

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Tanaman Aglaonema**

Tanaman *Aglaonema* adalah tanaman hias dengan nama ilmiah *aglaonema sp* atau di Indonesia lebih dikenal dengan sebutan *Sri Rejeki*. *Aglaonema* diperkirakan berasal dari Asia Tenggara bahkan sebagian varietasnya berasal dari Indonesia. *Aglaonema* berasal dari bahasa Yunani, yaitu *Aglos* yang berarti sinar dan *nema* yang berarti benang, sehingga *Aglaonema* dapat diartikan sebagai benang yang bersinar.

Dalam sistematika (taksonomi) tumbuhan, kedudukan tanaman hias *Aglaonema* diklasifikasikan sebagai berikut : (Pulungan, 2008)

Kingdom : Plantae  
Divisio : Magnolophyta  
Kelas : Liliopsida  
Ordo : Alismatales  
Suku : Araceae  
Marga : Aglaonema  
Species : Aglaonema Sp.

Tanaman *Aglaonema* disukai banyak orang karena memiliki warna dan tekstur daun yang unik. Secara umum *Aglaonema* terbagi dua, yaitu *Aglaonema* spesies dan *Aglaonema* hibrida (persilangan). Kedua jenis tersebut memiliki perbedaan diantaranya *Aglaonema* spesies umumnya memiliki warna kehijau-hijauan dengan corak hijau kehitaman, sedangkan *Aglaonema* hibrida (persilangan) umumnya memiliki warna daun lebih bervariasi, seperti putih, biru, hijau muda, hijau tua, merah muda, merah hingga kuning. Bentuk dan ukuran daunnya bermacam-macam tergantung dari jenisnya. Permukaan daun rata, licin dan tidak berbulu serta memiliki tepi daun rata (Leman, 2004). Variasi jenis tanaman *Aglaonema* dapat dilihat pada gambar 2.1 berikut :



**Gambar 2.1 Variasi Jenis Tanaman *Aglaonema***

Tanaman *Aglaonema* di dunia diperkirakan memiliki hampir 8000 jenis *Aglaonema* yang terdiri dari jenis *Aglaonema* spesies maupun hasil persilangan (*hibrida*) (Gusadha, 2011). Beberapa nama jenis tanaman *Aglaonema* yang digunakan pada penelitian ini dapat dilihat pada Tabel 2.1 berikut :

**Tabel 2.1 Beberapa Jenis *Aglaonema***

No	Jenis <i>Aglaonema</i>	Tipe
1.	<i>Aglaonema Dona Carmen</i>	Hibrida
2.	<i>Aglaonema Jatayu</i>	Hibrida
3.	<i>Aglaonema Streetlight</i>	Hibrida
4.	<i>Aglaonema Pattaya Beauty</i>	Hibrida
5.	<i>Aglaonema Manila Whirl</i>	Spesies
6.	<i>Aglaonema Stripes</i>	Hibrida
7.	<i>Aglaonema Widuri</i>	Hibrida
8.	<i>Aglaonema Star</i>	Hibrida
9.	<i>Aglaonema Chiang May</i>	Hibrida
10.	<i>Aglaonema Chochinchinense</i>	Spesies
11.	<i>Aglaonema Butterfly</i>	Hibrida
12.	<i>Aglaonema Sparkling Sarah</i>	Hibrida
13.	<i>Aglaonema Red Jewel</i>	Hibrida
14.	<i>Aglaonema Diana</i>	Hibrida
15.	<i>Aglaonema Ria</i>	Hibrida

## 2.2 Citra Digital

Citra (*image*) merupakan gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus

(*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam. (Munir, 2004).

### **2.2.1 Elemen-elemen Citra Digital**

Citra digital mengandung sejumlah elemen-elemen dasar yaitu : (Munir, 2004)

#### **1. Kecerahan (*brightness*).**

Kecerahan merupakan intensitas cahaya. Pada sebuah titik (*pixel*) di dalam citra kecerahan bukanlah intensitas yang riil, tetapi intensitas rata-rata dari suatu area yang melingkupinya.

#### **2. Kontras (*contrast*).**

Kontras menyatakan sebaran terang (*lightness*) dan gelap (*darkness*) pada sebuah gambar. Citra dengan kontras rendah memiliki ciri sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

#### **3. Kontur (*contour*)**

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada *pixel-pixel* yang bertetangga. Karena adanya perubahan intensitas inilah mata kita mampu mendeteksi tepi-tepi (*edge*) objek di dalam citra.

