

MODUL PRAKTIKUM

PENGOLAHAN CITRA DIGITAL

NAMA MAHASISWA

NIM MAHASISWA

lABORATORIUM TEKNIK ELEKTRO

jURUSAN tEknik ELEKTRO

FAKULTAS TEKNik

universitas muhammadiyah malang

201

*Halaman ini sengaja dikosongkan*

# STANDART OPERASIONAL PROSEDUR

**LABORATORIUM TEKNIK ELEKTRO**

**UNIVERSITAS MUHAMMADIYAH MALANG**

1. **PRA PRAKTIKUM**
2. Ka Laboratorium bersama Ketua Prodi menetapkan daftar Mata Praktikum yang akan dilaksanakan pada semester berjalan
3. Laboran atau Staf mengumumkan daftar Mata Praktikum dan pengumuman lainnya via web lab-elektro.umm.ac.id
4. Staf / Laboran menerima pendaftaran calon praktikan yang mengulang
5. Staf / Laboran mengumumkan daftar peserta Mata Praktikum berdasarkan data peserta mata kuliah dan peserta mengulang di web lab-elektro.umm.ac.id
6. Kepala lab dan wakil kepala lab menetapkan daftar Instruktur dan Asisten Mata Praktikum dan diusulkan untuk ditetapkan SK Dekan
7. Ka. Lab mengundang Peserta Mata Praktikum untuk mengikuti pertemuan persiapan dan pembagian jadwal peserta mengikuti praktikum dan peraturan serta prosedur praktikum dan K3
8. Instruktur dan Asisten mengundang peserta Mata Praktikum untuk mengikuti Ujian Pra Praktikum (Memberikan Tugas Pra Praktikum)

1. **PRA PELAKSANAAN PERCOBAAN PRAKTIKUM**
   * + 1. Asisten dan Praktikan hadir 15 menit sebelum dimulai jam praktikum
       2. Asisten mempersiapkan instrumen ukur serta modul praktikum dan peralatan pendukung seperti kabel, jumper dan lain lain
       3. Praktikan membaca petunjuk praktikum dan mempersiapkan kebutuhan peralatan sebelum masuk ruang/lab
       4. Asisten memberikan salam dan ucapan selamat datang dengan senyum serta memberikan arahan kepada kelompok Praktikan tentang prosedur pelaksanaan praktikum dan penjelasan daftar peralatan dan modul
       5. Asisten menunjuk peserta yang menjadi petugas pencatat, melakukan pengukuran dan pembantu pelaksanaan
       6. Asisten meminta kelompok Praktikum untuk membaca doa/Basmalah sebelum dimulai pemasangan dan instalasi praktikum dan dipandu oleh Asisten

1. **PRAKTIKUM BERLANGSUNG**
2. Asisten memberikan instruksi kepada kelompok praktikan pemasangan atau instalasi modul dan mengawasi dan mengevaluasi serta memeriksa hasil pemasangan dan memastikan kebenaran instalasi
3. Praktikan dan asisten saling menjaga kenyamanan dan ketertiban praktikum sesuai tata tertib yang berlaku serta menjaga keamanan perangkat lab selama pelaksanaan praktikum dari satu percobaan ke percobaan berikutnya.
4. Asisten berhak menegur dan menindak praktikan apabila ketahuan merusak, mengubah atau memindahkan perlengkapan lab tanpa ijin.
5. Asisten melakukan penilaian dan pengawasan tiap praktikan melakukan pengukuran selama percobaan.
6. Asisten dan kelompok praktikan mengakhiri praktikum dengan membaca hamdallah dan mengucap salam serta meminta praktikan untuk merapikan peralatan dan modul serta kursi dan membuang sampah di sekitarnya.

1. **PRAKTIKUM BERAKHIR**
2. Praktikan meninggalkan ruangan dengan rapi dan teratur.
3. Asisten Mengkondisikan ruangan kembali,
   1. Mengembalikan/mengatur kursi kembali.
   2. Merapikan sampah yang ditemukan berserakan dalam ruangan.
   3. Mengembalikan peralatan dan modul ke Lemari Alat dan Modul sesuai nama jenis Mata Praktikum
   4. Mengunci pintu
   5. Mematikan lampu apabila tidak ada praktikum berikutnya.
4. Asisten menandatangani presensi kelompok dan memberikan daftar penilaian kerja percobaan kelompok ke ruang administrasi (Laboran).
5. Instruktur dan atau asisten melakukan evaluasi reguler praktikum jika diperlukan.

1. **PASCA PRAKTIKUM**

Praktikan menyusun laporan semua percobaan

Praktikan melakukan asistensi laporan ke Asisten Praktikum min 4 kali

Setelah laporan praktikum ditandatangani oleh Asisten, Tiap Praktikum menghadap Instruktur sesuai jadwal yang ditetapkan Instruktur

