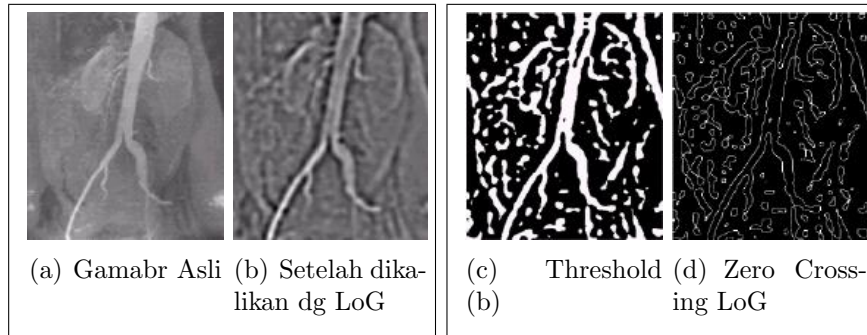
Gambar 6.4: Laplacian of Gaussian (LoG)

Sedangkan gambar 6.3 menunjukkan masing-masing deteksi sisi secara diagonal.

Operator lainnya adalah Laplacian of Gaussian. Laplacian of Gaussian (LoG) membentuk suatu cross-zero section, yaitu diagram yang memotong sumbu v yang menunjukkan suatu perbedaan antara daerah terang dan gelap. Gambar 6.4 menunjukkan bentuk plot 3-D dari LoG tersebut.

Bentuk operator matriks LoG adalah sebagai berikut:

6.2. SEGMENTATION



Gambar 6.5: Proses Citra Zero Crossing dari Laplacian of Gaussin (LoG)

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

Jumlah keseluruhan koefisien dari matriks di atas adalah nol. Bentuk operator LoG untuk derivatif ke dua merupakan sautu operator linear, dimana pertama adalah menggunakan Gaussian untuk melakukan penghalusan (smoothing) dan kemudian menggunakan Laplacian untuk memperjelas sisi.

Zero crossing dan Log merupakan aproksimasi zero crossing pada citra yang sudah dikalikan dengan operator LoG, yaitu melakukan threshold dengan mengubah semua nilai positif sebagai warna putih dan semua nilai negatif sebagai warna hitam. Zero crossing muncul antara nilai positif dan negatif dari LoG yang sudah di-threshold. Gambar 6.5 menunjukkan proses model zero cross section dari operator LoG tersebut. Keuntungan zero crossing adalah menghasilkan deteksi sisi yang lebih tipis dan sudah menghilangkan noise. Kerugiannya adalah menghasilkan closed-loop (efek spaghetti) sehingga mengakibatkan komputasi yang lama. Sehingga gradient lebih sering digunakan.

6.2 Segmentation

Segmentasi sering digunakan sebagai fase pertama dalam analisis citra. Tujuan utamanya adalah membagi citra ke dalam basis elemen sesuai dengan

kriteria yang ditentukan. Bentuk elemen bergantung pada aplikasi. Misalkan citra yang terdiri dari udara hingga daratan, cukup memisahkan antara jalan dari lingkungan dan kendaraan yang bergerak di atas jalan. Segmentasi citra otomatis merupakan model yang sangat sulit dalam pengolahan citra.

Algoritma segmentasi untuk citra monochrome (citra dengan tingkat keabuan) secara umum berbasiskan pada satu atau dua sifat dasar nilai tingkat keabuan: diskontinuitas dan similaritas. Operasi pengolahan citra dasar untuk mendeteksi diskontinuitas adalah seperti penjelasa di bagain sebelumnya (sperti deteksi titik, garis dan sisi). Algoritma segmentasi similaritas dapat berbasiskan nilai threshold, region grow(perkembangan daerah), atau pemisahan dan penggabungan daerah. Secara formal segmentasi citra dapat didefinisikan sebagai proses membagi suatu citra R ke dalam beberapa daerah R_1, R_2, \dots, R_n sedemikian sehingga:

- Setiap segmentasi adalah lengkap : $\bigcup_{i=1}^n R_i = R$
- Setiap daerah adalah Uniform
- Setiap daerah tidak saling beririsan : $R_i \cap R_j = \emptyset, \forall i \neq j$
- Piksel pada suatu daerah memiliki sifat yang sama
- Daerah yang saling bertetangga memiliki sifat yang bebrbeda.