#### **4. Warna (*color*)**

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

#### **5. Bentuk (*shape*)**

Bentuk (*Shape*) adalah properti intrinsik dari objek tiga dimensi, dengan pengertian bahwa bentuk merupakan properti intrinsik utama untuk sistem visual manusia. Manusia lebih sering mengasosiasikan objek dengan bentuknya ketimbang elemen lainnya. Informasi bentuk objek dapat diekstraksi dari citra pada permulaan pra-pengolahan dan segmentasi citra.

## 6. Tekstur (*texture*)

Tekstur dicirikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan *pixel-pixel* yang bertetangga. Jadi, tekstur tidak dapat didefinisikan untuk sebuah *pixel*. Sistem visual manusia pada hakikatnya tidak menerima informasi citra secara independen pada setiap *pixel*, melainkan suatu citra dianggap sebagai suatu kesatuan.

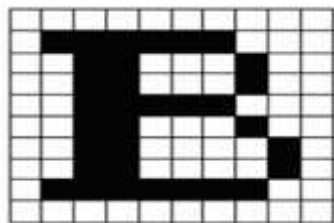
### 2.2.2 Jenis Citra

Beberapa jenis citra yang sering digunakan dalam pengolahan citra digital adalah adalah : (Putra, 2010).

#### 1. Citra Biner ( Monokrom)

Citra biner merupakan citra digital yang hanya memiliki nilai *pixel* hitam dan putih. Citra biner disebut juga citra B & W ( *Black and White* ) atau citra monokrom. Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengambangan, *morphologi* ataupun *dithering*.

Contoh Citra biner dapat dilihat pada gambar 2.2 dan representasi citra biner dapat dilihat pada gambar 2.3 berikut : (Rachmawati, 2013)



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

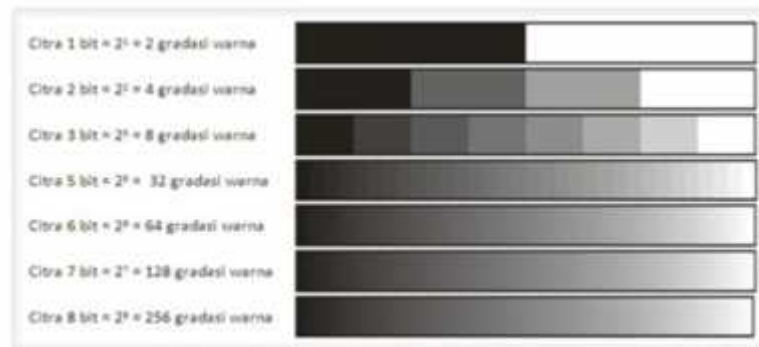
Gambar 2.2 Citra Biner    Gambar 2.3 Representasi Citra Biner

#### 2. Citra *Grayscale*

Citra *grayscale* merupakan citra digital yang memiliki satu nilai pada setiap *pixel*-nya, dengan kata lain bagian RED, GREEN atau BLUE. Infotmisi yang dibutuhkan pada setiap piksel citra *grayscale* lebih sedikit dibandingkan dengan citra warna. sehingga pemrosesan data dalam *image processing* lebih mudah, dan juga berfungsi untuk menyederhanakan model citra.

Citra *grayscale* menangani gradasi warna hitam dan putih, yang menghasilkan efek warna abu-abu. Jenis citra *grayscale* ini, untuk warna dinyatakan dengan intensitas. Intensitas berkisar antara 0 sampai dengan 255.

Nilai 0 menyatakan hitam dan nilai 255 menyatakan putih (Kadir, 2013). Gradasi citra biner dapat dilihat pada gambar 2.4 berikut ini : (Wicaksono, 2010).



**Gambar 2.4 Gradasi Warna Citra *Grayscale***

Tujuan *grayscale* antara lain : (Rachmawati, 2013)

- a. Mempresentasikan aras abu-abu (*grayscale*) atau kode warna.
- b. Kisaran nilai ditentukan oleh bit yang dipakai dan akan menunjukkan resolusi aras abu-abu (*gray level resolution*).
  1. 1 bit - 2 aras atau warna : [0,1]
  2. 4 bit – 16 aras atau warna : [0,15]
  3. 8 bit – 256 aras atau warna : [0,255]
  4. 24 bit – 16 juta warna (true color)
    - Kanal merah – red (R) : [0,255]
    - Kanal hijau – green (G) : [0,255]
    - Kanal biru – blue (B) : [0,255]

Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing *r*, *g* dan *b* menjadi citra *grayscale* dilakukan proses konversi rgb ke *grayscale* dengan cara mengambil rata-rata dari nilai *r*, *g* dan *b* sesuai dengan rumus berikut :

$$S = \frac{r+g+b}{3} \quad (2.1)$$

Keterangan :

- s* = nilai derajat keabuan
- r* = nilai *red* pada suatu nilai RGB
- g* = nilai *green* pada suatu nilai RGB
- b* = nilai *blue* pada suatu nilai RGB

Pada penelitian ini, proses mengubah citra berwarna menjadi citra *grayscale* dilakukan pada tahap *preprocessing* sebelum dilakukan proses ekstraksi ciri tekstur menggunakan metode statistik orde dua.

### 3. Citra Warna ( True Color)

Citra warna merupakan citra digital yang memiliki kombinasi warna RED, GREEN dan BLUE. Citra warna dibagi dalam beberapa jenis berdasarkan bit-nya yaitu :

1. Citra Warna (8 bit)

Setiap pixel dari citra warna (8 bit) hanya diwakili oleh 8 bit dengan jumlah warna maksimum 256 warna.

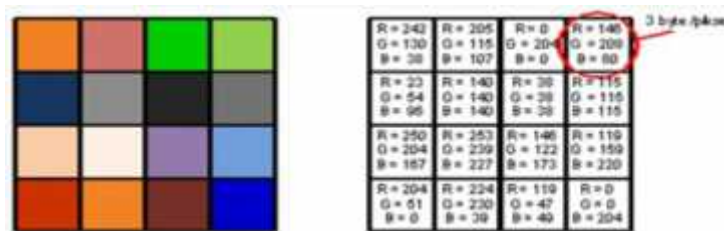
2. Citra Warna (16 bit)

Citra warna 16 bit (biasanya sering disebut sebagai citra highcolor) dengan setiap pixel-nya diwakili dengan 2 byte memori (16 bit). Warna 16 bit memiliki 65.536 warna.

3. Citra Warna (24 bit)

Setiap pixel dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat dilihat penglihatan manusia.

Representasi citra warna dapat dilihat pada gambar 2.5 berikut ini :



Gambar 2.5 Representasi Citra Warna

### 2.3. Pengolahan Citra

Pengolahan citra adalah pemrosesan citra, khususnya menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Tujuan pengolahan Citra adalah memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (komputer). Teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra dengan kualitas

citra yang lebih baik daripada citra masukan. Operasi-operasi pengolahan citra dapat diklasifikasikan sebagai berikut: (Munir, 2004)

1. Perbaikan kualitas citra (*image enhancement*).

Tujuan *image enhancement* adalah memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Dengan operasi ini, ciri-ciri khusus yang terdapat di dalam citra lebih ditonjolkan. Contoh-contoh operasi perbaikan citra:

- a. Perbaikan kontras gelap/terang
- b. Perbaikan tepian objek (*edge enhancement*)
- c. Penajaman (*sharpening*)
- d. Pemberian warna semu (*pseudocoloring*)
- e. Penapisan derau (*noise filtering*)

2. Pemugaran Citra (*Image Restoration*)

Pemugaran citra bertujuan untuk menghilangkan atau meminimumkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra. Bedanya, pada pemugaran citra penyebab degradasi gambar diketahui. Contoh-contoh operasi pemugaran citra:

- a. penghilangan kesamaran (*deblurring*).
- b. penghilangan derau (*noise*)

3. Pemampatan citra (*image compression*).

Operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Namun, dalam operasi ini, citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus.

3. Segmentasi citra (*image segmentation*).

Jenis operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu.

4. Pengorakan citra (*image analysis*)

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik pengorakan citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh-contoh operasi pengorakan citra:

- a. Pendeteksian tepi objek (edge detection)
  - b. Ekstraksi batas (boundary)
  - c. Representasi daerah (region)
6. Rekonstruksi citra (*image reconstruction*)

Rekonstruksi citra bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis. Misalnya foto *rontgen*, dengan sinar X digunakan untuk membentuk ulang gambar organ tubuh.