Instruktur menguji praktikum mengenai proses pelaksanaan praktikum

Instruktur memberikan nilai akhir praktikan

Nilai akhir prakatikum diserahkan ke Lab untuk proses administrasi

1. **SANKSI**
2. Keterlambatan asistensi pertama kali sanksi point 1
3. Tidak memenuhi minimal 4 kali asistensi sanksi point 2
4. Datang terlambat 15 menit dari waktu yang telah ditentukan sanksi point 3
5. \* Tidak mengikuti proses praktikum tanpa adanya konfirmasi sanksi point 4
6. \* Tidak mengikuti ujian koordinator tanpa adanya konfirmasi sanksi point 5
7. Keterlambatan pengumpulan laporan resmi sanki point 6
8. \* Tidak mengikuti ujian instruktur sesuai dengan jadwal yang ditentukan instruktur sanksi point 7
9. Pemalsuan tanda tangan selama proses praktikum berlangsung sanksi point 8
10. Merusakkan perlatan Lab. Teknik Elektro sanksi point 9

\* Maksimal konfirmasi 2 x 24 jam sejak jadwal resmi diumumkan untuk penggantian jadwal ujian

|  |  |
| --- | --- |
| Point 1 | Menulis materi modul bab 1 |
| Point 2 | Menulis materi modul bab 1-3 & Pengurangan Nilai |
| Point 3 | Menulis materi 1 bab & Pengurangan Nilai |
| Point 4 | Mengulang (tidak konfirmasi sesuai waktu yang telah ditentukan) atau Pengurangan Nilai |
| Point 5 | Mengulang (tidak konfirmasi sesuai waktu yang telah ditentukan) atau Pengurangan Nilai |
| Point 6 | Membeli buku berkaitan dengan bidang Teknik elektro |
| Point 7 | Pengurangan Nilai Instruktur |
| Point 8 | Mengulang Praktikum atau mendapat Nilai E |
| Point 9 | Mengganti peralatan tersebut sesuai dengan spesifikasi atau mirip dan memiliki fungsi yang sama |

1. **KESELAMATAN DAN KESEHATAN KERJA (K3)**
2. Sebelum memulai praktikum, praktikan memahami tata tertib dan keselamatan di Laboratorium
3. Mengetahui tempat dan cara penggunaan perlatan Laboratorium
4. Memperhatikan dan waspada terhadap tempat-tempat sumber listrik ( stop kontak dan circuit breaker)
5. Praktikan harus memperhatikan dan menaati peringatan (warning) yang biasa tertera pada badan perlatan praktikum maupun rambu peringatan yang terdapat di ruangan Laboratorium
6. Jika melihat ada kerusakan yang berpotensi menimbulkan bahaya, segera laporkan ke asisten terkait atau dapat langsung melapor ke laboran.
7. Hindari daerah atau benda yang berpotensi menimbulkan bahaya listrik ( sengatan listrik) secara tidak sengaja, missal seperti jala-jala kabel yang terkelupas
8. Keringkan bagian tubuh yang basah, seperti keringat atau sisa air wudhu
9. Selalu waspada terhadap bahaya listrik pada setiap aktifitas praktikum.
10. Jika terjadi kecelakaan akibat bahaya listrik, berikut ini adalah hal-hal yang harus diikuti praktikan:
11. Jangan panik
12. Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing dan di meja praktikum yang tersengat arus listrik.
13. Bantu praktikan yang tersengat arus listrik untuk melepaskan diri dari sumber listrik
14. Beritahukan dan minta bantuan kepada laboran, praktikan lain dan orang di sekitar anda tentang terjadinya kecelakaan akibat bahaya listrik.
15. Jangan membawa benda-benda mudah terbakar (korek api, gas, dll) ke dalam ruangan

laboratorium bila tidak disyaratkan dalam modul praktikum.

1. Jangan melakukan sesuatu yang menimbulkan api, percikan api, atau panas yang berlebihan.
2. Jangan melakukan sesuatu yang menimbulkan bahaya api atau panas berlebih pada diri sendiri atau orang lain.
3. Selalu waspada terhadap bahaya api atau panas berlebih pada setiap aktivitas di laboratorium.
4. Jika terjadi kecelakaan akibat bahaya listrik, berikut ini adalah hal-hal yang harus diikuti praktikan:
5. Jangan panik
6. Matikan semua peralatan elektronik dan sumber listrik di meja masing-masing.
7. Beritahukan dan minta bantuan laboran, praktikan lain dan orang di sekitar anda tentang terjadinya bahaya api atau panas berlebih
8. Menjauh dari ruang praktikum
9. Dilarang membawa benda tajam (pisau, gunting dan sejenisnya) ke ruang praktikum bila tidak diperlukan untuk pelaksanaan percobaan
10. Dilarang memakai perhiasan dari logam misalnya cincin, kalung, gelang, dll
11. Hindari daerah, benda atau logam yang memiliki bagian tajam dan dapat melukai.

Tidak melakukan sesuatu yang dapat menimbulkan luka pada diri sendiri atau orang lain.

