BAB II

LANDASAN TEORI

2.1. Citra Dijital

Citra dijital (*Digital Image*) adalah gambar hasil keluaran suatu sistem perekaman data yang elemennya dinyatakan dengan suatu besaran numerik yang membentuk array dua dimensi. Selanjutnya, untuk menyebut citra dijital akan digunakan istilah gambar atau citra saja.

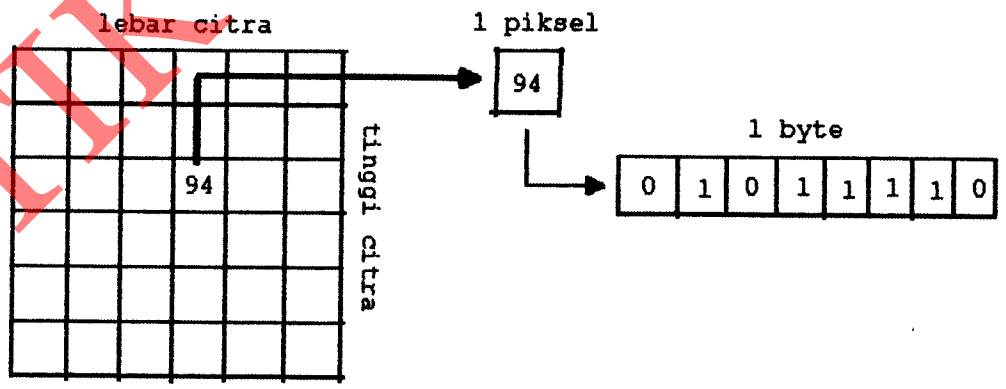
'Gambar apapun yang tampak di layar komputer sebenarnya tersusun dari titik-titik kecil yang disebut *piksel* (*pixel* = *picture element*), yaitu bagian terkecil dari citra. Jika beberapa piksel diletakkan berderet, akan tampaklah suatu garis. Jadi garis sehalus apapun yang tampak di layar komputer, sebenarnya adalah deretan piksel. Kumpulan piksel yang disusun dalam bentuk baris dan kolom akan membentuk suatu citra. Jumlah bit yang menyusun piksel akan menentukan jumlah warna maksimum yang dapat membentuk suatu citra.'

Piksel yang dibentuk dari sebuah bit akan menghasilkan citra Black and White (monochrome). Piksel yang dibentuk dari empat buah bit akan menghasilkan citra dengan jumlah warna maksimum 16 warna . Sedangkan piksel

yang terbentuk dari 8 buah bit akan menghasilkan citra dengan jumlah warna maksimum 256 warna.

Citra Grayscale adalah suatu citra yang mempunyai warna piksel berupa tingkat keabuan yang berbeda untuk membedakan obyek gambar yang satu dengan obyek gambar yang lain maupun dengan backgroundnya. Jumlah tingkat keabuan yang terdapat pada citra tersebut tergantung pada jumlah bit per pikselnya. Umumnya nilai 0 (nol) adalah nilai paling gelap dan $2^n - 1$ adalah nilai paling terang, dimana n adalah suatu nilai integer yang menunjukkan jumlah bit per piksel.

Pada gambar 2.1, satu kotak dianggap mewakili satu piksel. Masing-masing kotak mempunyai nilai yang menunjukkan warna dari piksel yang bersangkutan. Jika satu piksel membutuhkan tempat sebesar satu byte, maka kotak tersebut mempunyai nilai antara 0 sampai dengan 255.



Gambar 2.1. Citra dijital

Bitmap adalah peta bit. *File Bitmap* adalah suatu file (biasanya citra) yang ditampilkan dengan cara dipetakan sebagai satu atau lebih bit dalam memori komputer yang mewakili setiap titik (piksel) dalam tampilan di layar monitor.

Gambar yang terdapat pada buffer adaptor video pada dasarnya merupakan gambar dalam format bitmap. Apabila ingin menampilkan gambar dari disk ke layar cukup dengan meletakkan file yang telah tersimpan di disk tersebut ke buffer, kemudian ditampilkan ke layar. Memory yang dibutuhkan untuk meyimpan gambar satu layar penuh pada adaptor EGA dengan resolusi 640x350 piksel 16 warna dibutuhkan paling tidak ruang 112.000 byte. Nilai tersebut diperoleh dari perhitungan (640x350)/2. Mengapa harus dibagi 2 ? Seperti yang sudah dijelaskan sebelumnya bahwa gambar 16 warna memerlukan 4 bit per piksel, atau dengan kata lain 1 byte ditempati oleh 2 buah piksel.

Adaptor VGA dengan resolusi 640x480 piksel 16 warna dibutuhkan ruang kira-kira 153.600 byte ((640x480)/2). Ruang yang digunakan pasti akan lebih banyak lagi jika kita menggunakan adaptor super VGA dengan resolusi 800x600 piksel atau IBM 8514/A yang dapat menampilkan gambar dengan resolusi 1024x768 piksel 256 warna. Bagi sebagian program gambar, gambar tersebut terlebih dahulu

dimampatkan dengan menggunakan teknik tertentu sebelum disimpan. Cara ini akan menghasilkan file dengan ukuran yang lebih kecil. Ketika membaca file, maka file tersebut harus dikembalikan ke bentuk aslinya supaya dapat ditampilkan pada layar.

Digital Image Processing adalah proses pengolahan citra, dimana baik masukan maupun keluarannya berbentuk berkas citra dijital.

Saat ini banyak sekali bidang pekerjaan yang memanfaatkan keberadaan pengolahan citra, antara lain untuk menentukan *brightness* gambar bintang yang diperoleh dari teleskop, menentukan struktur virus pada gambar hasil mikroskop, menghasilkan ketepatan yang tinggi pada pemetaan bumi dari gambar yang dihasilkan oleh satelit, mendesain pola-pola tekstil sebelum ditenun, dan membantu memperbaiki lukisan-lukisan klasik. Pengolahan citra juga diterapkan pada bidang kedokteran, pembuatan peta, bidang industri, permesinan, penerbitan, bahkan bidang kosmetik.

Pengolahan citra dititik beratkan pada pemrosesan gambar dengan menggunakan komputer atau biasanya gambar yang sudah diubah dalam bentuk numerik. Gambar ini berasal dari beberapa sumber. Suatu foto dapat dibuat dijital dengan menggunakan peralatan yang disebut *Scanner*.

Pada umumnya kegunaan dari digital image processing adalah untuk mempertinggi atau memperbaiki mutu suatu citra/gambar dalam beberapa cara.

Biasanya operasi yang dilakukan adalah membuat gambar yang kabur menjadi lebih terang, menghilangkan bintik-bintik pada gambar, memperbaiki kontras gambar, membagi citra dalam bagian obyek dan background, memperbesar, memperkecil, atau memutar suatu gambar, dan pengkodean citra dalam beberapa cara untuk penyimpanan.

Gambar 2.2 menggambarkan bidang-bidang lain yang menggunakan komputer untuk memroses data gambar atau data deskripsi.

		Output	
		Image	Description
Input	Image	Image Processing	Image Pattern Recognition Computer Vision
	Description	Computer Graphics	Other Data Processing

Gambar 2.2. Pengolahan data berdasarkan input/output

Image Processing mempunyai input berupa gambar dan outputnya juga berupa gambar, dimana output tersebut diperbaiki dengan beberapa cara. *Computer Graphics*

menghasilkan gambar sebagai output, tetapi inputnya adalah informasi deskripsi.

Image Pattern Recognition dan *Computer Vision* merupakan kebalikan dari computer graphics, yaitu mempunyai input berupa gambar dan menghasilkan informasi deskripsi sebagai outputnya.

2.2. Tagged Image File Format

Dalam file TIFF terdapat tiga bagian, yaitu : *Header, Image File Directory (IFD),* dan *Image File Directory Entry (IFD Entry).*

TIFF Header selalu menempati 8 byte pertama dari suatu file TIFF. Apabila kita ingin membaca file TIFF, maka tugas pertama yang harus dilakukan adalah membaca header ini.