6.2.1 Segmentasi Regular Block

The easiest way to segment an image is to divide it into blocks with regular shapes. The shape is often fixed throughout the image and the segmentation is based on varying the size of the blocks. These techniques are often found in image compression methods that do not require a perfect segmentation of the image, rather than aims at minimizing the contrast (variance) within the blocks.

Quadtree segmentation

Quadtree segmentation begins with only one segment which is the entire image. Let the size of the image be $n \times n$ pixels. The image is then recursively split into four equally sized sub-blocks of size $\frac{n}{2} \times \frac{n}{2}$ until the blocks meet a predefined uniformity criterion. This criterion is generally the variance of the pixels in the block. The block is considered homogenous if its variance is below a predefined threshold ϵ , otherwise it is divided into sub blocks.

6.2. SEGMENTATION

See Figure 3.1 for an example of a quadtree, and Figure 3.2 for a quadtree segmentation applied to the image Camera.

Cara segmentasi citra termudah adalah membagi citra ke beberapa blok dengan bentuk yang regular. Bentuk sering ditetapkan pada keseluruhan citra dan segmentasi dilakukan dengan membedakan ukuran blok. Teknik ini sering ditemukan pada metode kompresi citra yang tidak memerlukan suatu segmentasi citra yang sempurna, selain bertujuan meminimumkan variansi dalam suatu blok.

Segmentasi Quadtree

Segmentasi Quadtree berawal hanya pada satu segmen yaitu keseluruhan citra. Misalkan ukuran citra $n \times n$ piksel. Citra kemudian secara rekursif dibagi kedalam empat daerah yang memiliki ukuran yang sama $\frac{n}{2} \times \frac{n}{2}$ hingga blok sesuai dengan kriteria uniform yang sudah didefinisikan. Kriteria tersebut secara umum merupakan variansi piksel dalam blok. Blok dianggap homogen jika variansi kurang dari nilai threshold (ambang) e tertentu, jika tidak maka blok dibagi kembali ke dalam sub-blok. Gambar 6.6 adalah contoh suatu quadtree, dan gambar 6.7 adalah implementasi pada citra.

6.2.2 Segmentasi Binary tree

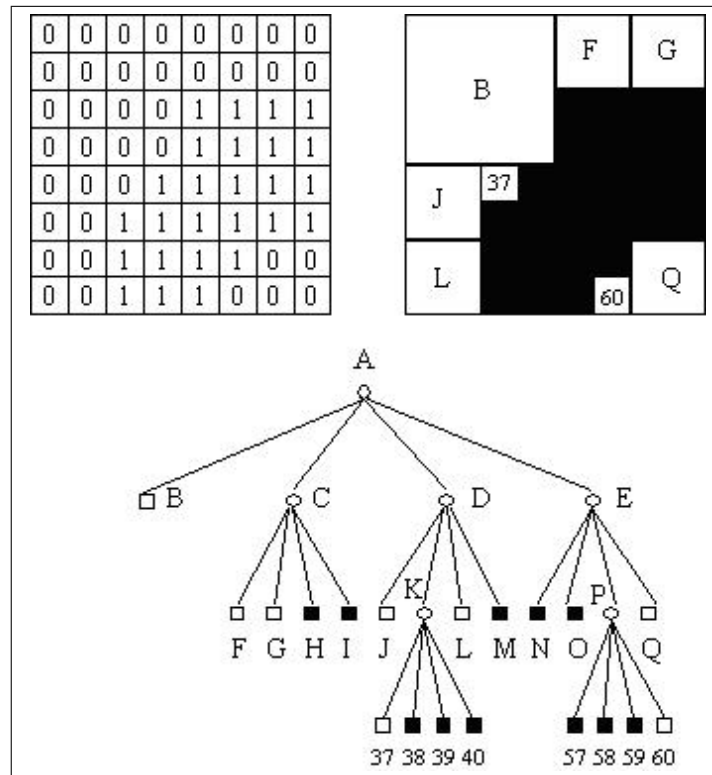
Segmentasi Binary-tree similar dengan segmentasi quadtree. Citra tidak dibagi ke dalam empat daerah melainkan ke dalam dua daerah yang memiliki ukuran yang sama. Arah pembagian bervariasi sehingga blok pada level pertama (=keseluruhan citra) dibagi secara horizontal). Pada level ke dua pembagian dilakukan dengan arah vertikal, dan pada level ke tiga dilakukan secara horizontal kembali dan seterusnya. Dengan cara ini blok akan selalu berbentuk bujur sangkar atau persegi panjang dengan dimensi 2 : 1

