

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Citra**

##### **2.1.1 Pengertian Citra**

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek. Citra merupakan output alat perekaman, seperti kamera yang bersifat analog maupun digital. Citra analog berupa sinyal-sinyal video seperti gambar pada televisi, sedangkan citra digital dapat langsung disimpan pada suatu media penyimpanan (Sutoyo dkk, 2009)

##### **2.1.2 Pengertian Citra Digital**

Citra digital merupakan sebuah larik (array) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu. Citra yang disimpan dalam memori komputer hanyalah angka-angka yang menunjukkan besar intensitas pada masing-masing piksel tersebut (Putra,2010)

#### **2.2 Pengolahan Citra**

Di dalam bidang komputer, ada 3 bidang studi yang berkaitan dengan citra, namun tujuan ketiganya berbeda, yaitu :

1. Grafika Komputer (Computer Graphics)
2. Pengolahan Citra (Image Processing)

### 3. Pengenalan Pola (Pattern Recognition/Image Interpretation)

Tetapi kebanyakan citra tidak sesuai yang diharapkan, salah satu faktor adanya cacat (noise) saat pengambilan gambar. Maka, proses pengolahan citra sangat diperlukan dalam kehidupan sehari-hari. Tujuan dari pengolahan citra digital sebagai berikut (Sutoyo dkk, 2009):

1. Memperbaiki kualitas gambar dilihat dari aspek radiometrik (peningkatan kontras, transformasi warna, restorasi citra) dan dari aspek geometrik (rotasi, translasi, skala, transformasi geometrik).
2. Melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra.
3. Melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data.

Proses citra, khususnya dengan menggunakan komputer akan menghasilkan hasil yang lebih baik dari sebelumnya. Berikut adalah alur dari pengolahan citra:

#### **Citra Asli Proses → Pengolahan Citra → Citra Hasil**

Pada umumnya, operasi-operasi pengolahan citra diterapkan pada citra apabila:

1. Perbaiki atau modifikasi citra untuk meningkatkan kualitas visual atau menonjolkan beberapa aspek informasi yang terkandung dalam citra.
2. Elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur.
3. Sebagian citra perlu di gabung dengan bagian citra yang lain.

Pengolahan citra juga mempunyai manfaat-manfaat pada bidang tertentu, antara lain:

1. Bidang perdagangan
  - a. Pembacaan barcode
  - b. Pengenalan huruf atau angka pada suatu formulir secara otomatis
2. Bidang militer
  - a. Mengenali sasaran peluru kendali melalui sensor visual.
  - b. Mengidentifikasi jenis pesawat musuh
3. Bidang kedokteran
  - a. Mammografi
  - b. Rekonstruksi foto janin hasil USG
4. Bidang biologi
  - a. Pengenalan jenis kromosom melalui citra mikroskopik
5. Komunikasi data
  - a. Kompresi citra yang akan ditransmisikan
6. Hiburan
  - a. Game
  - b. Kompresi video
7. Hukum
  - a. Pengenalan sidik jari
  - b. Pengenalan foto narapidana

## **2.3 Jenis Citra**

Nilai suatu piksel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum, jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Secara umum jangkauannya adalah 0-255. Berikut adalah jenis-jenis citra berdasarkan nilai pikselnya

### **2.3.1 Citra Biner**

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner hanya membutuhkan satu bit untuk mewakili nilai setiap piksel dari citra biner(Darma,2010)

### **2.3.2 Citra Grayscale**

Citra grayscale adalah citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian red = green = blue. Citra grayscale memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan)(Darma,2010)

### **2.3.3 Citra Warna (8 bit)**

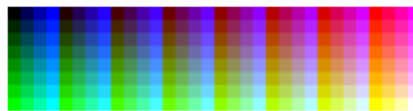
Setiap piksel dari citra warna (8 bit) dengan jumlah warna maksimum yang dapat digunakan adalah 256 warna. Setiap titik (piksel) pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar yaitu merah, hijau, dan biru yang biasa disebut citra RGB (Red, Green, Blue). Ada dua jenis citra warna 8 bit. Pertama, citra warna 8 bit dengan menggunakan palet warna 256 dengan setiap paletnya memiliki pemetaan nilai (colormap) RGB tertentu (Darma,2010). Model ini lebih

sering digunakan. Kedua, setiap piksel memiliki format 8 bit sebagai berikut:

Tabel 2.1 Citra 8 Bit

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
R	R	R	G	G	G	B	B

Bentuk kedua dinamakan 8 bit truecolor. Dapat dilihat pada gambar dibawah ini :



Gambar 2.1 8 Bit Truecolor

### 2.3.4 Citra Warna (16 bit)

Citra warna 16 bit biasanya disebut sebagai citra highcolor. Setiap pikselnya diwakili dengan 2 byte memory (16 bit). Warna 16 bit memiliki 65.536 warna. Dalam formasi bitnya, nilai merah dan biru mengambil tempat di 5 bit di kanan dan kiri. Komponen hijau memiliki 5 bit ditambah 1 bit ekstra. Pemilihan komponen hijau dengan deret 6 bit dikarenakan penglihatan manusia lebih sensitive terhadap warna hijau (Darma,2010)

Tabel 2.2 Citra 16 Bit

Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit
t	t	t	t	t	t	9	8	7	6	5	4	3	2	1	0
15	14	13	12	11	10										
R	R	R	R	R	G	G	G	G	G	G	B	B	B	B	B

Gambar dibawah ini menampilkan deret warna yang dihasilkan dari warna 16 bit.



Gambar 2.2 16 Bit Highcolour

### 2.3.5 Citra Warna (24 bit)

Setiap piksel dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini tentunya sudah sangat memvisualisasikan seluruh warna yang dapat dilihat oleh penglihatan manusia. Setiap poin informasi piksel (RGB) disimpan ke dalam 1 byte data. 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai hijau pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah.



Gambar 2.3 Citra Warna 24 bit

## 2.4 Citra Warna

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang yang berbeda (Fatmawati, 2011). Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu mempunyai panjang gelombang paling rendah.