## **2.4 Ekstraksi Ciri**

Ekstraksi ciri adalah proses mengambil ciri-ciri yang terdapat pada citra. Ciri-ciri tersebut adalah besaran komponen tertentu dari citra objek yang mewakili sifat utama citra objek, sekaligus mengurangi dimensi citra objek menjadi sekumpulan bilangan yang lebih sedikit tetapi representative. (Syaputra, 2009)

Pada penelitian ini, ciri yang digunakan sebagai proses ekstraksi adalah ciri tekstur dan ciri warna.

### **2.4.1 Tekstur**

Tekstur merupakan keteraturan pola tertentu secara berulang-ulang dengan interval jarak dan arah tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital. Suatu permukaan dikatakan mempunyai suatu informasi tekstur, bila luasannya diperbesar tanpa mengubah skala, maka sifat-sifat permukaan hasil perluasan mempunyai kemiripan dengan permukaan asalnya. Informasi tekstur dapat digunakan untuk membedakan sifat-sifat permukaan suatu benda (kasar dan halus) dalam citra, dan sifat-sifat spesifik dari kekasaran dan kehalusan permukaan. Syarat terbentuknya tekstur ada dua, yaitu : (Arifah, 2010)

1. Adanya pola-pola primitif yang terdiri dari satu atau lebih piksel. Bentuk-bentuknya dapat berupa titik, garis lurus, garis lengkung, luasan, dan lain-lain yang merupakan elemen dasar dari sebuah bentuk.
2. Pola-pola primitif tadi muncul berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat diprediksi atau ditemukan karakteristik pengulangannya.



Metode untuk mengekstrak ciri-ciri tekstur dapat digolongkan ke dalam empat kategori, yaitu (Mihran, 1998) :

#### 1. Metode statistikal

Metode statistikal menggolongkan tekstur dengan distribusi statistik pada intensitas gambar. Metode statistikal yang paling sering digunakan adalah *co-occurrence matrices*. Metode statistikal lainnya meliputi: *Fourierpower spectra*, dan *shift-invariant principal component analysis (SPCA)* dan Statistik Orde Dua.

#### 2. Metode geometrik

Metode ini menggambarkan tekstur dengan mengidentifikasi struktur sederhana dan aturan-aturan penempatannya. Meliputi : *Voronoi tessellation features dan structural methods*.

#### 3. Metode berdasarkan model

Metode ini didasarkan pada susunan model suatu gambar, yang tidak hanya dapat digunakan untuk menggambarkan tekstur, tetapi menyatukannya juga, meliputi : *Markov random field dan fractal model*.

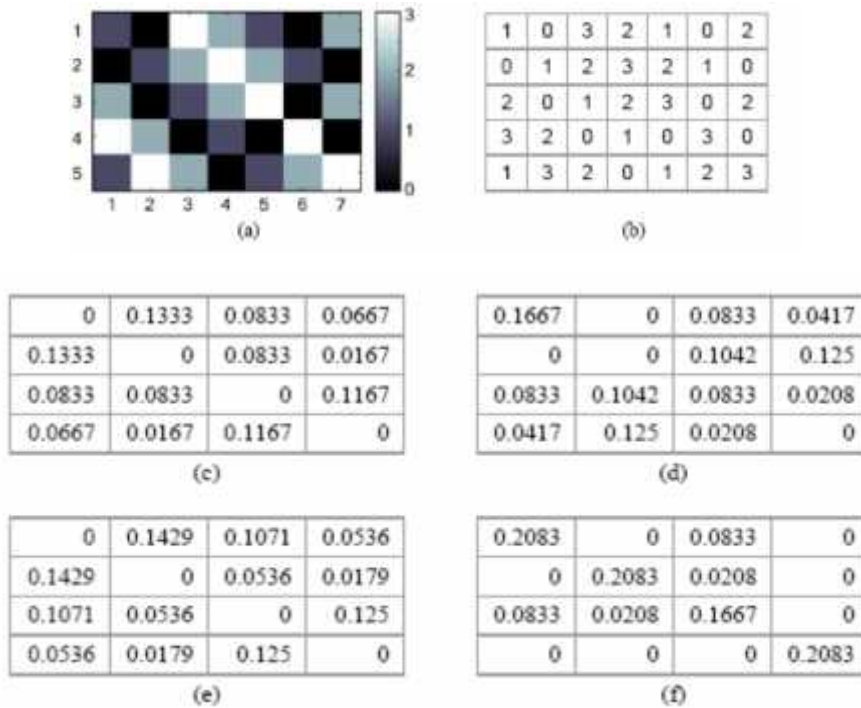
#### 4. Metode pemrosesan sinyal/transformasi

Metode ini menggambarkan sebuah gambar di dalam bentuk yang baru, dimana karakteristik dari tekstur dapat diperoleh dengan lebih mudah, meliputi: *spatial domain filters, fourier domain filtering, Filter gabor dan transformasi Wavelet*.