Kriteria threshold dapat dilakukan secara variasi seperti pada segmentasi quadtree. Kriteria yang cocok lainnya, dapat digunakan untuk segmentasi quadtree dan binary tree, begitu pula segmentasi polygon.

6.2.3 Polygon segmentation

Generalisasi segmentasi binary tree yang dibagi pada sembarang arah, tidak hanya horizontal dan vertikal. Suatu garis lurus yang digambarkan dari sembarang piksel batas daerah ke piksel batas daerah yang lainnya. Garis

BAB 6. SEGMENTASI CITRA

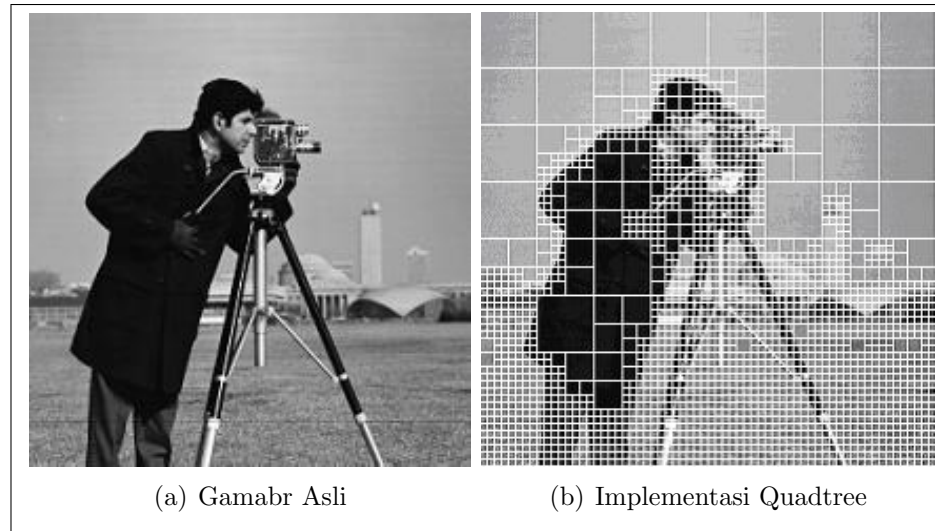


Gambar 6.6: Contoh Struktur Data Quadtree

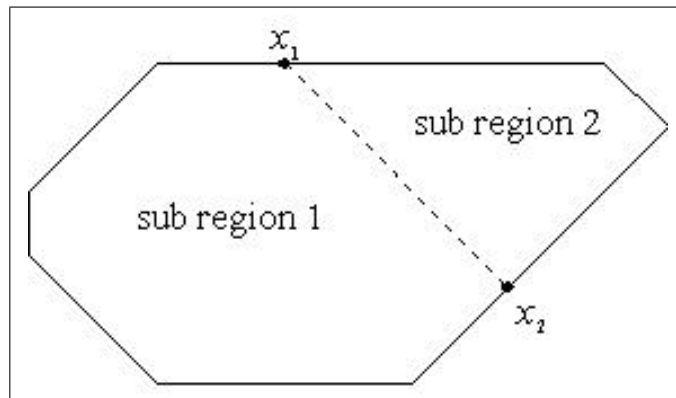
tersebut membagi daerah menjadi dua sub daerah. Hasil daerah tersebut selalu berbentuk polygon, sehingga metoda ini disebut segmentasi polygon. Gambar ... adalah contoh pembagian blok polygon.

Untuk membagi daerah tersebut mejadi

6.2. SEGMENTATION



Gambar 6.7: Contoh Implementasi Segmentasi Quadtree



Gambar 6.8: Contoh Pembagian Blok Polygon