TIFF header terdiri dari tiga bagian, yaitu :

Byte Order	2	byte
Versi TIFF	2	byte
Offset IFD Pertama	4	byte

Gambar 2.3. Header file TIFF

Byte Order adalah urutan byte yang digunakan pada saat mengakses file TIFF. Bagian ini menenpati dua byte pertama dari header. Bagian ini mempunyai nilai *II* atau *MM*. II berarti *Intel* atau *little-endian* (hexa : 0x4949) dengan urutan dari *Least Significant Byte* (LSB) ke *Most Significant Byte* (MSB), sedangkan MM berarti *Motorola* atau *big-endian* (hexa : 0x4D4D) dengan urutan dari Most Significant Byte ke Least Significant Byte. Pada umumnya bernilai II (0x4949).

Versi TIFF menyatakan versi dari TIFF yang digunakan untuk membuat citra. Panjangnya 2 byte. Nilainya adalah 42 (hexa : 0x002A atau 0x2A00, tergantung dari nilai Byte Order, apakah II atau MM).

Offset IFD Pertama merupakan 4 byte terakhir dari header yang menyatakan offset (dalam satuan byte) dari awal file ke IFD pertama.

Image File Directory (IFD) merupakan array dari *IFD Entry*. Jika suatu file berisi citra lebih dari satu, maka file tersebut terdiri dari satu IFD per citra, yang berarti mempunyai IFD lebih dari satu. Pada umumnya satu file TIFF mempunyai citra sebanyak satu.

Ada tiga field yang terdapat pada IFD, yaitu :

Jumlah IFD Entry	2 byte
Array IFD Entry	12 byte x Jumlah IFD Entry
Offset IFD Berikutnya	4 byte

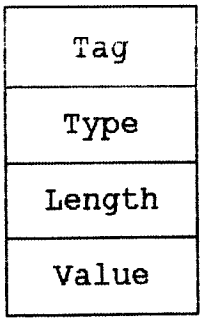
Gambar 2.4. Image File Directory

Jumlah IFD Entry adalah dua byte integer yang menyatakan jumlah IFD Entry yang terdapat dalam IFD tersebut.

Array IFD Entry mempunyai sejumlah IFD Entry, di mana masing-masing IFD Entry sebesar 12 byte. Jadi pada bagian inilah letak IFD Entry. Besarnya adalah jumlah IFD Entry dikalikan dengan 12 byte karena satu IFD Entry sebesar 12 byte.

Offset IFD Berikutnya menyatakan offset ke IFD berikutnya. Besarnya 4 byte. Jika bernilai 0 (nol), berarti tidak ada IFD berikutnya.

Image File Directory Entry (IFD Entry) menyatakan ukuran dan type dari citra serta di mana citra itu disimpan dalam file TIFF. IFD Entry memerlukan tempat sebanyak 12 byte dengan perincian sebagai berikut :



Gambar 2.5. Image File Directory Entry

Tag digunakan untuk mengidentifikasi data citra dalam file TIFF tersebut. Tag ini akan dijelaskan pada bagian berikutnya.

Type menunjukkan type dari masing-masing data tag di atas. Misal : tag 256 dan 257 yang masing-masing menunjukkan lebar dan tinggi citra mempunyai type short.

Length menunjukkan jumlah tipe data yang terdapat dalam suatu tag. Misalnya suatu tag yang bertype long mempunyai nilai Length sama dengan dua, berarti tag tersebut mempunyai tipe data long sebanyak dua.

Value menyatakan informasi mengenai citra yang berada dalam file TIFF tersebut.

Ciri dari file TIFF yaitu adanya *Tag*. Dalam format TIFF, tag menunjukkan struktur dasar dari file dan digunakan untuk mengidentifikasi data citra yang terdapat dalam file TIFF yang bersangkutan. Macam-macam tag yang terdapat dalam format TIFF tampak seperti tabel 2.1.

Tabel 2.1. Tag dalam file TIFF

NAMA TAG	NOMOR TAG		TYPE DATA
	DESIMAL	HEXADESIMAL	
NewSubfileType	254	(FE)	LONG
SubfileType	255	(FF)	SHORT
ImageWidth	256	(100)	SHORT/LONG
ImageLength	257	(101)	SHORT/LONG
BitsPerSample	258	(102)	SHORT
Compression	259	(103)	SHORT
PhotometricInterpretation	262	(106)	SHORT
Thresholding	263	(107)	SHORT
CellWidth	264	(108)	SHORT
CellLength	265	(109)	SHORT
FillOrder	266	(10A)	SHORT
DocumentName	269	(10D)	ASCII
ImageDescription	270	(10E)	ASCII
Make	271	(10F)	ASCII
Model	272	(110)	ASCII
StripOffsets	273	(111)	SHORT/LONG
Orientation	274	(112)	SHORT
SamplesPerPixel	277	(115)	SHORT
RowsPerStrip	278	(116)	SHORT/LONG
StripByteCounts	279	(117)	LONG/SHORT
MinSampleValue	280	(118)	SHORT
MaxSampleValue	281	(119)	SHORT
XResolution	282	(11A)	RATIONAL
YResolution	283	(11B)	RATIONAL
PlanarConfiguration	284	(11C)	SHORT
PageName	285	(11D)	ASCII
XPosition	286	(11E)	RATIONAL
YPosition	287	(11F)	RATIONAL
FreeOffsets	288	(120)	LONG
FreeByteCounts	289	(121)	LONG
GrayResonseUnit	290	(122)	SHORT
GrayResponseCurve	291	(123)	SHORT
Group3Options	292	(124)	LONG
Group4Options	293	(125)	LONG
ResolutionUnit	296	(128)	SHORT
PageNumber	297	(129)	SHORT
ColorResponseUnit	300	(12C)	SHORT
ColorResponseCurve	301	(12D)	SHORT
Software	305	(131)	ASCII
DateTime	306	(132)	ASCII
Artist	315	(13B)	ASCII

NAMA TAG	NOMOR TAG		TYPE DATA
	DESIMAL	HEXADESIMAL	
HostComputer	316	(13C)	ASCII
Predictor	317	(13D)	SHORT
WhitePoint	318	(13E)	RATIONAL
PrimaryChromaticities	319	(13F)	RATIONAL
ColorMap	320	(140)	SHORT
HalftoneHints	321	(141)	SHORT
TileWidth	322	(142)	SHORT/LONG
TileLength	323	(143)	SHORT/LONG
TileOffsets	324	(144)	LONG
TileByteCounts	325	(145)	SHORT/LONG
InkSet	332	(14C)	SHORT
InkNames	333	(14D)	ASCII
NumberOfInks	334	(14E)	SHORT
DotRange	336	(150)	BYTE/SHORT
TargetPrinter	337	(151)	ASCII
ExtraSamples	338	(152)	BYTE
SampleFormat	339	(153)	SHORT
SMinSampleValue	340	(154)	any
SMaxSampleValue	341	(155)	any
JPEGProc	512	(200)	SHORT
JPEGInterchangeFormat	513	(201)	LONG
JPEGInterchangeFormatLngth	514	(202)	LONG
JPEGRestartInterval	515	(203)	SHORT
JPEGLosslessPredictors	517	(205)	SHORT
JPEGPointTransform	518	(206)	SHORT
JPEGTables	519	(207)	LONG
JPEGTables	520	(208)	LONG
JPEGTables	321	(209)	LONG
YCbCrCoefficients	529	(211)	RATIONAL
YCbCrSubSampling	530	(212)	SHORT
YCbCrPositioning	531	(213)	SHORT
ReferenceBlackWhite	532	(214)	LONG
Copyright	33432	(8298)	ASCII

Pada umumnya tag tersebut tidak digunakan secara keseluruhan, tetapi hanya sebagian saja. Tag yang umum digunakan meliputi :

ImageWidth

Mendefinisikan lebar citra (arah horisontal) dalam satuan piksel.

ImageLength

Menyatakan panjang citra (arah vertikal) dalam satuan piksel.

BitsPerSample

Tag ini mendefinisikan jumlah bit per sample. Defaultnya adalah satu bit per sample.

Compression

Tag ini mendefinisikan type dari metode pemampatan data yang digunakan suatu citra dalam menyimpan file TIFF. Defaultnya adalah satu, yang berarti tanpa pemampatan.