*Halaman ini sengaja dikosongkan*

# PENGOLAHAN CITRA DIGITAL MENGGUNAKAN MATLAB

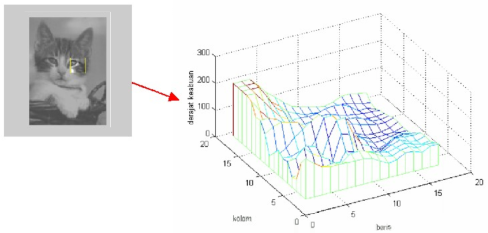
## Pendahuluan

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

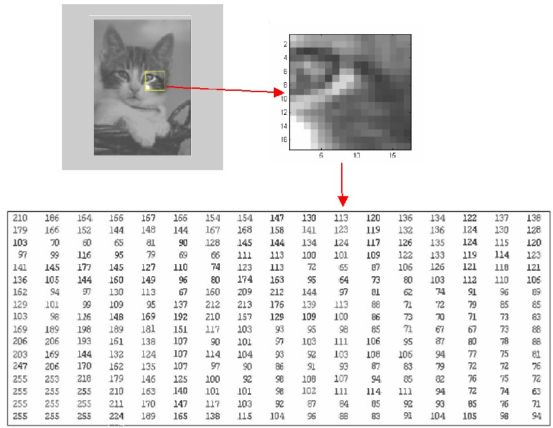
## Model Citra Digital

Citra atau gambar digital adalah citra yang diambil berdasarkan sampling dan kuantisasi tertentu sehingga citra digital ini terbentuk dari piksel-piksel yang besarnya tergantung pada besar kecilnya sampling dan nilainya (besarnya derajat keabuan) tergantung pada kuantisasi. Berdasarkan pengertian ini maka model citra digital dinyatakan dalam bentuk matrik yang masing-masing kolom dan barisnya berisikan berupa nilai yang merupakan derajat keabuan yang mewakili pada satu piksel seperti terlihat pada Gambar 1.2.

Model ini menyatakan model dari citra gray-scale yaitu citra yang terdiri dari derajat keabuan tertentu. Dengan kata lain dapat dinyatakan bahwa citra digital adalah citra yang didefinisikan sebagai fungsi f(x,y) dimana x menyatakan nomor baris, y menyatakan nilai kolom, dan f menyatakan nilai derajat keabuan dari citra. Sehingga (x,y) adalah posisi dari piksel dan f adalah nilai derajat keabuan pada titik (x,y) seperti terlihat pada



Gambar 1.1 Representasi Citra Keabuan dalam kurva keabuan



Gambar 1.2 Representasi Citra Digital Keabuan (*Gray Scale*) dengan Matriks

Representasi model pada Gambar 1.1 dan Gambar 1.2 adalah representasi dari citra gray-scale (menggunakan derajat keabuan). Untuk citra berwarna, terdapat beberapa model yang disebut sebagai Kanal Warna (*Color Space*). Pada citra berwarna untuk membentuk suatu warna dalam suatu piksel, Kanal kanal warna yang masing masing memiliki nilai tertentu tersebut dicampur/digabungkan. Terdapat beberapa model kanal warna yang digunakan untuk membentuk citra yakni, model RGB (*Red-Green­Blue*), HSV(*Hue-Saturation-Value*), YUV (Y adalah “*Luma*”/”*Luminance*”(kecerahan);UV(Informasi Warna)) dan YCbCr (Y adalah “*Luma*”/”*Luminance*”(kecerahan);Cb(*Blue*),Cr(*Red*)). Pada Gambar 1.3 merupakan salah satu contoh dimana satu citra berwama dinyatakan sebagai 3 buah matrik yang berupa satu matrik untuk Red (R-layer), satu matrik untuk Green (G-layer) dan matrik lainya untuk Blue (B-layer).



Gambar 1.3 Representasi Citra berwarna dengan model RGB

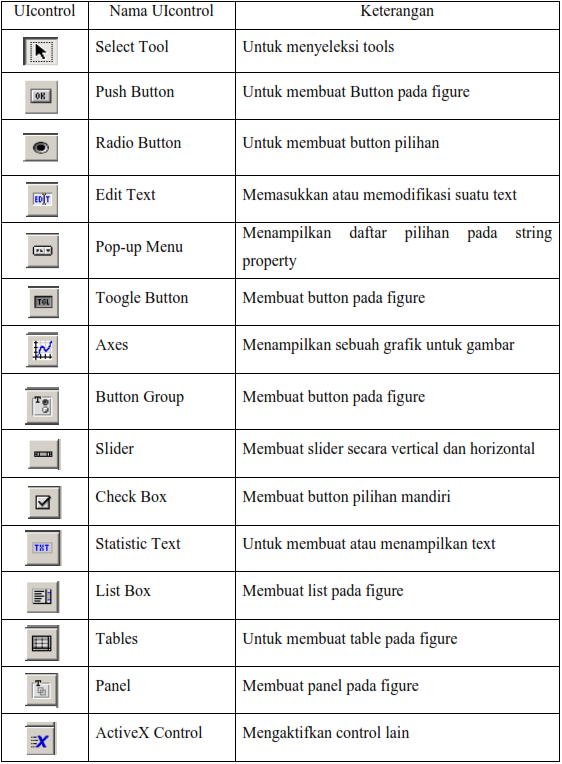
## Pengolahan Citra dengan Matlab

MATLAB (Matrik Labority) merupakan perangkat lunak produksi dari The Math Work, Inc. MATLAB adalah sebuah bahasa dengan (hight-performance) kinerja tinggi untuk komputasi masalah teknik. Matlab mengintegrasikan komputasi, visualisasi, dan pemrograman dalam suatu model yang sangat mudah untuk pakai dimana masalah-masalah dan penyelesaiannya diekspresikan dalam notasi matematika yang familiar.