Citra warna adalah citra dengan sistem grafik yang memiliki satu set nilai tersusun yang menyatakan berbagai tingkat warna (Siregar, 2009). Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = Red Green Blue).

Manusia punya 5-7 juta sel yang dibagi menjadi tiga kategori sensor (merah, hijau, dan biru). 65% sel kerucut sensitif pada warna merah, 33% pada warna hijau, dan 2% pada warna biru. Pada karakteristik mata manusia warna terlihat seperti kombinasi warna variabel yang disebut primer, yaitu merah (red), hijau (green), dan biru (blue). Warna primer dapat digunakan untuk menghasilkan warna sekunder.

1. Magenta = merah + biru

2. Cyan = hijau + biru

3. Kuning = merah + hijau

Warna-warna yang diterima oleh mata merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Kombinasi warna yang memberikan rentang warna yang paling lebar adalah red(R), green(G) dan blue(B) dan

warna bukan merupakan besaran fisik tetapi warna merupakan suatu sensasi yang dihubungkan dengan sistem saraf kita, seperti halnya rasa maupun bau. Sensasi warna diperoleh dengan adanya interaksi antara warna dengan sistem saraf sensitive warna kita.

Sistem RGB digabungkan untuk memperoleh warna tertentu. Misalnya warna putih diperoleh dari hasil gabungan warna merah = 255, hijau = 255, dan biru = 255. Dalam sistem RGB, warna putih cerah dinyatakan dengan RGB (255, 255, 255). Nilai dari setiap primer adalah 0 sampai 255. Sehingga kemungkinan warna yang didapat adalah  $256 \times 256 \times 256$  yakni kurang lebih 16.7 juta warna. Pada tabel 2.1 akan diperlihatkan beberapa hasil penggabungan kode warna RGB.

Tabel 2.3 Kode Warna RGB

Colour	Red	Green	Blue
Black	0	0	0
Blue	0	0	255
Green	0	255	0
Cyan (Blue + Green)	0	255	255
Red	255	0	0
Magenta (Red+Blue)	255	0	255
Yellow (Red+Green)	255	255	0
White (Red+Green+Blue)	255	255	255
Gray	128	128	128



## 2.5 CIELab

Warna adalah deskriptor kuat dalam segmentasi citra yang menyederhanakan identifikasi objek dan ekstraksi dari gambar (Rasmana, 2013). Model warna memfasilitasi spesifikasi warna dengan cara yang standar. Sebuah subruang sebuah model warna memberikan satu titik untuk mewakili warna.

*CIELab* adalah salah satu struktur warna yang didefinisikan *CIE (Comission International de l'Eclairage/The International Commission on Illumination* pada tahun 1976 (*CIE 1976 L\*a\*b\**). Pada *CIELab*, besaran *CIE\_L\** untuk mendeskripsikan kecerahan warna, 0 untuk hitam dan  $L^* = 100$  untuk putih. Dimensi *CIE\_a\** mendeskripsikan jenis warna hijau – merah, dimana angka negatif  $a^*$  mengindikasikan warna hijau dan sebaliknya  $CIE_a^*$  positif mengindikasikan warna merah. Dimensi *CIE\_b\** untuk jenis warna biru – kuning, dimana angka negatif  $b^*$  mengindikasikan warna biru dan sebaliknya  $CIE_b^*$  positif mengindikasikan warna kuning. Transformasi RGB (*Red, Green, Blue*) – *CIELab* dapat dilakukan dengan berikut ( Gernot Hoffmann,2010):

$$C=CR.....(2.1)$$

Dimana

$$C=C'G.....(2.2)$$

G= 2.2. dan C= R, G, B

$$X1=X/Xn.....(2.3)$$

$$Y1=Y/Yn.....(2.4)$$

$$Z1=Z/Zn.....(2.5)$$

$$X1 = \{X11/3 \text{ jika } X1 > 0.0088567.787X1 + 16116 \dots \dots \dots (2.6)$$

$$Y1 = \{Y11/3 \text{ jika } Y1 > 0.0088567.787Y1 + 16116 \dots \dots \dots (2.7)$$

$$Z1 = \{Z11/3 \text{ jika } Z1 > 0.0088567.787Z1 + 16116 \dots \dots \dots (2.8)$$

Sehingga  $L^*a^*b^*$  menjadi:

$$L^* = 116Y1 - 16 \dots \dots \dots (2.9)$$

$$a^* = 500(X1 - Y1) \dots \dots \dots (2.10)$$

$$b^* = 200(Y1 - Z1) \dots \dots \dots (2.11)$$

Hasil uji menggunakan *CIELab* pada penelitian sebelumnya dari disertasi Rasmana pada tahun 2013 didapatkan nilai rata-rata untuk  $\Delta a^*$  adalah 0,84, dan rata-rata untuk  $\Delta b^*$  1.4. Hal ini menunjukkan bahwa perbedaan warna huruf prasasti untuk komponen warna  $\Delta a^*$  dan  $\Delta b^*$  tersebut adalah kecil. Perbedaan yang cukup besar ada pada unsur intensitas warna  $\Delta L^*$  yang bernilai 11,47. Demikian pula dari hasil segmentasi prasasti logam, dari pengukuran dengan metode RMSE didapatkan nilai error rata-rata untuk segmentasi dengan komponen intensitas warna  $L^*$  paling kecil 116,57, dibandingkan dengan komponen  $a^*$  dan  $b^*$  yang bernilai 215,79 dan 205,81. Berdasarkan hasil ini maka dikembangkan ke proses peningkatan kualitas citra prasasti logam berdasarkan fitur tekstur. Tekstur yang merupakan variasi spasial intensitas dari citra juga merupakan karakteristik dari suatu objek. Dari karakteristik tekstur inilah diharapkan dapat dideteksi pahatan huruf dan selanjutnya diproses untuk menjadi lebih jelas.

## 2.6. Segmentasi Citra

Segmentasi merupakan teknik untuk membagi suatu citra menjadi beberapa daerah (*region*) dimana setiap daerah memiliki kemiripan atribut (Darma,P.2010).Pengolahan citra digital merupakan sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar(rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri(*feature images*) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra,melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. *Input* dari pengolahan citra adalah citra, sedangkan *output*-nya adalah citra hasil pengolahan.