Nilai yang dapat digunakan sebagai berikut :

1	No compression
2	Huffman run length encoding
3	CCITT group 3 FAX compression
4	CCITT group 4 FAX compression
32773	PackBits run length encoding

PhotometricInterpretation

Menyatakan bagaimana suatu citra disimpan.

Nilainya sebagai berikut :

0 Jika citra hitam putih disimpan secara terbalik (kebalikan dari citra aslinya, misalnya piksel hitam menjadi piksel putih, begitu pula sebaliknya).

1 Jika citra hitam putih disimpan secara normal.

2 Citra yang disimpan menggunakan warna RGB (Red, Green, Blue).

3 Citra yang akan disimpan menggunakan warna palet.

4 Citra hitam-putih transparency mask.

StripOffset

Mendefinisikan offset masing-masing strip suatu citra terhadap awal file.

SamplesPerPixel

Mendefinisikan jumlah sample per piksel.

Defaultnya adalah 1.

RowsPerStrip

Tag ini mendefinisikan tinggi (dalam satuan baris) dari masing-masing strip untuk citra yang memiliki strip lebih dari satu.

Pada citra yang memiliki strip hanya satu, nilai tag ini lebih besar atau sama dengan tinggi citra.

StripByteCounts

Mendefinisikan jumlah byte pada masing-masing strip dari suatu citra.

XResolution

Mendefinisikan jumlah piksel secara horisontal tiap Resolution Unit

YResolution

Mendefinisikan jumlah piksel secara vertikal tiap Resolution Unit.

PlanarConfiguration

Tag ini menunjukkan apakah piksel-piksel citra disimpan secara bersebelahan atau dalam plane (halaman memori) yang terpisah.

Defaultnya adalah 1.

Nilainya sebagai berikut :

1 piksel disimpan secara bersebelahan dalam satu plane.

2 piksel disimpan dalam plane yang terpisah.

2.3. ROM BIOS Video Services

BIOS (Basic Input/Output System) merupakan perangkat lunak tingkat rendah yang telah ada pada komputer. Perangkat lunak yang terdapat pada level terendah system ini biasanya disebut ROM (Read Only Memeory) BIOS. ROM BIOS dapat juga diartikan sebagai instruksi-instruksi elektronik yang telah dipatrikan pada chip komputer, sehingga instruksi-instruksi tersebut tetap berada pada memori walaupun komputer telah

dimatikan. Karena BIOS merupakan input dan output dasar pada system, maka fungsi-fungsi atau subroutine yang berada ROM BIOS dapat dipanggil oleh program-program aplikasi yang membutuhkan hubungan dengan perangkat keras komputer. Perangkat keras di sini dapat diartikan media disk, layar monitor, memori, dan perangkat keras lainnya yang berhubungan dengan komputer.

ROM BIOS Video Services menyediakan fasilitas untuk menampilkan teks dan grafik pada layar tampilan IBM PC. Seorang programmer dijamin softwarenya akan bekerja dengan beberapa mesin IBM yang kompatibel dan dengan beberapa macam tampilan warna atau monochrome yang kompatibel (kecuali mode grafik dari *Hercules Graphics Card*) dengan menggunakan service-service yang terdapat pada ROM BIOS.

Semua ROM BIOS video services dibangkitkan dengan menggunakan interrupt 16 (10H). Service-service yang dipilih ditentukan oleh nomor service yang ditempatkan pada register AH sebelum interrupt dibangkitkan. Seringkali suatu service memerlukan bermacam-macam parameter masukan dalam register-register mikroprosesor yang lain sebelum interrupt dibangkitkan. Kadang-kadang data yang dihasilkan oleh service disimpan dalam register yang lain untuk digunakan setelah interrupt diakhiri.

Berikut ini adalah ROM BIOS Video Service yang digunakan untuk menyelesaikan permasalahan pada tugas akhir ini.

Tabel 2.2. Set mode video

INPUT KE BIOS	OUTPUT DARI BIOS
AH = 00H AL = mode video	tidak ada

Tabel 2.3. Menulis piksel ke layar

INPUT KE BIOS	OUTPUT DARI BIOS
AH = 0CH AL = warna piksel BH = halaman tampilan DX = nomor baris CX = nomor kolom	tidak ada

Tabel 2.4. Menampilkan mode video yang sedang aktif

INPUT KE BIOS	OUTPUT DARI BIOS
AH = 0FH	AL = mode video yang aktif AH = jumlah karakter per baris BH = nomor halaman yang aktif

Tabel 2.5. Menentukan nilai intensitas warna primer

INPUT KE BIOS	OUTPUT DARI BIOS
AH = 10H AL = 10H BX = nomor warna CH = nilai Green CL = nilai Blue DH = nilai Red	tidak ada

Tabel 2.6. Menampilkan nilai intensitas warna primer

INPUT KE BIOS	OUTPUT DARI BIOS
AH = 10H AL = 15H BX = nomor warna	CH = nilai Green CL = nilai Blue DH = nilai Red

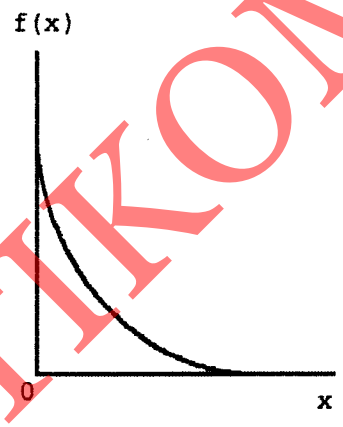
2.4. Noise

Pada proses perekaman citra dijital dapat terjadi gangguan (noise) yang bersifat frekuensi rendah, yaitu terjadi proses pemerataan intensitas cahaya pada suatu titik dengan titik-titik tetangganya. Gangguan lain yang sering terjadi pada proses perekaman citra dijital adalah terjadinya gangguan berbentuk garis-garis sebagai akibat adanya kerusakan pada sebagian detektor sensor . Sering juga dijumpai gangguan dalam bentuk bercak hitam atau putih yang tersebar secara acak di seluruh bagian citra.

Kesulitan yang dihadapi dalam hal ini adalah mendapatkan citra yang mempunyai gangguan dari hasil proses perekaman. Oleh karena itu diciptakanlah gangguan (noise) tersebut dengan cara tertentu. Gangguan tersebut dibangkitkan dengan distribusi tertentu dari *Distribusi Uniform(0,1)* dengan menggunakan metode *Transformasi Invers.* Berikut ini contoh metode Transformasi Invers untuk beberapa distribusi, (M. Law, 1991).

2.4.1. Distribusi Eksponensial

Fungsi densitas : $f(x) = \begin{cases} \frac{1}{\beta} e^{\frac{-x}{\beta}} & ; x \geq 0 \\ 0 & ; x < 0 \end{cases}$



Mean : β

Varian : β^2

Gambar 2.6. Fungsi densitas distribusi eksponensial

Fungsi Distribusinya diperoleh dari hasil integral fungsi densitas di atas.

$$F(x) = \int_0^x f(x)\ d(x)$$

$$= \int_0^x 1/\beta\ e^{-x/\beta}\ dx$$

$$= 1 - e^{-x/\beta}$$

Hasil invers Fungsi Distribusi digunakan untuk membangkitkan noise.

$$F(x) = 1 - e^{-x/\beta}$$

$$F^{-1}(x) = -\beta\ \ln(1-x)$$

Fungsi distribusi : $F(x) = \begin{cases} 1 - e^{\frac{-x}{\beta}} & ; x \geq 0 \\ 0 & ; x < 0 \end{cases}$

Berdasarkan perhitungan di atas, maka untuk membangkitkan noise yang berdistribusi Eksponensial dapat digunakan rumus sebagai berikut :

$$ExpNoise = F^{-1}(U)$$

$$= -\beta \ln(1-U)$$

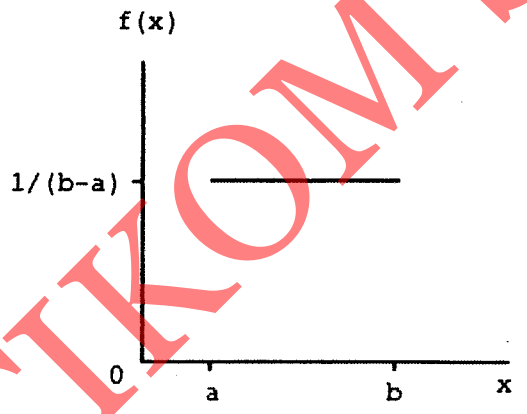
$$\boxed{ExpNoise = -\beta \ln U}$$

U adalah bilangan random antara 0 sampai dengan 1.