### **2.4.1.1 Metode Statistik Orde Dua**

Statistik orde dua adalah penghitungan probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientasi sudut tertentu. Cara kerjanya dengan membentuk sebuah matriks kookurensi dari data citra, kemudian dilanjutkan dengan menentukan ciri.

Kookurensi berarti kejadian bersama, yaitu jumlah kejadian satu level nilai piksel bertetangga dengan satu level nilai piksel lain dalam jarak ( $d$ ) dan orientasi sudut ( $\theta$ ) tertentu. Jarak dinyatakan dalam piksel dan orientasi dinyatakan dalam derajat. Orientasi dibentuk dalam empat arah sudut yaitu  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ . Sedangkan jarak antar piksel biasanya ditetapkan sebesar 1 piksel. Contoh ilustrasi nilai kookurensi matrik dapat dilihat pada gambar 2.6 (Haralick.1973).



**Gambar 2.6 Ilustrasi Pembuatan Matriks Kookurensi**

Keterangan :

- (a) Citra masukan
- (b) Nilai intensitas citra masukan
- (c) Hasil matriks kookurensi 0°
- (d) Hasil matriks kookurensi 45°
- (e) Hasil matriks kookurensi 90°
- (f) Hasil matriks kookurensi 135°

Setelah dibuat matriks kookurensi tersebut, selanjutnya dihitung ciri statistik orde dua yang merepresentasikan citra yang diamati. Ciri digunakan dalam penelitian ini menggunakan 6 ciri, yaitu *Angular Second Moment*, *Contrast*, *Correlation*, *Variance*, *Inverense Different Moment* dan *Entropy* (Haralick.1973).

a. Angular Second Moment (ASM)

*Angular Second Moment* (ASM) menunjukkan ukuran sifat homogenitas citra. Persamaan untuk menghitung *Angular Second Moment* (ASM) dapat dilihat pada pada persamaan (2.2)

$$ASM = \sum_i \sum_j \{p(i, j)\}^2 \tag{2.2}$$

$p(i, j)$  menyatakan nilai pada baris i dan kolom j pada matriks kookurensi yang dihitung sebelumnya.

b. Contrast

*Contrast* menunjukkan penyebaran (*momen inersia*) elemen-elemen matriks citra. Jika letaknya jauh dari diagonal utama, nilai kekontrasan besar. Secara visual, nilai kekontrasan adalah ukuran variasi antar derajat keabuan suatu daerah citra. Persamaan untuk menghitung *Contrast* dapat dilihat pada persamaan (2.3).

$$CON = \sum_k \sum_i \sum_j p(i, j) \quad |i-j|=k \quad (2.3)$$

c. Correlation

*Correlation* menunjukkan ukuran ketergantungan linear derajat keabuan citra sehingga dapat memberikan petunjuk adanya struktur linear dalam citra. Persamaan untuk menghitung *Correlation* dapat dilihat pada persamaan (2.4).

$$COR = \frac{\sum_i \sum_j i \cdot j \cdot p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (2.4)$$

d. Variance

*Variance* menunjukkan variasi elemen-elemen matriks kookurensi. Citra dengan transisi derajat keabuan kecil akan memiliki variansi yang kecil pula. Persamaan untuk menghitung *Variance* dapat dilihat pada persamaan (2.5).

$$VAR = \sum_i \sum_j (i - \mu_x) (j - \mu_y) p(i, j) \quad (2.5)$$

e. Inverse Different Moment (IDM)

*Inverse Different Moment* menunjukkan kehomogenan citra yang berderajat keabuan sejenis. Citra homogen akan memiliki harga *Inverse Different Moment* (IDM) yang besar. Persamaan untuk menghitung *Inverse Different Moment* dapat dilihat pada persamaan (2.6).

$$IDM = \sum_i \sum_j \frac{1}{1+i-j} p(i, j) \quad (2.6)$$

f. Entropy

Entropy digunakan untuk menunjukkan ukuran ketidakteraturan bentuk. Harga ENT besar untuk citra dengan transisi derajat keabuan merata dan bernilai kecil jika struktur citra tidak teratur (bervariasi). Persamaan untuk menghitung Entropy dapat dilihat pada persamaan (2.7).