Segmentasi citra merupakan bagian dari proses pengolahan citra. Proses segmentasi citra ini lebih banyak merupakan suatu proses pra pengolahan pada sistem pengenalan objek dalam citra. Segmentasi citra (*image segmentation*) mempunyai arti membagi suatu citra menjadi wilayah-wilayah yang homogen berdasarkan kriteria keserupaan yang tertentu antara tingkat keabuan suatu piksel dengan tingkat keabuan piksel-piksel tetangganya, kemudian hasil dari proses segmentasi ini akan digunakan untuk proses tingkat tinggi lebih lanjut yang dapat dilakukan terhadap suatu citra, misalnya proses klasifikasi citra dan proses identifikasi objek. Adapun dalam proses segmentasi citra itu sendiri terdapat beberapa algoritma, diantaranya :

algoritma Deteksi Titik, Deteksi Garis, dan Deteksi Sisi (berdasarkan Operator Robert dan Operator Sobel).

Metode segmentasi citra dikelompokkan menjadi dua bagian, yaitu metode segmentasi *low level* dan *high level*. Metode segmentasi *low level* misalnya *mean shift*, *watershed*, *level set*, dan *super pixel* biasanya membagi gambar menjadi beberapa daerah kecil. Meskipun hasil segmentasinya sering over segmentation, namun metode segmentasi *low level* memberikan dasar yang bagus untuk operasi segmentasi *high level* selanjutnya. Metode segmentasi *high level* misalnya region merging, graph cut (Ning dkk 2009).

## 2.7. Segmentasi Citra Berbasis *Clustering*

Segmentasi citra berbasis *clustering* menggunakan data multidimensi untuk mengelompokkan piksel citra ke dalam beberapa *clustering*. Data multidimensi pada citra ini maksudnya adalah banyaknya atribut atau komponen penyusun suatu citra, misalnya citra grayscale mempunyai satu buah dimensi, citra RGB mempunyai tiga buah dimensi, dan sebagainya. Pada umumnya piksel di-*clustering* berdasarkan kedekatan jarak antar piksel. Segmentasi berbasis *clustering* ini mulai populer sejak diimplementasikan pada aplikasi OCR (*Optical Character Recognition*), pengenalan sidik jari hingga *remote sensing*. Keberhasilan dari proses segmentasi berbasis *clustering* ini ditentukan dari keberhasilan dalam mengelompokkan fitur-fitur yang berdekatan ke dalam satu *cluster* (Darma, 2010).

Metode-metode dalam segmentasi berbasis *clustering* di antaranya adalah metode iterasi, K-Means, Fuzzy C-means, jaringan syaraf Kohonen, dan berbagai teknik cluster lainnya. Salah satu metode yang sangat baik digunakan untuk segmentasi citra adalah Fuzzy C-Means *clustering*. Fuzzy C-Means ini merupakan algoritma K-means yang diimprovisasi dengan *Fuzzy Set Theory* dengan menerapkan derajat keanggotaan, dimana satu piksel citra dapat dimiliki oleh beberapa cluster. “*Soft*” *clustering* ini memberikan komputasi yang lebih tepat dalam menentukan keanggotaan dari cluster.

## 2.8. Algoritma Fuzzy C-Means

Fuzzy C-Means adalah suatu teknik *clustering* (pengelompokan) data di mana keberadaan titik-titik data dalam suatu *cluster* ditentukan oleh derajat keanggotaan. Penentuan titik *cluster* dilakukan secara berulang-ulang hingga diperoleh data yang akurat berdasarkan derajat keanggotaannya. Perulangan ini didasarkan pada minimalisasi fungsi obyektif yang menggambarkan jarak dari titik data ke pusat *cluster* yang terbobot oleh derajat keanggotaan. Akibat adanya derajat keanggotaan tersebut, maka suatu titik data bisa dimiliki lebih dari satu kelompok. Metode ini merupakan minimasi dari fungsi obyektif (B. Sowmya.2010)

Algoritma Fuzzy C Means pertama kali diperkenalkan oleh Dunn (1974), kemudian dikembangkan oleh Bezdek (1981), kemudian direvisi oleh Rouben (1982), Trauwert (1985), Goth dan Geva (1989), Gu dan Gubuisson (1990), Xiedan Beni (1991). Namun, algoritma FCM dari Bezdek yang paling

banyak digunakan. Fuzzy C-Means adalah salah satu teknik pengelompokan data yang mana keberadaan tiap titik data dalam suatu kelompok (*cluster*) ditentukan oleh derajat keanggotaan. Berbeda dengan k-means *clustering*, dimana suatu objek hanya akan menjadi anggota satu *cluster*, dalam Fuzzy C-Means setiap objek dapat menjadi anggota dari beberapa *cluster*. Batas-batas dalam k-means adalah tegas (*hard*) sedangkan dalam Fuzzy C-Means adalah *soft*. Fuzzy C-Means bersifat sederhana, mudah diimplementasikan, memiliki kemampuan untuk mengelompokkan data yang besar, lebih kokoh terhadap data *outlier*. Langkah-langkah algoritma Fuzzy C-Means adalah sebagai berikut:

1) Peng-input-an data yang akan di *cluster* X, berupa matriks berukuran

$n \times m$ .

dengan :

$n$  : jumlah data *sample*

$m$  : atribut setiap data

$X_{ij}$  : data *sample* ke- $i$  ( $i = 1, 2, \dots, n$ ), atribut ke- $j$  ( $j = 1, 2, \dots, m$ )

2) Menentukan:

a. Jumlah *cluster* :  $c$

b. Pangkat :  $m$

c. Maksimum iterasi :  $MaxIter$

d. *Error* terkecil yang diharapkan :  $\square$

e. Fungsi obyektif awal :  $P_0 = 0$

f. Iterasi awal :  $t = 1$

3) Membangkitkan nilai acak  $\mu_{ik}$ ,  $i=1,2,\dots,n$ ;  $k=1,2,\dots,c$ ; sebagai elemen-elemenmatriks partisi awal  $u$ .  $\mu_{ik}$  adalah derajat keanggotaan yangmerujuk pada seberapa besar kemungkinan suatu data bisa menjadianggota ke dalam suatu *cluster*. Posisi dan nilai matriks dibangun secararandom. Dimana nilai keanggotaan terletak pada interval 0 sampai dengan