### GUI (Graphic User Interface) Matlab

Tampilan GUI terdapat berbagai komponen yang terdiri dari beberapa uicontrol (kontrol user interface), seperti pada bahasa pemrograman visual lainnya, yaitu: *pushbutton, togglebutton, radiobutton, chexkboxes, edit text, static text, slider, frames, listboxes, popup menu*, dan *axes*. Kita dapat meletakkan semua kontrol pada layout editor dan selanjutnya hanya tinggal mengaturnya melalui property inspector. Semua kontrol pada GUI dapat dimunculkan pada *layout/figure* dengan cara mendrag and drop kontrol yang diinginkan ke *figure*. Adapun penjelasan fungsi masing masing kontrol pada GUI Matlab dijabarkan pada Tabel 1.1.

Tabel 1.1 Fungsi Kontrol Pada GUI Matlab



### Image Loading, Displaying, Saving

1. Imread

Gambar dimuat ke dalam memori kerja menggunakan perintah :

>> f = imread(“rice.jpg”);

Tanda titik koma dalam Matlab diakhir perintah digunakan untuk menghilangkan output (tidak ditampilkan). Jika tidak disertakan, maka Matlab akan menampilkan output ke layar. Penyebutan nama file tampa path akan mengasumsikan bahwa file tersebut berada di current working directory (directory kerja). Selain itu pembacaan file juga dapat dilakukan dengan path relative. Perhatikan contoh berikut :

>> f = imread(“F:\citra\ rice.jpg”);

1. Imshow

Untuk menampilkan gambar dapat digunakan perintah berikut :

>> imshow(f)

Dimana F adalah variable yang akan ditampilkan tetapi dengan syarat F harus sudah di open terlebih dahulu.

1. Figure

Gambaran pada umumnya ditampilkan didalam figure window. Jika suatu gambaran ditampilkan akan overwrite dulu, kecali jika fungsi figure digunakan :

>> figure, imshow(f);

akan menghasilkan suatu jendela figure baru dan memanjang gambardidalamnya.

1. Imwrite

Gambar dapat disimpan ke disk menggunakan fungsi imwrite, yang memiliki fungsi dasar :

>> imwrite(array name,'file name')

String yang menjadi isi parameter nama file harus memasukkan estensi format file yang dikenali. Alternatifnya, format yang diinginkan dapat disebutkan secara eksplisit pada masukkan parameter ketiga. Tempat penyimpanan fungsi ini (lokasi file output) ada di derektori kerja. Contoh dibawah ini adalah untuk menyimpan gambar i ke TIFT file dengan nama rice\_01:

>> imwrite(i ,'rice\_01.tif');

1. Image Information

Informasi tentang sebuah gambar dapat ditemukan dengan :

>> iminfo filename

1. Clc

clc digunakan untuk membersihkan semua yang di input dan di outputkan dari Command Window display

>> clc

# OPERASI DASAR PENGOLAHAN CITRA

Pada bab 1 kita telah ketahui bahwa citra digital merupakan susunan dari beberapa matriks (array) untuk membentuk citra tersebut. Dimana secara matematika, gambar dapat didefinisikan sebagai fungsi , di mana x dan y adalah koordinat spasial (piksel), dan amplitudo fungsi output di setiap pasangan x dan y disebut intensitas. Sebagai contoh pada Gambar 2.1 (a) merupakan gambar *grayscale* dari sebuah objek dan apabila ditampilkan dalam matriks adalah seperti Gambar 2.1 (b). dan ketika diakses menggunakan per pixel Contoh :149.

|  |  |
| --- | --- |
| (a) | (b) |

Gambar 2.1 (a).Gambar Grayscale (b).Matriks gambar a

Karena gambar tersusun dari beberap matriks maka operasi aritmatika dan logika dapat di terapkan untuk memanipulasi hasil gambar. Beberapa operasi aritmatika yang dapat digunakan memanipulasi gambar antara lain, penjumlahan, pengurangan, perkalian, dan pembagian. Operasi tersebut dilakukan pada masing masing kolom dan baris yang bersesuaian.

## Operasi Artimatika

### Operasi Penjumlahan

Operasi penjumlahan dilakukan dengan menjumlahkan masing masing kolom dan baris pada dua buah gambar yang dijumlahkan. Pada operasi penjumlahan gambar yang perlu diperhatikan adalah ukuran pixel gambar yang dijumlahkan harus sama, kemudian jenis gambar juga harus sama sehingga memiliki jumlah kolom dan baris yang sama. Operasi penjumlahan ditunjukan pada persamaan (2.1).

|  |  |
| --- | --- |
|  | (2.1) |

Langkah-langkah Percobaan:

1. Buatlah source code berikut :

|  |
| --- |
| clear all;  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage1 = imread([pathname filename]);  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage2 = imread([pathname filename]);  resultImage = sourceImage1+sourceImage2;  figure(1)  imshow(sourceImage1)  figure(2)  imshow(sourceImage2)  figure(3)  imshow(resultImage) |

1. Run Program Tersebut
2. pilih gambar 1 (image1.jpg)
3. Pilih gambar 2 (image2.jpg)
4. Screen shoot hasil percobaan
5. Berikan Analisa

### Operasi Pengurangan

Operasi penguraangan dilakukan dengan mengurangkan masing masing kolom dan baris pada yang dikurangkan. Pada operasi pengurangan gambar sama halnya dengan operasi sebelumnya yang perlu diperhatikan adalah ukuran pixel gambar yang dikurangkan harus sama, kemudian jenis gambar juga harus sama sehingga memiliki jumlah kolom dan baris yang sama. Operasi pengurangan ditunjukan pada persamaan(2.2).