$$ENT = - \sum_{i,j} p(i,j) \log p(i,j) \quad (2.7)$$

Keterangan :

$\mu_x$  adalah nilai rata-rata baris ke- $i$        $\mu_y$  adalah nilai rata-rata baris ke- $j$

$\sigma_x$  adalah standar deviasi baris ke- $i$        $\sigma_y$  adalah standar deviasi baris ke- $j$

#### 2.4.1.2 Normalisasi

Tujuan utama normalisasi adalah agar terjadi sinkronisasi data, disamping itu juga untuk memudahkan dalam proses komputasi. Hasil dari normalisasi adalah sekumpulan bilangan yang berkisar antara 0 dan 1. Metode normalisasi yang digunakan pada penelitian ini adalah berdasarkan rumus (Purnomo, M.H., 2006) :

$$A'_i = \frac{A_i}{A_{max}} \quad (2.8)$$

$A_i$  merupakan bilangan ke- $i$  dan  $A_{max}$  merupakan bilangan maksimum dalam suatu N (data) bilangan masukan serta  $A'_i$  merupakan bilangan baru ke- $i$  yang telah dilakukan proses normalisasi.

#### 2.4.2 Warna

Ruang warna atau kadang disebut model warna merupakan suatu spesifikasi sistem koordinat dan suatu subruang dalam sistem tersebut dengan setiap warna dinyatakan dengan satu titik didalamnya . Tujuan dibentuknya ruang warna adalah untuk memfasilitasi spesifikasi warna dalam bentuk suatu standar. Jenis-jenis ruang warna diantaranya : (Kadir, 2013).

##### 1. Ruang Warna RGB

Ruang warna ini menggunakan tiga komponen dasar yaitu merah (R), hijau (G), dan biru (B). Setiap piksel dibentuk oleh ketiga komponen tersebut.

RGB biasa digunakan karena kemudahan dalam perancangan *hardware*, tetapi sebenarnya tidak ideal untuk beberapa aplikasi.

Warna merah, hijau, dan biru sesungguhnya berkorelasi erat, sangat sulit untuk beberapa algoritma pemrosesan citra. Sebagai contoh, kebutuhan untuk memperoleh warna alamiah seperti merah dengan menggunakan RGB menjadi sangat kompleks mengingat komponen R dapat berpasangan dengan G dan B, dengan nilai berapa saja. Hal ini menjadi mudah jika menggunakan ruang warna HSL atau HSV.

## 2. Ruang Warna CMY/CMYK

Model warna CMY (*cyan, magenta, yellow*) mempunyai hubungan dengan RGB sebagai berikut:

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

Pada *CMY*, warna hitam diperoleh jika *C*, *M*, dan *Y* bernilai sama. Namun, pada aplikasi printer, warna hitam ditambahkan tersendiri sehingga membentuk *CMYK*, dengan *K* menyatakan warna hitam. Alasannya, kalau ada warna hitam, warna dapat diambilkan secara langsung dari tinta hitam, tanpa mencampur dengan warna lain.

## 3. Ruang Warna YIQ

Ruang warna YIQ, yang juga dikenal dengan nama ruang warna NTSC, dirumuskan sebagai NTSC ketika mengembangkan sistem televisi warna di Amerika Serikat. Pada model ini, *Y* disebut *luma* (yang menyatakan luminans) dan *I* dan *Q* menyatakan *chroma*.

## 4. Ruang Warna YCbCr

Ruang warna YCbCr sering digunakan pada video digital. Pada ruang warna ini, komponen *Y* menyatakan intensitas, sedangkan *Cb* dan *Cr* menyatakan informasi warna.

## 5. Ruang Warna HSI, HSV, dan HSL

HSV dan HSL merupakan contoh ruang warna yang merpresentasikan warna seperti yang dilihat oleh manusia. *H* berasal dari kata “hue”, *S* berasal dari “saturation”, *L* berasal dari kata “luminance”, *I* berasal dari kata “intensity”, dan *V* berasal dari “value”.