1. Pada posisi awal matriks partisi  $U$  masih belum akurat begitu juga pusatclusternya. Sehingga kecendrungan data untuk masuk suatu *cluster* jugabelumakurat.Selanjutnya menghitung jumlah setiap kolom (atribut)

$$Q_j = \sum_{k=1}^c \mu_{ik} \quad (2.1)$$

$Q_j$  adalah jumlah nilai derajat keanggotaan perkolom = 1 dengan  $j=1,2,\dots,m$

Selanjutnya dilakukan perhitungan sebagai berikut :

$$\mu_{ik} = \frac{\mu_{ik}}{Q_j} \quad (2.2)$$

4) Menghitung pusat *cluster* ke- $k$  :  $V_{kj}$ , dengan  $k=1,2,\dots,c$ ; dan  $j=1,2,\dots,m$ .

$$V_{kj} = \frac{\sum_{i=1}^n ((\mu_{ik})^m \cdot X_{ij})}{\sum_{i=1}^n (\mu_{ik})^m} \quad (2.3)$$

- 5) Menghitung fungsi obyektif pada iterasi ke- $t$ ,  $P_t$ . Fungsi obyektif digunakan sebagai syarat perulangan untuk mendapatkan pusat *cluster* yang tepat. Sehingga diperoleh kecendrungan data untuk masuk ke *cluster* mana pada langkah akhir.

$$P_t = \sum_{i=1}^n \sum_{k=1}^c ([\sum_{j=1}^m (X_{ij} - V_{kj})^2] (\mu_{ik})^m) \quad (2.4)$$

- 6) Menghitung perubahan matriks partisi:

$$\mu_{ik} = \frac{[\sum_{j=1}^m (X_{ij} - V_{kj})^2]^{\frac{-1}{m-1}}}{\sum_{k=1}^c [\sum_{j=1}^m (X_{ij} - V_{kj})^2]^{\frac{-1}{m-1}}} \quad (2.5)$$

- 7) Mengecek kondisi berhenti:

Jika:

$$(|P_t - P_{t-1}| < \zeta)$$

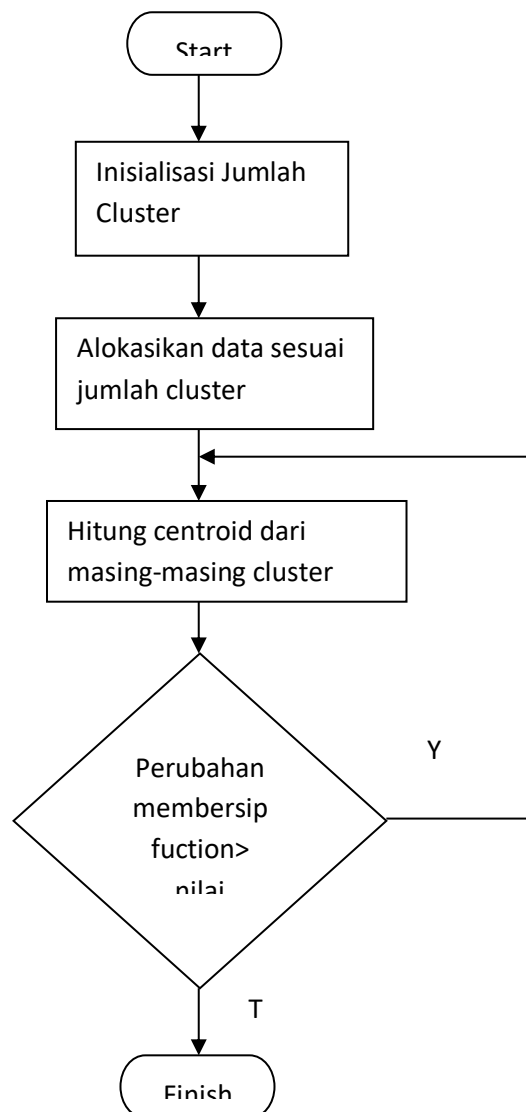
else

$$(t > \text{maxIterasi})$$

maka berhenti;

Jika tidak:  $t = t + 1$ , dilakukan pengulangan langkah ke-4.





Gambar 2.4 Bagan Umum Algoritma Fuzzy C-Means

## 2.9 K-Means

Algoritma K-means merupakan salah satu algoritma dengan partitional, karena K-Means didasarkan pada penentuan jumlah awal kelompok dengan mendefinisikan nilai centroid awalnya (Madhulatha, 2012). Algoritma K - means menggunakan proses secara berulang-ulang. untuk mendapatkan basis data cluster. Dibutuhkan jumlah cluster awal yang diinginkan sebagai

masuk dan menghasilkan titik centroid akhir sebagai output. Metode K-means akan memilih pola k sebagai titik awal centroid secara acak atau random. Jumlah iterasi untuk mencapai cluster centroid akan dipengaruhi oleh calon cluster centroid awal secara random. Sehingga didapat cara dalam pengembangan algoritma dengan menentukan centroid cluster yang dilihat dari kepadatan data awal yang tinggi agar mendapatkan kinerja yang lebih tinggi (HUNG dkk. 2005 ; Eltibi & Ashour, 2011).

Dalam penyelesaiannya, algoritma K-Means akan menghasilkan titik centroid yang dijadikan tujuan dari algoritma K-Means. Setelah iterasi K-Means berhenti , setiap objek dalam dataset menjadi anggota dari suatu cluster. Nilai cluster ditentukan dengan mencari seluruh objek untuk menemukan cluster dengan jarak terdekat ke objek. Algoritma K -means akan mengelompokan item data dalam suatu dataset ke suatu cluster berdasarkan jarak terdekat (Bangoria et al., 2013). Nilai centroid awal yang dipilih secara acak yang menjadi titik pusat awal, akan dihitung jarak dengan semua data menggunakan rumus Euclidean Distance. Data yang memiliki jarak pendek terhadap centroid akan membuat sebuah cluster. Proses ini berkelanjutan sampai tidak terjadi perubahan pada setiap. kelompok (Agrawal & Gupta, 2013 ; Chaturved & Rajavat, 2013).