Berikut ini terdapat beberapa macam distribusi :

2.4.2. Distribusi Uniform

Fungsi densitas : $f(x) = \begin{cases} \frac{1}{(b-a)} & ; a \leq x \leq b \\ 0 & ; x < a \text{ dan } x > b \end{cases}$



Mean = (a+b)/2

Varian = $(b-a)^2/12$

Gambar 2.7. Fungsi densitas distribusi uniform

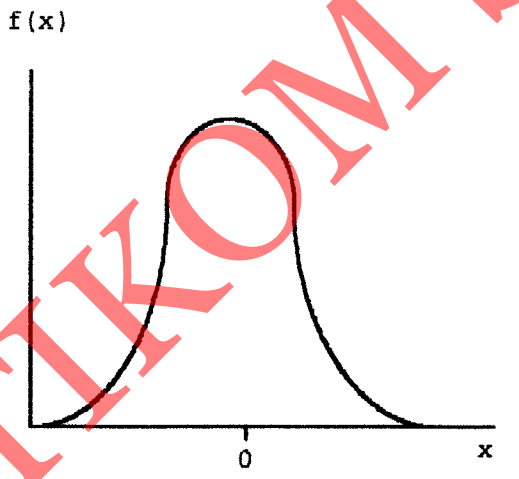
$$\text{Fungsi distribusi} : F(x) = \begin{cases} 0 & ; x < a \\ \dfrac{(x-a)}{(b-a)} & ; a \leq x \leq b \\ 1 & ; b < x \end{cases}$$

Rumus untuk membangkitkan noise yang berdistribusi Uniform adalah :

$$\text{UniformNoise} = a+(b-a)U$$

2.4.3. Distribusi Gaussian

$$\text{Fungsi densitas} : f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad ; \quad -\infty < x < \infty$$



Mean = μ

Varian = σ^2

Gambar 2.8. Fungsi densitas distribusi gaussian

Fungsi Distribusi : (tidak ada)

Untuk membangkitkan noise yang berdistribusi Gaussian digunakan rumus seperti di bawah ini :

$$X_i = \mu + \sqrt{\sigma^2}\,(2U_i - 1)\sqrt{\frac{-2 \ln \sum_{j=1}^{2}(2U_j - 1)^2}{\sum_{j=1}^{2}(2U_j - 1)^2}}$$

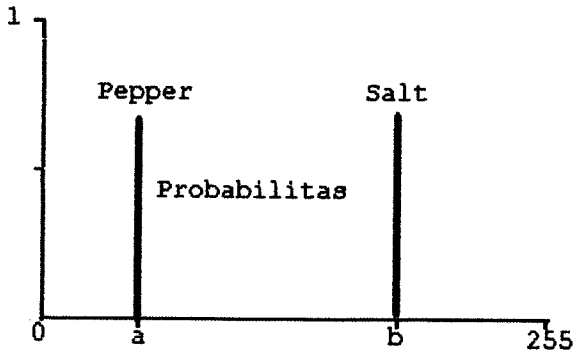
$$i = 1..2$$

2.4.4. Salt and Pepper Noise

Salt and Pepper Noise adalah tipe noise yang umum terjadi dalam suatu citra. Noise ini tampak sebagai titik hitam dan/atau titik putih yang tersebar secara acak di seluruh citra.

Pada citra grayscale, Salt tampak sebagai piksel dengan tingkat keabuan sebesar 255, sedangkan Pepper mempunyai tingkat keabuan sebesar 0.

Bentuk Histogramnya seperti tampak pada Gambar 2.9.



Gambar 2.9. Histogram salt and pepper noise

2.5. Filtering

Pada proses perekaman citra dijital dapat terjadi gangguan yang bersifat frekuensi rendah, dimana terjadi proses pemerataan intensitas cahaya pada suatu titik dengan titik tetangganya. Gangguan lain yang sering terjadi pada proses perekaman citra dijital adalah terjadinya gangguan bentuk garis-garis sebagai akibat adanya kerusakan pada sebagian detektor sensor. Sering juga dijumpai gangguan lain dalam bentuk bercak hitam yang acak. Filtering adalah salah satu proses yang terdapat pada pengolahan citra guna menghilangkan gangguan-gangguan diatas. Proses filtering mempunyai beberapa metode, antara lain Adaptive Filter, Morphological Filter, Nonlinear Filter dan Spatial Filter. Masing-masing metode tersebut juga terdiri dari

beberapa cara yang akan dijelaskan pada bagian berikutnya.

2.5.1. Adaptive Filter

Tergantung pada jenis gangguan (noise) yang muncul pada suatu citra, jenis filter yang dipilih untuk menghilangkan gangguan tersebut hendaknya tidak mempengaruhi detail-detail penting yang harus tetap terjaga pada citra yang bersangkutan. Contohnya, Gaussian noise lebih baik dihilangkan menggunakan metode Mean Filter, sedangkan Salt and Pepper noise lebih baik dihilangkan menggunakan Median Filter. Kerugian penggunaan Mean Filter dibandingkan Median Filter yaitu Mean Filter dapat menghilangkan gangguan spatial frekuensi tinggi tetapi mengaburkan batas tepi (edge) suatu citra. Gangguan spatial frekuensi tinggi (*High Spatial Frequancy*) adalah suatu gangguan di mana terjadi perubahan intensitas yang besar dan cepat dari suatu piksel ke piksel tetangganya, sehingga membentuk garis-garis pada citra.

Suatu filter dianggap ideal untuk digunakan pada suatu citra jika filter tersebut dapat menyesuaikan diri dengan cara mengubah karakteristiknya agar sesuai dengan isi lokal window pada citra yang bersangkutan. Contohnya,

jika isi lokal window hanya berupa batas tepi (edge), maka Median Filter dapat digunakan untuk mempertahankan batas tepinya. Jika dalam lokal window terdeteksi adanya background yang homogen, maka filter akan mengubah karakteristiknya dengan pemakaian Mean Filter dalam lokal window ini. Filter-filter yang secara dinamik mengubah karakteristik filternya sesuai dengan isi citranya dikenal sebagai adaptive filter. Daya guna Adaptive Filter tergantung pada ketepatan penentuan mean, standard deviasi, dan standard deviasi noise.

Dalam adaptive filter dikenal adanya metode Double Window Modified Trimmed Mean (DW-MTM) Filter dan Minimum Mean-Square Error (MMSE) Filter.

A. DW-MTM Filter

Adaptive *Double Window Modified Trimmed Mean* (DW-MTM) filter efektif untuk menghilangkan titik-titik yang terisolir (outlier) seperti yang diakibatkan oleh salt and pepper noise.

Algoritma adaptive DW-MTM filter digambarkan sebagai berikut, misalnya suatu piksel X berada di tengah-tengah lokal window 3x3 (ukuran lokal window biasanya berupa bilangan ganjil 3x3,5x5,7x7,atau 9x9). Piksel di lokal window itu dicari nilai mediannya

($MED(C_i)$). Nilai median yang diperoleh digunakan untuk menentukan nilai mean dari lokal window 5x5 dengan X sebagai pusatnya.

Dalam menghitung nilai mean pada lokal window 5x5, hanya menggunakan piksel yang mempunyai nilai graylevel:

$$MED(C_i) - s \quad \text{sampai dengan} \quad MED(C_i) + s$$
$$i = 1,2,3,\ldots,9$$

C_i adalah piksel dalam lokal window, dimana C_1 adalah piksel di kolom pertama baris pertama, C_2 adalah piksel di kolom kedua baris pertama, dan seterusnya, sampai dengan C_9 di kolom ketiga baris ketiga.

Nilai mean yang diperoleh merupakan output dari DW-MTM filter yang akan menggantikan nilai piksel X.

s adalah standard deviasi suatu noise, dimana s mempunyai nilai sebagai berikut :

$$s = K \cdot \sigma_n$$

Umumnya nilai *K* adalah 1.5 sampai dengan 2.5, (R. Weeks, 1993). σ_n adalah standard deviasi suatu noise.