### 2.4.2.1 Model Warna HSV

HSV (hue, saturation, value) merupakan model warna yang diturunkan dari RGB. Model warna HSV mendefinisikan warna dalam terminologi *Hue*, *Saturation*, dan *Value*. *Hue* menyatakan warna sebenarnya, seperti merah, violet, dan kuning. *Hue* berasosiasi dengan panjang gelombang cahaya. *Saturation* menyatakan tingkat kemurnian suatu warna, yaitu mengindikasikan seberapa banyak warna putih diberikan pada warna. *Value* adalah atribut yang menyatakan banyaknya cahaya yang diterima oleh mata tanpa memperdulikan warna. (Karmilasari, 2011)

*Hue* merupakan sudut 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. Gamabr elemen warna *Hue* dapat dilihat pada Gambar 2.7 berikut :



**Gambar 2.7 Lingkaran Elemen Warna Hue (Putranto, 2010)**

Karena elemen warna *hue* berupa lingkaran dan dituliskan dalam sudut, maka setiap operasi yang berkaitan dengan elemen warna *hue* baik itu penambahan, pengurangan, perhitungan toleransi, filter warna, merupakan operasi sudut. Jika terjadi penambahan nilai *hue* sebesar  $n$ , maka akan terjadi pergeseran sudut sebesar  $n^0$  searah jarum jam sebaliknya jika terjadi pengurangan nilai *hue* sebesar  $n$  akan terjadi pergeseran sudut sebesar  $n^0$  berlawanan arah jarum jam.

*Saturation* merupakan salah satu elemen HSV yang mewakili tingkat intensitas warna. Pada nilai tingkat kecerahan (*value*) yang sama nilai *saturation* menggambarkan nilai kedekatan suatu warna pada warna abu-abu. Pada sistem nilai *saturation* memiliki rentang antara 0 (minimum) dan 1 atau 100% (maksimum).

Dalam ruang warna HSV, elemen *value* digunakan untuk merepresentasikan tingkat kecerahan warna yang berkisar antara 0% hingga 100%. Pada nilai *value* maksimum warna yang dihasilkan adalah warna dengan tingkat kecerahan maksimum (warna putih), sedangkan pada *value* minimum dihasilkan warna dengan tingkat kecerahan minimum (warna hitam).

Berapapun nilai hue dan saturatin warna, jika nilai value yang dimiliki adalah 0 (minimum) maka warna yang dihasilkan adalah warna hitam. Nilai maksimum adalah 1 (100%), dimana warna yang dihasilkan akan memiliki tingkat kecerahan maksimum. (Putranto, 2010)

Karena model warna HSV merupakan model warna yang diturunkan dari model warna RGB, maka untuk mendapatkan warna HSV ini kita harus melakukan proses konversi warna dari RGB ke HSV.

#### 2.4.2.2 Konversi RGB ke HSV

Warna dibentuk oleh model warna merupakan hasil campuran dari warna merah, hijau, dan biru (RGB) berdasarkan komposisi tertentu, untuk mendapatkan warna HSV harus dilakukan proses konversi warna dari RGB ke HSV dengan menggunakan fungsi pada matlab yaitu '*rgb2hsv*'. Nilai HSV yang digunakan pada penelitian ini adalah nilai *mean* dari masing-masing nilai H, S, dan V.

*Mean* adalah rerata hitung ukuran tendensi sentral yang memberikan gambaran umum mengenai data. Rata-rata hitung diperoleh dengan menjumlahkan semua nilai data dengan jumlah data. Rata-rata hitung populasi merupakan nilai rata-rata dari populasi. Perhitungan rata-rata dapat menggunakan persamaan berikut : (Kusumanto, 2011)

$$\mu = \frac{\sum x}{N} \quad (2.9)$$

Keterangan :  $\mu$  : rata-rata hitung populasi

$\Sigma$  : simbol operasi penjumlahan

X : nilai data dalam populasi

N : jumlah observasi (data)

$\Sigma N$  : jumlah keseluruhan nilai X (data) dalam populasi

## 2.4 Klasifikasi

Klasifikasi adalah metode yang digunakan untuk mengelompokkan suatu objek ke dalam kelompok atau kelas tertentu. Klasifikasi merupakan suatu proses menemukan model yang menggambarkan dan membedakan kelas dari data yang bertujuan untuk memprediksi kelas dari objek yang kelasnya belum diketahui. Klasifikasi data terdiri dari 2 langkah. Pertama adalah fase training, dimana algoritma klasifikasi dibuat untuk menganalisa data training lalu direpresentasikan dalam bentuk rule klasifikasi. Proses kedua adalah klasifikasi, dimana data tes digunakan untuk memperkirakan akurasi dari rule klasifikasi (Han, 2006).