### **2.9.1 Keuntungan Algoritma K-Means**

Algoritma K-Means juga memiliki keuntungan yaitu :

1. Dalam implementasi menyelesaikan masalah, algoritma K-Means sangat simple serta fleksibel. Artinya perhitungan komputasinya tidak

terlalu rumit dan algoritma ini dapat diimplementasikan pada segala bidang.

2. Algoritma K-Means sangat mudah untuk dipahami, terutama dalam implementasi data yang sangat besar serta dapat mengurangi kompleksitas data yang dimiliki (Bangoriadkk 2013)

### **2.9.2 Kelemahan Algoritma K-Means**

Kelemahan yang dimiliki oleh algoritma K-Means yaitu :

1. Di Algoritma K-Means user memerlukan angka yang tepat dalam menentukan jumlah cluster sebanyak k karena terkadang pusat cluster awal dapat berubah sehingga kejadian ini bisa mengakibatkan pengelompokan data menjadi tidak stabil (Joshi & Nalwade, 2013)
2. Algoritma K-Means tidak bisa maksimal dalam menentukan atau menginisialisasi nilai centroid awalnya, karena pada pengelompokan data dengan algoritma K-Means sangat bergantung pada nilai centroidnya (Ahmed & Ashour, 2011).
3. Output dari K-Means tergantung pada nilai – nilai pusat yang dipilih pada clustering. Sehingga pada algoritma ini nilai awal titik pusat cluster menjadi dasar dalam penentuan cluster. Pemilihan centroid cluster awal secara acak akan memberikan pengaruh terhadap kinerja cluster tersebut (Sujatha & Sona, 2013)

### **2.9.3 Algoritma K-Means**

Berikut ini langkah-langkah yang terdapat pada algoritma K-Means (Ediyanto dkk 2013) :

1. Tentukan k sebagai jumlah cluster yang dibentuk Untuk menentukan banyaknya cluster k dilakukan dengan beberapa pertimbangan seperti pertimbangan teoritis dan konseptual yang mungkin diusulkan untuk menentukan berapa banyak cluster. Dapat dihasilkan dari perbandingan hasil SSE (Sum of Squared Error) dengan rumus SSE seperti dibawah ini (Irwanto dkk, 2012):

$$SSE = \sum_{k=1}^K \sum_{x_i \in S_k} \|x_i - C_k\|^2$$

Dimana  $X_i$  menyatakan norma euclid ( $L_2$ ) dan  $C_k$  adalah pusat kluster  $S_k$  yang dihitung berdasarkan rata-rata jarak titik-titik kluster ke pusat kluster.

2. Tentukan k centroid (titik pusat cluster) awal secara random. Penentuan centroid awal dilakukan secara random/acak dari objek-objek yang tersedia sebanyak k cluster, kemudian untuk menghitung centroid cluster ke-i berikutnya, digunakan rumus sebagai berikut :

$$v = \frac{\sum_{i=1}^n x_i}{n} \quad ; i = 1, 2, 3, \dots, n$$

Dimana; v : centroid pada cluster

xi : objek ke-i

$n$  : banyaknya objek/jumlah objek yang menjadi anggota cluster

3. Hitung jarak setiap objek ke masing-masing centroid dari masing-masing cluster. Untuk menghitung jarak antara objek dengan centroid dapat menggunakan Euclidian Distance

$$d(x, y) = \|x - y\| = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad ; i = 1, 2, 3, \dots, n \quad 2.8$$

Dimana;  $x_i$  : objek  $x$  ke- $i$

$y_i$ : daya  $y$  ke- $i$

$n$  : banyaknya objek

4. Alokasikan masing-masing objek ke dalam centroid yang paling dekat. Untuk melakukan pengalokasian objek kedalam masing-masing cluster pada saat iterasi secara umum dapat dilakukan dengan cara hard kmeans dimana secara tegas setiap objek dinyatakan sebagai anggota cluster dengan mengukur jarak kedekatan sifatnya terhadap titik pusat cluster tersebut.
5. Lakukan iterasi, kemudian tentukan posisi centroid baru dengan menggunakan persamaan (2.7)
6. Ulangi langkah 3 jika posisi centroid baru tidak sama

### 2.10. Preprocessing

Data Preparation atau bisa disebut juga dengan data *preprocessing* adalah suatu proses/langkah yang dilakukan untuk membuat data mentah menjadi

data yang berkualitas(input yang baik untuk data mining tools)(Handkk,2011)..Tahapan *Data Preprocessing* terbagi menjadi:

### ***Data Preprocessing***

Pada bagian ini menyajikan gambaran dari *data preprocessing*. Pada bagian *data quality*, mengilustrasikan banyak unsur yang menentukan kualitas data. Ini memberikan insentif balik bagi *Data preprocessing* dan selanjutnya menguraikan tugas utama dalam *data preprocessing* .

### ***Data Cleaning***

Pembersihan data (atau *data cleansing*) ber-upaya untuk mengisi nilai-nilai yang hilang, menghaluskan *noisy data*, mengidentifikasi *outlier* , dan inkonsistensi yang benar dalam data.

### ***Data Integration***

Integrasi data merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru. Tidak jarang data yang diperlukan untuk *data mining* tidak hanya berasal dari satu *database* tetapi juga berasal dari beberapa *database* atau *file teks*. Integrasi data dilakukan pada atribut-aribut yang mengidentifikasikan entitas-entitas yang unik seperti atribut nama, jenis produk, nomor pelanggan dan lainnya. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya. Sebagai contoh bila integrasi data berdasarkan jenis produk ternyata menggabungkan produk dari

kategori yang berbeda maka akan didapatkan korelasi antar produk yang sebenarnya tidak ada.