|  |  |
| --- | --- |
|  | (2.2) |

Langkah Percobaan:

1. Buatlah source code berikut :

|  |
| --- |
| clear all;  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage1 = imread([pathname filename]);  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage2 = imread([pathname filename]);  resultImage = sourceImage1-sourceImage2;  figure(1)  imshow(sourceImage1)  figure(2)  imshow(sourceImage2)  figure(3)  imshow(resultImage) |

1. Run Program Tersebut
2. pilih gambar 1 (image1.jpg)
3. Pilih gambar 2 (image2.jpg)
4. Screen shoot hasil percobaan
5. Berikan Analisa

### Operasi Perkalian

Operasi Perkalian dilakukan dengan mengalikan masing masing kolom dan baris pada yang dikalikan. Pada operasi perkalian gambar sama halnya dengan dua operasi sebelumnya yang perlu diperhatikan adalah ukuran pixel gambar yang dikalikan juga harus sama, kemudian jenis gambar juga harus sama sehingga memiliki jumlah kolom dan baris yang sama. Operasi perkalian ditunjukan pada persamaan (2.3).

|  |  |
| --- | --- |
|  | (2.3) |

1. Langkah Percobaan: Buatlah source code berikut :

|  |
| --- |
| clear all;  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage1 = imread([pathname filename]);  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage2 = imread([pathname filename]);  resultImage = sourceImage1.\*sourceImage2;  figure(1)  imshow(sourceImage1)  figure(2)  imshow(sourceImage2)  figure(3)  imshow(resultImage) |

1. Run Program Tersebut
2. pilih gambar 1 (image1.jpg)
3. Pilih gambar 2 (image2.jpg)
4. Screen shoot hasil percobaan
5. Berikan Analisa

### Operasi Pembagian

Operasi pembagian dilakukan dengan membagi nilai pada masing masing kolom dan baris. Pada operasi pembagian gambar sama halnya dengan beberapa operasi sebelumnya yang perlu diperhatikan adalah ukuran pixel gambar yang dijumlahkan juga harus sama, kemudian jenis gambar juga harus sama sehingga memiliki jumlah kolom dan baris yang sama. Operasi perkalian ditunjukan pada persamaan (2.4).

|  |  |
| --- | --- |
|  | (2.4) |

Langkah Percobaan:

1. Buatlah source code berikut :

|  |
| --- |
| clear all;  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage1 = imread([pathname filename]);  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage2 = imread([pathname filename]);  resultImage = sourceImage1./sourceImage2;  figure(1)  imshow(sourceImage1)  figure(2)  imshow(sourceImage2)  figure(3)  imshow(resultImage) |

1. Run Program Tersebut
2. pilih gambar 1 (image1.jpg)
3. Pilih gambar 2 (image2.jpg)
4. Screen shoot hasil percobaan
5. Berikan Analisa

## Operasi Logika (Citra Biner)

Pada gambar, khususnya pada jenis gambar biner dapat dilakukan operasi logika berupa operasi And, Or, dan Not. Seperti halnya pada operasi aritmatika operasi logika juga dilakukan secara masing masing kolom dan baris. Perintah perintah operasi logika dapat dilihat pada table berikut:

|  |  |
| --- | --- |
| Operasi Logika | Perintah Matlab |
| And | and(ImageP,ImageQ) |
| Or | or(ImageP,ImageQ) |
| Xor | xor(ImageP,ImageQ) |
| Not | not(ImageQ) |
| Nand | not(and(ImageP,ImageQ)) |

Langkah Percobaan:

1. Buatlah source code berikut :

|  |
| --- |
| clear all;  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage1 = imread([pathname filename]);  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage2 = imread([pathname filename]);  resultImage = and(sourceImage1,sourceImage2);  figure(1)  imshow(sourceImage1)  figure(2)  imshow(sourceImage2)  figure(3)  imshow(resultImage) |

1. Run Program Tersebut
2. pilih gambar 1 (Biner1.jpg)
3. Pilih gambar 2 (Biner2.jpg)
4. Screen shoot hasil percobaan
5. Berikan Analisa
6. Lanjutkan Untuk operasi Or, Xor, Nand, Nor, dan Xnor
7. Dan berikan Analisa

## Histogram Citra

*Histogram Citra* dapat didefinisikan sebagai banyaknya jumlah piksel dalam sebuah citra yang memiliki nilai intensitas yang sama tanpa memperhitungkan dimana posisi piksel itu berada. Pada citra *grayscale,* kurva histogram menggambarkan distribusi tingkat kecerahan/keabuan citra mulai dari yang gelap hingga yang paling terang.

|  |  |
| --- | --- |
| (a) | (b) |

Gambar 2.2 (a) Gambar (b) Histogram Gambar

Pada Gambar 2.2 (a) merupakan salah satu contoh gambar dengan kurva histogram ditunjukan pada Gambar 2.2 (b) dimana sumbu X merupakan nilai intensitas gambar mulai dari yang paling gelap (0) hingga yang paling terang (255)

### Percobaan 2.2

Pada percobaan ini bagaiman cara menampilkan suatu kurva histogram sebuah citra

Langkah percobaan:

1. Buatlah Source Code sebagai berikut:
2. Run Program tersebut

|  |
| --- |
| clear  clc  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  sourceImage1 = imread([pathname filename]);  figure(1)  imshow(sourceImage1);  figure(2)  imhist(sourceImage1(:,:,1)); |

1. Pilih Gambar (Image1.jpg)
2. Simpan Hasil Histogram
3. Analisa Hubungan gambar dengan kurva histogramnya

### Histogram Kumulatif & Histogram Equalization

**Histogram kumulatif** merupakan penjumlahan nilai histogram citra yang kemudian di normalisasi sehingga bernilai anatara 0 hingga 1 (prosentase jumlah pixel pada intensitas tertentu). Histogram Kumulatif digunaakan untuk menghitung Histogram Equalization dimaa histogram kumulatif dinyatakan dengan persamaan

|  |  |
| --- | --- |
|  | (2.5) |

dengan N dan M adalah ukuran kolom dan baris citra dalam pixel, H adalah histogram citra dan L adalah jumlah data pada Histogram citra.

**Histogram Equalization** merupakan cara yang digunakan untuk meningkatkan kecerahan citra berdasarkan Histogram Kumulatifnya. Peningkatan kecerahan dilakukan dengan cara merubah nilai keabuan pada citra sehingga didapatkan nilai keabuan yang lebih merata.histogram equalization dinyatakan dengan persamaan

|  |  |
| --- | --- |
|  | (2.6) |

dimana adalah histogram kumulatif dan I adalah citra asal dan adalah nilai maksimum dari tingkat keabuan yaitu 255. Pada matlab perintah histogram equalization dilakukan dengan fungsi *histeq(I).*

### Percobaan 2.3

Langkah Percobaan:

1. Buatlah Source Code sebagai berikut:

|  |
| --- |
| clear all;  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  I = imread([pathname filename]);  I=rgb2gray(I);  figure('NumberTitle', 'off', 'Name', 'Gambar Citra Asli');  imshow(I);  [N,M]=size(I);  Histo=imhist(I);  figure('NumberTitle', 'off', 'Name', 'Histogram Citra Asli');  plot(Histo);  xlabel('Nilai Intensitas')  ylabel('Jumlah Pixel')    [L] = size(Histo);  Hist\_kum(1) = Histo(1);  for l = 2 : L,  Hist\_kum(l) = Histo(l) + Hist\_kum(l-1);  end  Hist\_kum=Hist\_kum/(N\*M);    figure('NumberTitle', 'off', 'Name', 'Histogram Kumulatif Citra Asli');  plot(Hist\_kum);  xlabel('Nilai Intensitas')  ylabel('Jumlah Pixel %')    for n = 1 : N  for m = 1 : M  Im\_equal(n,m) = uint8(round( Hist\_kum(I(n,m)+1) \*255));  end  end  figure('NumberTitle', 'off', 'Name', 'Citra Hasil Equalisasi');  imshow(uint8(Im\_equal));    I\_equal = histeq(I); %citra equalization dengan fungsi histeq() MATLAB  figure('NumberTitle', 'off', 'Name', 'Citra Equalisasi dengan Histeq');  imshow(I\_equal);    Histo\_eq=imhist(Im\_equal);  figure('NumberTitle', 'off', 'Name', 'Histogram Hasil Equalisasi');  plot(Histo\_eq);  xlabel('Nilai Intensitas')  ylabel('Jumlah Pixel') |

1. Run Program tersebut
2. Pilih Gambar (Image1.jpg)
3. Simpan Hasil Percobaan
4. Analisa gambar sebelum dan sesudah equalisasi berdasarkan kurva histogramnya

## Konvolusi Citra

Operasi konvolusi sering digunakan dalam pengolahan citra seperti *filtering,* *masking,* dan transformasi. Konvolusi dilakukan dengan 2 buah vector atau dua matriks berukuran sama baris dan kolomnya (persegi) dengan jumlah baris kolom adalah ganjil. Operasi konvolusi diskrit pada vector 2 dimensi dijabarkan pada persamaan

|  |  |
| --- | --- |
|  | (2.7) |

contoh operasi konvolusi pada pada matriks 1 dimensi adalah sebagai berikut:

|  |
| --- |
| Contoh:  Dimana  Sehingga |

Contoh beberapa tahapan konvolusi 2 dimensi:

|  |
| --- |
|  |

Pada matlab perintah konvolusi 2 dimensi dilakukan dengan perintah sebagai berikut

dimana dan adalah citra yang dikonvolusikan dengan *same* adalah metode konvolusi dengan hasil koncolusi memiliki kolom dan baris yang sama dengan citra asli.

### Percobaan 2.4

Langkah percobaan

1. Buatlah Source Code sebagai berikut:

|  |
| --- |
| clear all;  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  I = imread([pathname filename]);  K=3; % Set ukuran matriks Kernel filter 3x3  F=ones(K,K)/(K\*K) ;  figure(1)  imshow(I);  [N,M,L]=size(I);  for p=1:L  Konv(1:N,1:M,p)=conv2(double(I(:,:,p)), F, 'same');  end  figure(2)  imshow(uint8(Konv)); |

1. Run Program tersebut
2. Pilih Gambar (Image1.jpg)
3. Simpan Hasil Percobaan
4. Ubah Nilai kernel K untuk nilai 3,5,7,9,11
5. Analisa gambar sebelum dan sesudah konvolusi berdasarkan nilai kernel

## Morfologi Citra

Morfologi matematis merupakan operasi matematis berdasarkan teori geometri dan himpunan. Proses morflogi secara umum terbagi menjadi dua jenis proses yaitu Dilasi dan Erosi, Kedua proses tersebut digunakan untuk proses segmentasi citra, image enhancement dan edge detection pada citra. Proses morfologi cendrung digunakan untuk citra *grayscale* terlebih untuk citra biner.