B. MMSE Filter

Minimum Mean-Square Error (MMSE) filter efektif digunakan pada *short tail noise*. Output dari MMSE filter untuk piksel yang berada di lokasi $C_{N/2+1/2}$ dalam suatu lokal window adalah :

$$MMSE(C) = (1 - \frac{\sigma_n^2}{\sigma_l^2}).C_{\frac{N}{2}+\frac{1}{2}} + \frac{\sigma_n^2}{\sigma_l^2}.K$$

Parameter K adalah nilai rata-rata (mean) dari piksel dalam lokal window.

σ_n^2 adalah variance dari noise, dan σ_l^2 adalah lokal variance yang diperoleh dari perhitungan :

$$\sigma_l^2 = \frac{\sum_{i=1}^{N} (C_i)^2}{N} - (\frac{\sum_{i=1}^{N} C_i}{N})^2$$

N adalah jumlah piksel dalam lokal window dan C_i menyatakan piksel yang berada dalam lokal window.

C_1 adalah piksel di kolom pertama baris pertama, C_2 adalah piksel di kolom kedua baris pertama, dan seterusnya sampai dengan C_N adalah piksel di kolom $\sqrt{N}$ baris $\sqrt{N}$.

Contoh : N = 9 lokal window 3x3

3	10	6
1	2	7
0	20	15

$C_1 = 3,\ C_2 = 10,\ C_3 = 6,\ C_4 = 1,\ C_5 = 2,\ C_6 = 7$

$C_7 = 0,\ C_8 = 20,\ C_9 = 15$

Pengertian di atas digunakan untuk uraian selanjutnya. Pada bagian citra yang homogen (background), nilai lokal variance kira-kira sama dengan nilai variance dari noise, sedangkan di daerah batas tepi (edge), nilai lokal variance lebih besar daripada nilai variance dari noise.

2.5.2. Morphological Filter

Morphological filter adalah salah satu metode pengolahan citra yang biasa dilakukan sebelum proses pengenalan pola (*pattern recognition*) dan proses identifikasi untuk memperhalus garis bentuk (*contour*)

suatu obyek dan menyusun kembali suatu citra dalam bentuk geometris dasarnya.

Operasi morphological filtering bisa dipisahkan dalam dua kategori, yaitu *binary morphological filtering* untuk citra monochrome (black and white) dan *graylevel morphological filtering* untuk citra grayscale. Semua operasi filtering untuk kedua kategori di atas diturunkan dari dua operasi dasar morphological, yaitu *dilation* dan *erosion*. Dalam tugas akhir ini tidak akan membahas mengenai binary morphological filtering, tetapi hanya graylevel morphological filtering.

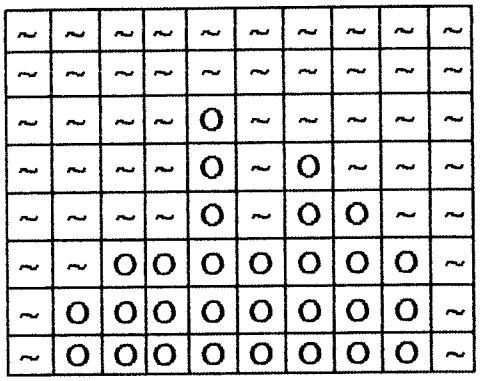
Metode filtering yang termasuk dalam morphological filter meliputi erosion filter, dilation filter, closing filter, dan opening filter.

A. Erosion Filter

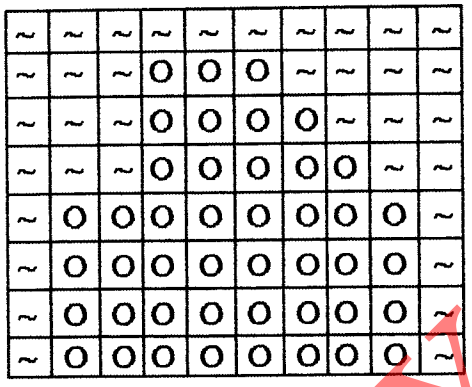
Erosion filter digunakan untuk memperhalus garis bentuk suatu obyek dengan cara membuat piksel di daerah batas tepi menjadi lebih gelap.

Perubahan citra yang dihasilkan dari operasi erosion filter tampak seperti pada gambar 2.10.

Erosion filter didefinisikan sebagai nilai minimum dari perbedaan antara piksel dalam lokal window dengan mask yang digunakan.



citra asli



citra hasil erosion filter

O : obyek (gelap)

~ : background (terang)

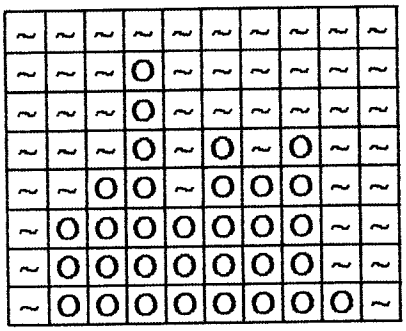
Gambar 2.10. Perubahan citra hasil proses erosion filter

$$C \ominus M = \min(C_i - M_i)$$
$$i = 1,2,3,\ldots,N$$

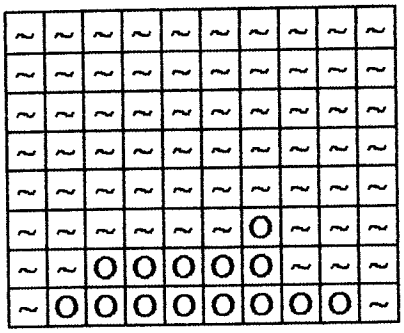
M adalah mask yang digunakan. Ukuran mask sama dengan ukuran lokal window. Jika semua nilai pada mask sama dengan nol, maka erosion filter menghasilkan *minimum filter*.

B. Dilation Filter

Dilation filter digunakan untuk memperhalus garis bentuk suatu obyek dengan cara membuat piksel di daerah batas tepi menjadi lebih terang.



citra asli



citra hasil dilation filter

O : obyek (gelap)

~ : background (terang)

Gambar 2.11. Perubahan citra hasil operasi dilation filter

Perubahan citra hasil operasi dilation filter dapat dilihat pada gambar 2.11.

Dilation filter didefinisikan sebagai nilai maksimum dari penjumlahan antara piksel dalam lokal window dengan mask yang digunakan.

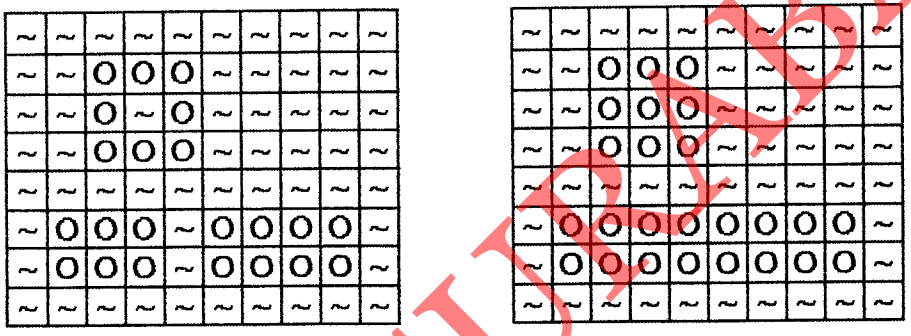
$$C \oplus M = \max(C_i + M_i)$$
$$i = 1,2,3,\ldots,N$$

Jika semua nilai pada mask yang digunakan sama dengan nol, maka dilation filter menghasilkan *maximum filter*.

C. Closing Filter

Closing filter didefinisikan sebagai operasi dilation filter diikuti oleh operasi erosion filter. Operasi closing filter akan menggabungkan obyek yang terpisah dan menutup lubang dalam obyek.