### ***Data Reduction***

*Data Reduction* berguna untuk mendapatkan pengurangan representasi dari kumpulan data yang jauh lebih kecil di dalam volume tetapi belum menghasilkan hasil yang sama (atau hampir sama) dari suatu hasil analisis (Han dkk, 2011).

## **2.11 CBIR**

CBIR atau *Content Based Image Retrieval* merupakan suatu teknik untuk mencari suatu gambar dengan membandingkan gambar query dengan gambar yang ada di database (*Query By Example*) (Ramadijanti, 2006). Secara lebih jelas, *Content Based Image Retrieval* dapat didefinisikan sebagai temu kembali citra dalam suatu basis data dengan cara pengindeksan citra menggunakan karakteristik visual citra seperti warna, tekstur dan bentuk (*content-based*) (Herdiyeni dan Widiyanto, 2007). Ada 3 modul utama dalam pencarian citra berbasis isi (*contentbased*), yaitu:

1. ekstraksi fitur
2. pengindeks-an multidimensi
3. pencarian

Sebuah citra dapat direpresentasikan dengan vektor multidimensi dari fitur citra. Sebuah vektor fitur dapat diasosiasikan sebagai sebuah titik dalam ruang multi dimensi. Fitur citra yang terekstrak disimpan sebagai metadata

dan citra diindeks berdasarkan informasi metadata. Informasi metadata dapat berisi beberapaukuran dari fitur-fitur citra yang terekstrak (Kusrini, 2006).

Suatu gambar memiliki ciri yang berbeda satu dan yang lainnya bergantung pada karakteristik yang menonjol pada gambar tersebut. Contoh, bunga matahari dan bunga melati dibedakan melalui perbedaan warnanya, kain dan kertas dapat dibedakan dari teksturnya, sedangkan gambar lingkaran dengan gambar kotak dibedakan melalui bentuknya. Masing-masing ciri dasar dari gambar ini didapatkan melalui proses ekstraksi ciri yang tidak mudah, karena satu gambar dapat mempunyai multiple feature. Proses ekstraksi ciri yang baik menentukan keberhasilan dalam membangun aplikasi image/citra (Ramadijanti, 2006).

## **2.12 Ekstraksi Ciri**

Ekstraksi ciri merupakan salah satu tahapan dalam sistem CBIR yang berguna untuk mendapatkan karakteristik visual citra (Herdiyeni dan Widiyanto, 2007). Ekstraksi ciri adalah proses mengambil ciri-ciri yang terdapat pada citra. Ciri-ciri tersebut adalah besaran komponen tertentu dari citra objek yang mewakilisifat utama citra objek, sekaligus mengurangi dimensi citra objek menjadisekumpulan bilangan yang lebih sedikit tetapi representative (Syahputra, 2009).

Tujuan utama dari proses ekstraksi ciri adalah untuk mengkarakterisasi objek yang ingin dikenali dari sebuah citra dengan menggunakan ukuran-ukuran yang memiliki nilai sangat mirip untuk objek pada kategori yang



sama dan sangat berbeda untuk objek pada kategori yang tidak sama. Artinya, dilakukan pencarian terhadap ciri-ciri dari objek yang membedakannya dengan objek yang lain dan ciri-ciri tersebut tidak akan berubah (*invariant*) meskipun ada pengaruh transformasi yang tidak relevan terhadap citra (Duda et al., 2001). Proses ekstraksi ciri yang baik menentukan keberhasilan dalam membangun aplikasi image (Ramadijanti, 2006).

### 2.12.1 Ciri Bentuk

Pada dasarnya temu kembali citra berdasarkan bentuk merupakan pengukuran kesamaan antara bentuk-bentuk yang direpresentasikan oleh fitur-fiturnya. Ciri bentuk ada yang dihitung berdasarkan region/area dan ada yang dihitung berdasarkan contour.

#### a. Ciri bentuk berdasarkan area

Ciri bentuk yang direpresentasikan oleh area dapat ditentukan dengan menggunakan fitur geometri. Beberapa fitur geometri sederhana dapat digunakan untuk menggambarkan bentuk (Mingqiang et al., 2008). Lima geometri dasar antara lain:

1. Diameter (D)
2. *Physiological Length* (Lp)
3. *Physiological Width* (Wp)
4. Luas Area (A)
5. *Perimeter* (P)

Beberapa ciri morfologi digital yang sering digunakan diturunkan dari limageometri dasar (Wu et al., 2007), antara lain:

1. *Aspect ratio* ( $L_p/W_p$ )

Perbandingan *Physiological Length* ( $L_p$ ) dan *Physiological Width* ( $W_p$ )

2. *Rectangularity* ( $L_p W_p / A$ )

Bentuk citra dibandingkan dengan persegi empat

3. *Form factor* ( $4 \sqrt{A} / P$ )

*Form factor* disebut juga *Roundness*. Bentuk citra dibandingkan dengan lingkaran. Nilai fitur ini tidak terpengaruh oleh adanya rotasi, translasi dan skala pada citra (Hsu et al., 2010)

4. *Narrow factor* ( $D/L_p$ )

5. *Perimeter ratio of Diameter* ( $P/D$ )

6. *Perimeter ratio of Physiological Length* ( $L_p$ ) dan *Physiological Width* ( $W_p$ ): ( $P/L_p + W_p$ )

Parameter lainnya yang dapat digunakan untuk membedakan bentuk suatu objek yaitu '**metric**'. Metric merupakan nilai perbandingan antara luas dan keliling objek. Metric memiliki rentang nilai antara 0 hingga 1. Objek yang berbentuk memanjang/mendekati bentuk garis lurus, nilai metricnya mendekati angka 0, sedangkan objek yang berbentuk bulat/lingkaran, nilai metricnya mendekati angka 1. Dengan rumus sebagai

berikut

$$M = \frac{4\pi \times A}{C^2}$$

Where,  
*M* = Metric  
*A* = Area  
*C* = Circumference

b. Ciri bentuk berdasarkan contour

1. Jarak Titik Pusat ke Tepi (CCD)

*Centroid-Contour Distance* (CCD) merupakan metode yang digunakan untuk memperoleh ciri bentuk dari suatu objek. Metode ini dapat merefleksikan karakter umum dari suatu bentuk tetapi kurva CCD tidak tahan terhadap perubahan rotasi. Metode ini menghitung jarak  $r$  dari *centroid* suatu bentuk dengan batasannya menurut kelipatan sudut  $\alpha$  (Isa dan Pradana, 2008).