### Dilasi

Dilasi adalah proses matematis yang digunakan untuk memperbesar objek pada citra biner. Operasi dilasi ditunjukan dengan persamaan

|  |  |
| --- | --- |
|  | (2.8) |

Dimana adalah citra asli dan adalah matriks kernel untuk dilasi apabila pada masing masing kolom dan baris yang bersesuaian antara dan memiliki nilai sama dengan satu maka hasil dilasi akan bernilai 1 dan apabila sama sama bernilai 0 atau berbeda maka akan bernilai 0.

dan maka

dan maka

dan maka

dan maka

Dari bebrapa contoh diatas maka operasi dilasi dapat diartikan dengan operasi OR untuk keseluruhan dan operasi AND untuk masing masing kolom dan baris

### Erosi

Erosi adalah proses kebalikan dari dilasi yang digunakan untuk memperkecil objek pada citra biner. Operasi dilasi ditunjukan dengan persamaan

|  |  |
| --- | --- |
|  | (2.8) |

Dimana adalah citra asli dan adalah matriks kernel. Hasil erosi akan bernilai 1 hanya jika pada masing masing kolom dan baris yang bersesuaian antara dan memiliki nilai sama dengan satu maka hasil dilasi

dan maka

dan maka

dan maka

dan maka

Dari bebrapa contoh diatas maka operasi dilasi dapat diartikan dengan operasi AND untuk keseluruhan dan operasi AND untuk masing masing kolom dan baris

### Opening & Closing

Operasi Opening & Closing adalaj operastor penting dalam proses morfologi. Operator *Opening* digunakan untuk menghilangkan objek objek kecil yang dianggap noise dimana besar kecilnya noise yang dihilangkan bergantung pada besar ukuran matriks kernel . Sebaliknya *Closing* digunakan untuk menutup celah antara dua objek atau menutup lubang pada sebuah objek. Sama seperti opening besar objek yang ditutup bergantung pada besar ukuran kernel . Kedua Operator dinyatakan dalam persamaan berikut:

|  |  |
| --- | --- |
|  | (2.10) |
|  | (2.11) |

Proses closing dilakukan dengan melakukan dilasi terhadap kernel kemudian dilakukan erosi terhadap kernel. Sedangkan proses opening dilakukan dengan erosi terlebih dahulu baru kemudian dilakukan dilasi terhadap kernel

### Percobaan 2.5

Langkah langkah Percobaan

1. Buatlah Source Code sebagai berikut:

|  |
| --- |
| [filename,pathname] = uigetfile({'\*.jpg;\*.bmp'},'Select an Image');  I = imread([pathname filename]); % Baca file citra format dan simpan pada matriks I  K=5; % Set ukuran matriks filter 3x3  Bs=ones(K,K); % Set matriks filter rata-rata F  figure(1), imshow(I); % tampilkan citra I  dilasi=imdilate(I,Bs);  erosi=imerode(I,Bs);  Closing=imerode(dilasi,Bs);  Opening=imdilate(erosi,Bs);  %Closing=imclose(I,Bs);  %Opening=imclose(I,Bs);  figure(2),  imshow(uint8(dilasi));  figure(3),  imshow(uint8(erosi));  figure(4),  imshow(uint8(Closing));  figure(5),  imshow(uint8(Opening)); |

1. Run Program tersebut
2. Pilih Gambar (Biner4.jpg)
3. Simpan Hasil Percobaan
4. Ubah Nilai kernel K untuk nilai 3,5,7,9,11
5. Analisa gambar sebelum dan sesudah berdasarkan nilai kernel

# DETEKSI TEPI & SEGMENTASI CITRA

## Deteksi Tepi

Pendeteksian tepi adalah proses penting dalam memilih atau memisahkan sebuah objek dalam gambar. Setiap objek tentu memiliki range warna atau tektur tertentu sehingga untuk memisahkan objek satu dengan yang lain perludiketahui batasan atau tepi antara satu objek dengan yang lain.

### Gradient Citra

Secara matematis, perbedaan intensitas nilai dari piksel ke piksel dilakukan dengan turunan pertama dari suatu fungsi. Berdasarkan konsep tersebut untuk mendeteksi tepi diperlukan nilai turunan dari setiap tepi. Dimana turunan parsial pertama dari sebuah matriks dijabarkan pada persamaan (3.1) , nilai gradient dan magnitudenya pada persamaan (3.2) dan (3.3)

|  |  |
| --- | --- |
|  | (3.1) |
|  | (3.2) |
|  | (3.3) |

Berdasarkan persamaan tersebut untuk menghitung nilai gradient dalam piksel ketetanggaan didapatkan dengan persamaan

|  |  |
| --- | --- |
|  | (3.4) |
|  | (3.5) |
|  | (3.6) |

### Operator Roberts

Pada operasi gradien arah turunan pertama dari fungsi gradien menggunakan arah vertical dan horizontal. Pada gradien x diturunkan terhadap sumbu x secara horizontal dan pada gradien y diturunkan secara vertical. Pada operator gradien robrt cara menurunkan nya menggunakan sudut dan sehingga persamaannya menjadi