Citra yang dihasilkan oleh operasi closing filter tampak seperti pada gambar 2.12.



citra asli citra hasil closing filter

O : obyek (gelap)

~ : background (terang)

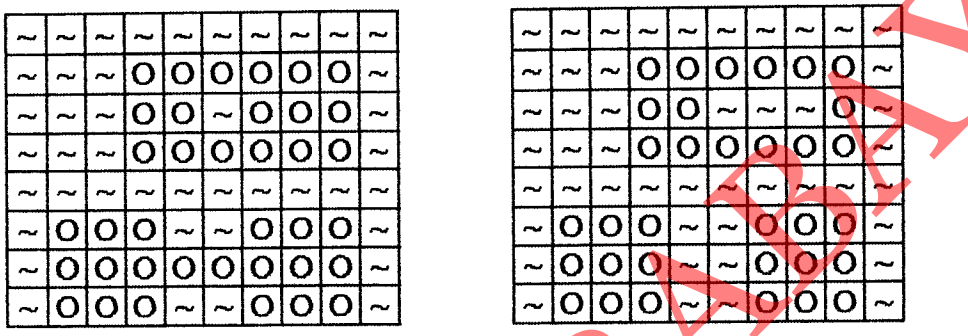
Gambar 2.12. Perubahan citra hasil proses closing filter

$$Close(C,M) = (C \oplus M) \ominus M$$

D. Opening Filter

Opening filter didefinisikan sebagai operasi erosion filter diikuti oleh operasi dilation filter.

Opening filter digunakan untuk memisahkan obyek-obyek yang saling berdekatan dan memperbesar lubang dalam obyek. Hasil perubahan operasi opening filter seperti pada gambar 2.13.



citra asli　　citra hasil opening filter

O : obyek (gelap)

~ : background (terang)

Gambar 2.13. Perubahan citra hasil proses opening filter

$$Open(C,M) = (C \ominus M) \oplus M$$

2.5.3. Nonlinear Filter

Nonlinear filtering merupakan bagian paling besar dari proses filtering. Nonlinear filter beroperasi pada citra dengan cara menghitung fungsi nonlinear seluruh lokal window dan mengganti piksel yang berada di tengah lokal window dengan nilai ini. Salah satu metode

nonlinear filter adalah *Median Filter*. Median filter umumnya digunakan untuk menghilangkan salt and pepper noise dengan hasil yang lebih baik daripada *Arithmetic Mean Filter* karena bisa mempertahankan keberadaan edge dalam citra. Nonlinear filter tidak hanya terbatas untuk menghilangkan noise dari citra. Contohnya *Range Filter* dapat digunakan untuk mendeteksi batas tepi (edge) dalam citra.

Nonlinear filter terdiri dari beberapa metode, antara lain Alpha-Trimmed Mean, Contra-Harmonic, Geometric Mean, Harmonic Mean, Maximum, Median, Midpoint, Minimum, Weighted Median, dan Yp Mean filter.
Berikut ini akan dijelaskan masing-masing metode di atas satu per satu.

A. Alpha-Trimmed mean Filter

Alpha-Trimmed Mean filter digunakan pada citra yang berisi *short* dan *long tailed noise*. Contohnya, citra yang berisi Gaussian dan salt and pepper noise. Untuk mendefinisikan alpha-trimmed mean filter, semua piksel yang berada di lokal window diurutkan dari nilai graylevel minimum ke maksimum.

$$A_1 \leq A_2 \leq A_3 \leq A_4 \leq \cdots \leq A_{N-1} \leq A_N$$

Alpha-trimmed mean filter mempunyai rumus sebagai berikut :

$$\text{AlphaMean}(C) = \frac{1}{N-2P} \sum_{i=P}^{N-P} C_i$$

Variabel P digunakan untuk menentukan berapa banyak jumlah piksel yang dihilangkan dari urutan data.

Contoh : lokal window 3x3

23	10	20
5	8	2
15	30	31

Piksel-piksel tersebut diurutkan menjadi :

2 5 8 10 15 20 23 30 31

Jika P=1 maka piksel 2 dan 31 tidak terlibat dalam perhitungan rata-rata.

Jika P=2 maka piksel 2,5,30, dan 31 tidak terlibat dalam perhitungan rata-rata, dan seterusnya.

Apabila P=0 dan P=(N/2)-1/2, dimana N adalah jumlah piksel dalam lokal window, maka alpha-trimmed mean filter menjadi Mean Filter dan Median Filter.

B. Contra-Harmonic Filter

Contra-Harmonic filter adalah salah satu metode dalam nonlinear filter yang efektif untuk menghilangkan positif outlier (white spot) dan negatif outlier (black spot). Definisi dari contra-harmonic filter adalah sebagai berikut :

$$\text{ContraHarmonic}(C) = \frac{\sum_{i=1}^{N} (C_i)^{P+1}}{\sum_{i=1}^{N} (C_i)^{P}}$$

Jika P bernilai negatif maka contra-harmonic mean filter digunakan untuk menghilangkan positif outlier, sebaliknya jika P bernilai positif maka digunakan untuk menghilangkan negatif outlier.

C. Geometric Mean Filter

Geometric Mean filter didefinisikan sebagai hasil kali piksel dalam lokal window yang sudah dipangkatkan dengan 1/N. N adalah jumlah piksel dalam lokal window. Geometric mean filter sangat rentan terhadap negatif outlier (black spot). Definisi dari geometric mean filter adalah sebagai berikut :

$$\text{GeometricMean}(C) = \prod_{i=1}^{N} (C_i)^{\frac{1}{N}}$$

D. Harmonic Mean Filter

Harmonic Mean filter efektif untuk menghilangkan positif outlier (white spot). Definisi harmonic mean filter adalah sebagai berikut :

$$\text{HarmonicMean}(C) = \frac{N}{\sum_{i=1}^{N} \left(\frac{1}{C_i}\right)}$$

E. Maximum Filter

Maximum filter biasanya digunakan pada citra untuk menghilangkan negatif outlier noise (black spot).
Pada filter ini outputnya diperoleh dengan cara mencari nilai maksimum dari seluruh piksel dalam lokal window.
Maximum filter didefinisikan sebagai berikut :

$$\text{Maximum}(C) = \max(C_i)$$
$$i = 1,2,3,\ldots,N$$

F. Median Filter

Operasi *Median* filter digunakan untuk menghilangkan *long tailed noise* seperti exponential dan salt and pepper noise. Langkah pertama yang harus dilakukan adalah mengurutkan semua piksel dalam lokal window kemudian mencari nilai mediannya yang akan menggantikan nilai piksel di pusat lokal window. Median filter efektif untuk menghilangkan outlier noise yang jumlahnya kurang dari 50% jumlah piksel dalam citra.

Definisi median filter dalam citra C adalah :

$$\text{Median}(C_i) = \text{med}(C_i)$$
$$i = 1,2,3,\ldots,N$$

G. Minimum Filter

Minimum filter biasanya digunakan pada citra untuk menghilangkan positive outlier noise (white spot). Output minimum filter diperoleh dengan cara mencari nilai minimum dari seluruh piksel dalam lokal window. Nilai itulah yang digunakan untuk mengganti nilai piksel di pusat lokal window.

Minimum filter didefinisikan sebagai berikut :

$$\text{Minimum}(C) = \min(C_i) \quad i = 1,2,3,\ldots,N$$

H. Midpoint Filter

Midpoint filter biasanya digunakan untuk filter citra yang berisi short tailed noise seperti Gaussian dan Uniform noise. Output midpoint filter adalah rata-rata nilai maksimum dan minimum graylevel dalam lokal window. Definisi dari midpoint filter adalah :

$$\text{Midpoint}(C) = \frac{\text{Maximum}(C) + \text{Minimum}(C)}{2}$$

I. Weighted Median Filter

Weighted Median berbeda dari median filter. Pada filter ini piksel dalam lokal window diulangi sebanyak masing-masing elemen mask, kemudian diurutkan. Dari data yang sudah urut tersebut, diambil nilai mediannya sebagai output weighted median filter.

Weighted median didefinisikan seperti di bawah ini :

$$\text{WeightedMedian}(C) = \text{Median}\ (\text{Repeat } M_i \text{ times } \{C_i\}) \quad i = 1,2,3,\ldots,N$$

M_i adalah elemen mask.