Untuk membedakan bentuk objek satu dengan objek lainnya, dapat menggunakan parameter yang disebut dengan '**eccentricity**'. Eccentricity merupakan nilai perbandingan antara jarak foci ellips minor dengan foci ellips mayor suatu objek. Eccentricity memiliki rentang nilai antara 0 hingga 1. Objek yang berbentuk memanjang/mendekati bentuk garis lurus, nilai eccentricitynya mendekati angka 1, sedangkan objek yang berbentuk bulat/lingkaran, nilai eccentricitynya mendekati angka

0. Dengan rumus sebagai berikut :

$$e = \sqrt{1 - \frac{b^2}{a^2}}$$

**Where,**  
 **$e$  = eccentricity**  
 **$a$  = mayor axis**  
 **$b$  = minor axis**

### 2.12.2 Ciri Warna

Citra digital menggunakan model warna RGB (Red Green Blue) sebagai standar acuan warna (Putranto et al., 2010). Model HSV (Hue Saturation Value) juga sering digunakan dalam CBIR karena HSV merupakan format warna alamiah yang didasarkan pada tiga persepsi manusia tentang warna (Widodo, 2007).

*Hue* adalah sudut dari 0 sampai 360 derajat, biasanya 0 adalah merah, 60 derajat adalah kuning, 120 derajat adalah hijau, 180 derajat adalah cyan, 240 derajat adalah biru, dan 300 derajat adalah warna magenta. *Hue* menunjukkan jenis warna (seperti merah, biru atau kuning) atau corak warna yaitu tempat warna tersebut ditemukan dalam spektrum warna. Merah, kuning dan ungu (purple) adalah kata-kata yang menunjukkan *hue*. Oleh karena elemen warna *hue* berupa lingkaran dan dituliskan dalam sudut maka setiap operasi yang berkaitan dengan elemen warna *hue* (penambahan/pengurangan, perhitungan toleransi, filter warna) merupakan operasi sudut. Penambahan nilai *hue* sebesar  $n$  akan terjadi pergeseran sudut sebesar  $n$  derajat searah jarum jam sedangkan untuk pengurangan sebesar  $n$  akan terjadi pergeseran sudut sebesar  $n$  derajat berlawanan dengan arah jarum jam.

Saturasi adalah ukuran seberapa besar kemurnian dari warna tersebut. Sebagai contoh suatu warna yang semuanya merah tanpa putih adalah saturasi penuh. Jika ditambahkan putih ke merah, hasilnya menjadi lebih berwarna-warni dan warna digeser dari merah ke merah muda

(pink). *Hue* masih tetap merah tetapi nilai saturasinya berkurang. Saturasi biasanya bernilai dari 0 sampai 1 (atau 0 sampai 100%) dan menunjukkan nilai keabuabuan warna dimana 0 menunjukkan abu-abu dan 1 menunjukkan warnaprimier murni. *Value* disebut juga intensitas yaitu ukuran seberapa besar kecerahandari suatu warna atau seberapa besar cahaya datang dari suatu warna. *Value* dapat bernilai 0 sampai 100%. Suatu warna dengan nilai *value* 100% akan tampak secerah mungkin dan suatu warna dengan nilai *value* 0 akan tampak gelap mungkin. Sebagai contoh jika *hue* adalah merah dan *value* bernilai tinggi maka warna kelihatan cerah, tetapi ketika nilai *value* rendah maka warna tersebut akan terlihat gelap (Darma, 2010).

### 2.13 Seleksi Ciri

Ciri yang tersedia bisa banyak, masalahnya adalah bagaimana memilih kombinasi ciri terbaik yang dapat mengenali setiap obyek dengan akurat. Kemudian berkembang berbagai metode seleksi ciri: pendekatan bottom up atau top down, pendekatan filter atau wrapper, dan pendekatan lainnya. Salah satu tujuan dari *feature selection* adalah data *reduction*. Beberapa metode seleksi ciri diantaranya sebagai berikut (Normakristagaluh, 2012):

1. SFS (*Sequential Forward Selection –bottom up*) adalah Search *feature* menggunakan pendekatan bottom up: artinya mulai dengan suatu set kosong yang selalu ditambah dengan 1 *feature* sampai diperoleh suatu set

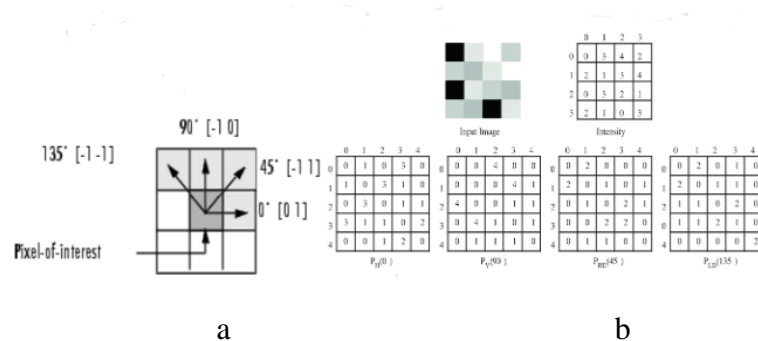
dengan jumlah *feature* yang kita ingini. Algoritma: dari  $n$  *feature* akan dipilih  $m$  *feature* yang optimal (bukan '*the best*'), dengan  $m < n$ . *Feature* yang optimal adalah fitur-fitur yang jika dikombinasikan dengan fitur lain dapat menghasilkan akurasi yang tinggi.