|  |  |
| --- | --- |
|  | (3.4) |
|  | (3.5) |
|  | (3.6) |

### Operator Sobel & Prewitt

Operator sobel dan perwit selain menghitung nilai gradien menggabungkan dengan menambahkan filter perata untuk menghilangkan nosie. Pemfilteran dilakukan dengan arah yang berlawanan dengan arah turunan pertama dari gradien. Turuna pertama dari operator sobel dan prewitt dilakukan dengan titik dan serta dan dan ditambahkan dengan operator filter dengan persamaan keseluruhan sebagai berikut

|  |  |
| --- | --- |
|  | (3.10) |
|  | (3.11) |

dengan dan adalah operator filter perata prewitt serta dan adalah operator filter perata Sobel dimana

### Percobaan 3.1

Langkah langkah Percobaan

1. Buatlah Source Code sebagai berikut:

|  |
| --- |
| [filename,pathname] = uigetfile({'\*.jpg;\*.bmp;\*.png'},'Select an Image');  I = imread([pathname filename]);  [Gx1,Gy1]=gradient(double(I));  G1=abs(Gx1)+abs(Gy1);  GS1=imgradient(I,'sobel');  GR1=imgradient(I,'roberts');  GP1=imgradient(I,'prewitt');  figure('NumberTitle', 'off', 'Name', 'Gambar Citra Asli');  imshow(I);  figure('NumberTitle', 'off', 'Name', 'Citra Hasil Gradient');  imshow(G1);  figure('NumberTitle', 'off', 'Name', 'Citra Hasil Roberts');  imshow(GR1);  figure('NumberTitle', 'off', 'Name', 'Citra Hasil Sobel');  imshow(GS1);  figure('NumberTitle', 'off', 'Name', 'Citra Hasil Prewitt');  imshow(GP1); |

1. Run Program tersebut
2. Pilih gambar ImageEdgeGray.jpg
3. Simpan Hasil Percobaan
4. Analisa gambar berdasarkan tipe operator

## Segmentasi

Segmentasi merupakan memisahkan objek objek yang berada pada warna dan tektur tertentu sesuai dengan referensi warna yang ditentukan untuk dipisahkan.

### Percobaan 3.2

1. Buatlah Source Code sebagai berikut:

|  |
| --- |
| clear;  clc;  [filename,pathname] = uigetfile({'\*.jpg;\*.bmp;\*.png'},'Select an Image');  Im = imread([pathname filename]); % Baca file citra format dan simpan pada matriks I  figure(1), imshow(uint8(Im));title('Citra Asli');  Th\_HSV=0.5; % tentukan nilai threshold 0.2 0.3 0.4  Th\_RGB=5;% tentukan nilai threshold 2 3 4  p = impoint(gca,[]); % pembacaan posisi pixel  p = wait(p); % klik 2 kali pada warna area  x=round(p(1,1)); % yang ingin disegmentasi  y=round(p(1,2));  RGB=Im(y,x,:); % warna referensi ruang RGB  R=RGB(:,:,1);  G=RGB(:,:,2);  B=RGB(:,:,3);  % Konversi warna pixel referensi  HSV=rgb2hsv(RGB);  H1=HSV(:,:,1); % warna referensi ruang HSV  S1=HSV(:,:,2);  V1=HSV(:,:,3);  % Konversi Citra ke ruang warna HSV dan HCL  citra\_hsv=rgb2hsv(Im);  [m,n,l]=size(citra\_hsv); % ukuran citra  % Inisialisasi matriks citra hasil  cit\_hasilHSV(1:m,1:n,1:l)= uint8(0);  cit\_hasilRGB(1:m,1:n,1:l)= uint8(0);    for i=1:m  for j=1:n  % hitung jarak warna RGB  dR=(abs(Im(i,j,1)-R))^2;  dG=(abs(Im(i,j,2)-G))^2;  dB=(abs(Im(i,j,3)-B))^2;  check=sqrt(double(dR+dG+dB));  if (check<=Th\_RGB) % warna similar?  cit\_hasilRGB(i,j,:)= Im(i,j,:);  end  % hitung jarak warna HSV  dH=citra\_hsv(i,j,1)-H1;  S2=citra\_hsv(i,j,2);  dV=(citra\_hsv(i,j,3)-V1)^2;  Dcyl=sqrt(double((dV +(S1^2)+(S2^2)) - (2\*S1\*S2\*cos(dH))));  if (Dcyl <=Th\_HSV) % warnanya similar?  cit\_hasilHSV(i,j,:)= Im(i,j,:);  end  end;  end;    % Tampilkan citra hasil  figure, imshow(cit\_hasilHSV);  figure, imshow(cit\_hasilRGB); |

1. Run Program tersebut
2. Pilih gambar brain\_tumor\_fluorescent.jpg
3. Arahkan kursor pada tumor (warna merah)
4. Double Click
5. Simpan Hasil Percobaan
6. Ubah nilai threshold rgb dengan nilai 4, 3, dan 2 serta threshold hsv 0.4, 0.3, 0.2
7. Analisa gambar hasil segmentasi

*Halaman ini sengaja dikosongkan*