Contoh :

lokal window 3x3		
25	17	10
33	15	20
12	29	28

mask 3x3		
1	2	3
4	5	6
7	8	9

Pengulangan :

25 17 17 10 10 10 33 33 33 33

15 15 15 15 15 20 20 20 20 20

20 12 12 12 12 12 12 12 29 29

29 29 29 29 29 29 28 28 28 28

28 28 28 28 28

Piksel di atas diurutkan menjadi :

10 10 10 12 12 12 12 12 12 12

15 15 15 15 15 17 17 20 20 20

20 20 **20** 25 28 28 28 28 28 28

28 28 28 29 29 29 29 29 29 29

29 33 33 33 33

Diambil nilai mediannya, yaitu piksel **20**

J. Yp Mean Filter

Yp Mean filter efektif untuk menghilangkan positive outlier jika P bernilai negatif dan negative outlier jika P bernilai positif. Definisi Yp mean filter adalah :

$$\text{YpMean}(C) = \left(\frac{\sum_{i=1}^{N} (C_i)^P}{N} \right)^{\frac{1}{P}}$$

2.5.4. Spatial Filter

Spatial filter mempunyai aplikasi dasar dalam pengolahan citra. Diantaranya menghilangkan gangguan (noise), penghalusan (smoothing), dan penajaman batas tepi (edge). Noise pada suatu citra biasanya tampak sebagai titik-titik hitam atau putih yang tersebar secara acak di seluruh citra.

Cara yang dilakukan dalam metode ini adalah dengan proses konvolusi antara suatu citra dengan suatu mask.

Dalam hal ini yang dimaksud proses konvolusi adalah operasi perkalian dan penjumlahan antara piksel dalam lokal window dengan elemen mask, yaitu citra kecil (misalnya : 3x3, 5x5, atau 7x7 piksel).

Contoh : lokal window 3x3 mask 3x3

7	10	2	-1	0	-1
6	8	15	0	4	0
12	1	5	-1	0	-1

proses konvolusi

$$y = (7x-1)+(10x0)+(2x-1)+(6x0)+(8x4)+$$
$$(15x0)+(12x-1)+(1x0)+(5x-1)$$
$$y = -7 + 0 - 2 + 0 + 32 + 0 - 12 + 0 - 5$$
$$y = 32 - 26$$
$$y = 6$$

Nilai 6 akan menggantikan piksel 8 dalam lokal window.

Pada Spatial filter terdapat beberapa metode, yaitu Arithmetic Mean Filter, Gaussian Filter, High Pass Filter, Low Pass Filter, dan Weighted Mean Filter.

A. Arithmetic Mean Filter

Operasi *Arithmetic Mean* filter digunakan untuk menghilangkan short tailed noise seperti Uniform dan Gaussian noise. Arithmetic mean filter didefinisikan sebagai nilai rata-rata dari semua piksel yang berada dalam lokal window. Dibandingkan dengan metode filtering lainnya, metode ini tidak dapat mempertahankan edge dalam

citra, atau dengan kata lain metode ini menyebabkan bagian citra yang berupa batas tepi tampak kabur. Semakin besar lokal window akan menyebabkan citra semakin kabur. Filter ini didefinisikan sebagai berikut :

$$\text{ArithmeticMean}(C) = \frac{1}{N} \sum_{i=1}^{N} C_i$$

B. Gaussian Filter

Gaussian filter adalah suatu metode filtering yang dilakukan dengan cara proses konvolusi antara citra dengan mask yang dibetuk dari distribusi Gaussian. Semakin besar kolom atau baris mask menyebabkan gambar tampak semakin tidak terfokus atau kabur.

Contoh Gaussian mask adalah seperti di bawah ini :

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

C. High Pass Filter

High Pass filter berfungsi untuk menajamkan batas tepi suatu citra. Pada filter ini dilakukan proses konvolusi antara citra dengan mask. Ciri-ciri mask yang digunakan adalah adanya elemen-elemen yang bernilai negatif dan positif. Jika elemen-elemen mask tersebut dijumlah akan menghasilkan nilai satu.

Contoh :

0 -1 0	-1 -1 -1	1 -2 1
-1 5 -1	-1 9 -1	-2 5 -2
0 -1 0	-1 -1 -1	1 -2 1

D. Low Pass Filter

Proses filtering suatu citra dengan *Low Pass* filter mempunyai efek pemerataan tingkat keabuan, sehingga citra yang diperoleh akan tampak agak kabur kontrasnya.

Proses ini dilakukan terhadap citra agar gangguan yang berbentuk garis tajam dapat dikurangi efeknya. Metode ini menggunakan proses konvolusi dengan suatu mask. Mask yang digunakan mempunyai elemen bernilai positif dan jika elemen-elemen tersebut dijumlah akan menghasilkan nilai satu.

Contoh :

1/9 1/9 1/9	0 1/6 0
1/9 1/9 1/9	1/6 1/3 1/6
1/9 1/9 1/9	0 1/6 0

1/10	1/10	1/10	1/16	1/8	1/16
1/10	1/5	1/10	1/8	1/4	1/8
1/10	1/10	1/10	1/16	1/8	1/16

E. Weighted Mean Filter

Weighted Mean filter didefinisikan sebagai rata-rata pembobotan semua piksel dalam lokal window. Bobot masing-masing piksel dalam lokal window ditentukan oleh mask yang digunakan.

Contoh mask 5x5 :

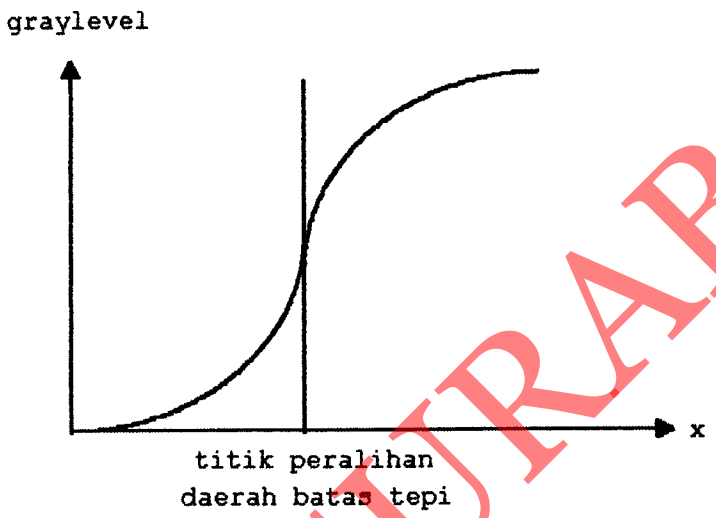
1	1	1	1	1
1	2	2	2	1
1	2	3	2	1
1	2	2	2	1
1	1	1	1	1

Rumus weighted mean filter :

$$\text{WeightedMean}(C) = \frac{\sum_{i=1}^{N} M_i . C_i}{\sum_{i=1}^{N} M_i}$$

2.6. Deteksi Batas Tepi

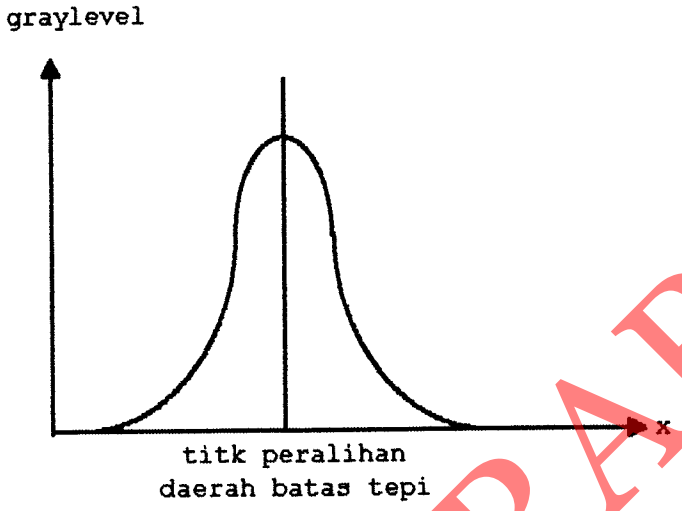
Batas Tepi (*edge*) adalah daerah dalam suatu citra yang berada di tengah perubahan nilai graylevel dari rendah menjadi tinggi seperti tampak pada gambar 2.14.