2. SBS (*Sequential Backward Selection –top down*) adalah Search *feature* menggunakan pendekatan top down: artinya mulai dengan suatu set penuh dengan jumlah *feature*  $n$ , eliminasi *feature* yang tidak relevan sampai diperoleh suatu set dengan jumlah *feature* yang kita ingini, misal  $m$ , dengan  $m < n$ . *Feature* yang tidak relevan adalah fitur-fitur yang jika dikombinasikan dengan fitur lain dapat menurunkan akurasi secara signifikan. Algoritma: dari  $n$  *feature* akan dipilih  $m$  *feature* yang optimal (bukan '*the best*'), dengan  $m < n$ .
3. SFFS (*Sequential Forward Floating Selection*) adalah Suatu *feature* yang sudah dipilih untuk dimasukkan atau dibuang dari set kombinasi *feature* yang sudah terpilih masih mungkin untuk dikeluarkan atau dimasukkan kembali. Adapun metode seleksi ciri terdiri dari 2 proses yaitu pencarian suatu fitur dan cek kriteria fungsi dan kombinasi fitur. Sedangkan pendekatan metode seleksi (*criterion function*) cirinya terdiri dari:
  - a. Filter (*Bhattacharya distance*) yaitu kombinasi *feature* yang baru dicek dengan Battacharya Distance, dimana kombinasi *feature* yang dipilih adalah yang jarak antar *feature* maksimum dan metode
  - b. Wrapper (*Classification method*) yaitu Pada setiap step pembentukan kombinasi *feature* yang baru, kombinasi yang terbaik dicek melalui

prosesklasifikasi citra dimana kombinasi *feature* yang terbaik adalah yang dapatmemberikan hasil ketelitian klasifikasi terbaik.

## 2.14 Metode GLCM ( Gray Level Co-Occurrence Matrix)

Metode GLCM merupakan suatu metode yang melakukan analisis terhadap suatu piksel pada citra dan mengetahui tingkat keabuan yang sering terjadi (Xie dkk 2010). Metode ini juga untuk tabulasi tentang frekuensi kombinasi nilai piksel yang muncul pada suatu citra. Untuk melakukan analisis citra berdasarkan distribusi statistik dari intensitas pikselnya, dapat dilakukan dengan mengekstrak fitur teksturnya. GLCM merupakan suatu metode untuk melakukan ekstraksi ciri berbasis statistik, perolehan ciri diperoleh dari nilai piksel matrik, yang mempunyai nilai tertentu dan membentuk suatu sudut pola (Kasim & Harjoko., 2014),(Xie dkk. 2010). Untuk sudut yang dibentuk dari nilai piksel citra menggunakan GLCM adalah  $0^0$ ,  $45^0$ ,  $90^0$ ,  $135^0$ (Eleyan & Demirel 2011), untuk sudut yang terbentuk terlihat pada gambar 2.5



Gambar 2.5 Sudut yang terbentuk: a) Piksel dengan berbagai sudut, b) Ilustrasi

Matrik co-ocurensi

Dari piksel-piksel tersebut terbentuk matrik ko-okurensi dengan pasangan pikselnya. Adanya matrik tersebut berdasarkan kondisi bahwa suatu matrik piksel akan mempunyai nilai perulangan sehingga terdapat pasangan aras keabuannya(Thakare & Patil 2014). Kondisi nilai piksel tersebut dinotasikan sebagai matrik dengan jarak dua posisi  $(x_1, y_1)$  dan  $(x_2, y_2)$ . Berdasarkan kondisi tersebut terlihat bahwa untuk membedakan antar matrik gambar dapat dilihat berdasarkan ciri matrik dengan menggunakan persamaan (Maurya dkk. 2014) sebagai berikut:

Tabel 2.4 Formula Ekstraksi Ciri

No	Feature	Formula
1	<i>Contras</i>	$\frac{\sum_i \sum_j (i - j)^2 C(i, j)}{\sum_i \sum_j C(i, j)}$
2	<i>Energy</i>	$\frac{\sum_i \sum_j C^2(i, j)}{\sum_i \sum_j C(i, j)}$
3	<i>Correlation</i>	$\frac{\sum_i \sum_j C(i, j) \log(C(i, j))}{\sum_i \sum_j C(i, j)}$
4	<i>Homogenitas</i>	$\frac{\sum_i \sum_j \frac{C(i, j)}{1 +  i + j }}{\sum_i \sum_j C(i, j)}$

- *Energy*

Mengukur tentang keseragaman atau sering disebut angular second moment. Energy akan bernilai tinggi ketika nilai pixel mirip satu sama lain sebaliknya akan bernilai kecil menandakan nilai dari GLCM normalisasi adalah heterogen. Nilai maksimum dari energy adalah 1 yang artinya distribusi pixel dalam kondisi konstan atau bentuk nya yang berperiodik (tidak acak).



- *Entropy*

Mengukur kompleksitas(keacakan) citra. Entropy akan bernilai tinggi ketika citra tidak seragam.

- *Contrast*

Mengukur frekuensi spasial dari citra dan perbedaan moment GLCM. Perbedaan yang dimaksudkan adalah perbedaan tinggi dan rendah nya pixel. Contrast akan bernilai 0 jika pixel ketetangaan mempunyai nilai yang sama.

- *Correlation*

Mengukur linearitas (the joint probability) dari sejumlah pasangan pixel (pairs).

- *Homogeneity*

Mengukur homogenitas. Nilai ini sangat sensitif terhadap nilai disekitar diagonal utama. Bernilai tinggi ketika semua pixel mempunyai nilai yang sama / seragam. Kebalikan dari contrast yaitu akan bernilai besar jika mempunyai nilai pixel yang sama pada saat energy bernilai tetap.

Berdasarkan Tabel 2.4, untuk feature kontras penyebaran elemen matrik pada citra terletak jauh dari diagonal utama akan mempunyai nilai kontras yang cukup besar. Nilai kontras merupakan suatu variasi antarderajat keabuan di suatu matrik pada citra (Maurya dkk. 2014). Pada rumus di atas,  $C$  mencerminkan nilai pixel (amplitude) dalam GLCM,  $i$  menunjukkan indeks untuk elemen secara lateral (trace ke- $i$ ) dan  $j$  indeks untuk elemen secara vertikal (sampel waktu ke- $j$ ).