Algoritma klasifikasi yang banyak digunakan secara luas, yaitu *Decision classification trees*, *Bayesian classifiers*, *Naive Bayes classifiers*, *Neural networks*, *Analisa Statistik*, *Algoritma Genetika*, *Rough sets*, *k-nearest neighbor*, *Metode Rule Based*, *Memory based reasoning*, dan *Support vector machines (SVM)*. (Leidiyana, 2013). Pada penelitian ini, metode yang digunakan untuk klasifikasi gambar tanaman *Agalonema* adalah metode *K-Nearest neighbour*.

### 2.5.1 Klasifikasi *K-Nearest Neighbour*

Algoritma *K-nearest neighbor* merupakan suatu teknik klasifikasi yang sangat populer yang diperkenalkan oleh Fix dan Hodges (1951), yang telah terbukti menjadi algoritma sederhana yang baik. (Chan *et al* dalam Whidhiasih, 2013). Algoritma KNN tergolong dalam algoritma *supervised* yaitu proses pembentukan algoritma diperoleh melalui proses pembelajaran (*learning*) pada record-record lama yang sudah terklasifikasi dan hasil pembelajaran tersebut dipakai untuk mengklasifikasikan record baru dengan output yang belum diketahui. (Pandie, 2012)

Algoritma *K-Nearest Neighbor* (KNN) melakukan klasifikasi terhadap objek berdasarkan jaraknya paling dekat dengan objek tersebut. Teknik ini sangat sederhana dan mudah diimplementasikan. Dalam hal ini jumlah tetangga atau nilai  $k$  terdekat ditentukan oleh user. Misalkan ditentukan  $k=5$ , maka setiap data uji dihitung jaraknya terhadap data latih dan dipilih 5 data latih yang jaraknya paling dekat ke data uji. Lalu periksa output atau labelnya masing-masing, kemudian tentukan output mana yang frekuensinya paling banyak. Lalu masukkan



data uji ke kelompok dengan output paling banyak. Misalkan dalam kasus klasifikasi dengan 3 kelas, lima data tadi terbagi atas tiga data dengan output 3 data dengan output kelas 1, satu data dengan output kelas 2 dan satu data dengan output kelas 3, maka dapat disimpulkan bahwa output dengan label kelas 1 adalah yang paling banyak. Sehingga data uji masuk kekelompok kelas 1. Prosedur ini dilakukan untuk semua data uji (Santosa, 2007).

Secara umum untuk mendefinisikan jarak digunakan rumus jarak *Euclidean*, seperti terlihat pada persamaan berikut : (Sikki, 2009)

$$d_i = \sqrt{\sum_{l=1}^p x_{2l} - x_{1l}^2} \quad (2.10)$$

Dengan:  $d_i$  = Jarak  $i$  = variabel data  
 $X_i$  = sampel data  $p$  = dimensi data  
 $X_2$  = data uji

Berikut ini merupakan langkah-langkah dalam algoritma *K-Nearest Neighbour* : (song *et al* dalam Yanti, 2012)

1. Tentukan nilai tetangga ( $k$ )
2. Hitung jarak data uji dengan setiap data latih menggunakan jarak *Euclidean Distance*.
3. Urutkan jarak tersebut dari data terdekat.
4. Dapatkan sebanyak  $k$  data yang memiliki jarak terdekat.
5. Tentukan kelas untuk data uji, berdasarkan label mayoritas  $k$ .

Adapun kelebihan dari KNN yaitu ketangguhan terhadap data training yang memiliki banyak noise dan efektif apabila data trainingnya besar. Sedangkan, kelemahan KNN adalah KNN perlu menentukan nilai dari parameter  $k$  (ketetanggan), training berdasarkan jarak tidak jelas mengenai jenis jarak apa yang harus digunakan dan atribut mana yang harus digunakan untuk mendapatkan hasil terbaik, dan biaya komputasi cukup tinggi karena diperlukan perhitungan jarak dari tiap query instance pada keseluruhan training sample.(Sikki, 2009)

## 2.6 Pengujian Akurasi

Akurasi merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya (true value atau reference value). Tingkat akurasi diperoleh dengan perhitungan sesuai dengan persamaan berikut: (Yanti, 2012))

$$\text{Akurasi Keberhasilan} = \frac{\sum \text{Pengujian yang bernilai benar}}{\sum \text{Banyak Data latih}} \times 100\% \quad (2.11)$$