Gambar 2.14. Nilai graylevel daerah batas tepi pada citra asli

Deteksi batas tepi (*edge detection*) merupakan salah satu proses pra pengolahan yang sering dibutuhkan pada analisis citra. Proses tersebut bertujuan untuk meningkatkan penampakan garis batas tepi pada citra. Batas tepi yang dideteksi berwarna lebih terang, sedangkan daerah lainnya berwarna gelap. Perubahan nilai graylevel dari suatu batas tepi selalu bernilai positif

atau nol, dan mencapai nilai maksimum di daerah batas tepi. Hal ini dapat dilihat pada gambar 2.15.



Gambar 2.15. Nilai graylevel daerah batas tepi yang sudah dideteksi

Proses konvolusi suatu citra dengan mask merupakan teknik yang umum digunakan dalam proses deteksi batas tepi. Proses deteksi batas tepi juga bisa menggunakan metode lain yang juga akan dibahas di sini.

A. Kirsch, Prewitt, dan Sobel Edge Detector

Mask Kirsch, Prewitt, dan Sobel adalah mask yang digunakan untuk proses deteksi batas tepi dengan metode konvolusi. Masing-masing mask tersebut bisa mendeteksi batas tepi sebanyak delapan arah, yaitu Utara, Timur Laut, Timur, Tenggara, Selatan, Barat Daya, Barat, dan Barat Laut. Mask di atas tidak dapat mendeteksi delapan

arah sekaligus, tetapi hanya satu arah diantara delapan arah tadi. Bentuk mask mask Kirsch, Prewitt, dan Sobel adalah sebagai berikut :

	Kirsch	Prewitt	Sobel
Utara	-3 -3 -3 -3 0 -3 5 5 5	-1 -1 -1 1 -2 1 1 1 1	-1 -2 -1 0 0 0 1 2 1
Timur Laut	-3 -3 -3 5 0 -3 5 5 -3	1 -1 -1 1 -2 -1 1 1 1	0 -1 -2 1 0 -1 2 1 0
Timur	5 -3 -3 5 0 -3 5 -3 -3	1 1 -1 1 -2 -1 1 1 -1	1 0 -1 2 0 -2 1 0 -1
Tenggara	5 5 -3 5 0 -3 -3 -3 -3	1 1 1 1 -2 -1 1 -1 -1	2 1 0 1 0 -1 0 -1 -2
Selatan	5 5 5 -3 0 -3 -3 -3 -3	1 1 1 1 -2 1 -1 -1 -1	1 2 1 0 0 0 -1 -2 -1
Barat Daya	-3 5 5 -3 0 5 -3 -3 -3	1 1 1 -1 -2 1 -1 -1 1	0 1 2 -1 0 1 -2 -1 0
Barat	-3 -3 5 -3 0 5 -3 -3 5	-1 1 1 -1 -2 1 -1 1 1	-1 0 1 -2 0 2 -1 0 1
Barat Laut	-3 -3 -3 -3 0 5 -3 5 5	-1 -1 1 -1 -2 1 1 1 1	-2 -1 0 -1 0 1 0 1 2

B. Quick Mask Edge Detector

Deteksi batas tepi dengan mask Kirsch, Prewitt, dan Sobel hanya bisa mendeteksi satu arah saja. Apabila ingin mendeteksi batas tepi delapan arah sekaligus hanya dengan satu proses konvolusi, maka bisa digunakan *Quick Mask*. Mask ini dibuat oleh Dwayne Philips. Bentuknya sebagai berikut :

```
-1  0 -1
 0  4  0
-1  0 -1
```

C. Faler Edge Detector

Wesley Faler menggunakan beberapa macam mask untuk meningkatkan perubahan nilai graylevel suatu batas tepi. Mask-mask tersebut adalah seperti di bawah ini :

```
-1  0  1      1  1  1      -1 -1 -1      0  1  0
-1  0  1      0  0  0      -1  8 -1     -1  0  1
-1  0  1     -1 -1 -1      -1 -1 -1      0 -1  0
```

Misalkan terdapat tiga buah piksel yang mempunyai nilai [3 5 7]. Perubahan nilai graylevel ketiga titik tersebut adalah (7-3)/2 = 2. Jika ketiga titik tersebut dilakukan proses konvolusi dengan mask [-1 0 1] akan didapat nilai -3+7 = 4. Dari sini dapat dilihat bahwa hasil yang diperoleh mempunyai nilai lebih besar untuk piksel yang berada di posisi batas tepi.

D. Range Filter

Range filter dapat digunakan untuk mendeteksi batas tepi suatu citra. Output range filter diperoleh dari perbedaan antara nilai minimum dan maksimum dari piksel yang berada dalam lokal window.

Range filter dapat didefinisikan seperti di bawah ini :

$$\text{Range}(C_i) = \max(C_i) - \min(C_i) \qquad i = 1,2,3,\ldots,N$$

C_i adalah piksel dalam lokal window. C_1 adalah piksel di pojok kiri atas, dan seterusnya sampai dengan C_N adalah piksel di pojok kanan bawah. Sedangkan N adalah jumlah piksel dalam lokal window.

E. Homogeneity Operator Edge Detector

Metode *Homogeneity Operator* menggunakan operasi pengurangan untuk mendeteksi batas tepi. Output metode ini diperoleh dari nilai maksimum diantara nilai absolut pengurangan antara piksel yang berada di tengah-tengah lokal window 3x3 dengan delapan buah piksel di sekelilingnya.

Contoh : lokal window

5 7 6
1 2 4
9 3 8

Output = max{ |2-5|, |2-7|, |2-6|, |2-1|,
|2-4|, |2-9|, |2-3|, |2-8| }
= max{ 3, 5, 4, 1, 2, 7, 1, 6 }
= 7

F. Difference Operator Edge Detector

Metode *Difference Operator* hampir sama dengan metode Homogeneity Operator, yaitu menggunakan operasi pengurangan. Outputnya didapat dari nilai maksimum perbedaan antara piksel-piksel dalam lokal window 3x3, yaitu nilai absolut piksel di kiri atas dikurangi piksel di kanan bawah, piksel di kanan atas dikurangi piksel di kiri bawah, piksel di atas dikurangi piksel di bawah, dan piksel di kiri dikurangi piksel di kanan.

Contoh : lokal window

5 7 6
1 2 4
9 3 8

Output = max{ |5-8|, |6-9|, |7-3|, |1-4| }
= max{ 3, 3, 4, 3 }
= 4

G. Contrast-based Edge Detector

Satu masalah yang sering dihadapai dalam deteksi batas tepi adalah pencahayaan yang tidak merata dalam suatu citra. *Contrast-based Edge Detector* membantu mengatasi masalah ini. Citra yang pencahayaannya baik mempunyai batas tepi dengan perbedaan graylevel yang besar. Jika batas tepi tersebut berada berada dalam citra dengan pencahayaan yan kurang baik akan mempunyai perbedaan graylevel yang lebih kecil.

Proses yang dilakukan Contrast-based Edge Detector adalah melakukan proses deteksi batas tepi suatu lokal window dengan menggunakan salah satu metode deteksi batas tepi. Output yang dihasilkan dibagi dengan nilai rata-rata area. Pembagian ini untuk meratakan pencahayaan dalam suatu citra. Yang dimaksud dengan nilai rata-rata area adalah hasil konvolusi antara lokal window dengan mask yang semua elemennya berupa angka satu dan dibagi dengan jumlah piksel dalam lokal window.

Contoh :

Quick mask				Mask		
-1	0	-1		1	1	1
0	4	0	1/9 *	1	1	1
-1	0	-1		1	1	1

Batas tepi dengan pencahayaan baik :

50 100 100
50 100 100
50 100 100

Konvolusi dengan Quick mask:

400-50-50-100-100 = 100

Konvolusi dengan Mask :

50+50+50+100+100+100+100+100+100 / 9 = 750/9 = 83

Output = 100/83 = 1

Batas tepi dengan pencahayaan kurang baik :

5 10 10
5 10 10
5 10 10

Konvolusi dengan Quick mask:

40-5-5-10-10 = 10

Konvolusi dengan Mask :

5+5+5+10+10+10+10+10+10 / 9 = 75/9 = 8

Output = 10/8 = 1