

## BAB II

### LANDASAN TEORI

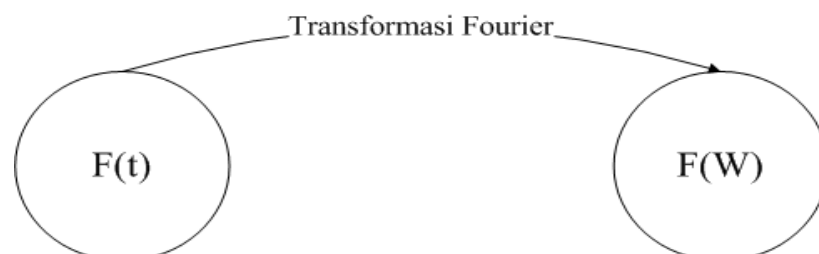
#### 2.1 Analisis Frekuensi Citra

##### 2.1.1 Metode Frekuensi Domain

Secara umum, metode yang digunakan dalam pemrosesan citra digital dapat dibagi menjadi dua kelompok, yaitu metode spatial domain dan metode frequency domain. Pada metode spatial domain, pemrosesan dilakukan dengan cara memanipulasi nilai pixel dari citra tersebut secara langsung. Sedangkan pada pengolahan citra digital dengan metode frequency domain, informasi citra digital ditransformasikan lebih dulu dengan transformasi Fourier, kemudian dilakukan manipulasi pada hasil transformasi Fourier tersebut. Setelah manipulasi selesai, dilakukan inverse transformasi Fourier untuk mendapatkan informasi citra kembali. Metode frequency domain ini dapat digunakan untuk menyelesaikan masalah-masalah tertentu yang sulit jika dilakukan dengan menggunakan metode spatial domain [4].

##### 2.1.2 Transformasi Fourier 1D

Transformasi Fourier adalah suatu model transformasi yang memindahkan domain spasial atau domain waktu menjadi domain frekuensi.



**Gambar 1.1** Transformasi Fourier

Transformasi Fourier merupakan suatu proses yang banyak digunakan untuk memindahkan domain dari suatu fungsi atau obyek ke

dalam domain frekuensi. Di dalam pengolahan citra digital, transformasi fourier digunakan untuk mengubah domain spasial pada citra menjadi domain frekuensi. Analisa-analisa dalam domain frekuensi banyak digunakan seperti filtering. Dengan menggunakan transformasi fourier, sinyal atau citra dapat dilihat sebagai suatu obyek dalam domain frekuensi [19].

Transformasi Fourier dari  $f(x)$ , didefinisikan sebagai berikut:

$$f(u) = \int_{-\infty}^{\infty} f(x) \exp[-j2\pi ux] dx \dots\dots\dots(2.1)$$

di mana

$$j = \sqrt{-1}$$

Sebaliknya, jika diketahui  $F(u)$ , maka  $f(x)$  dapat diperoleh dengan Inverse Transformasi Fourier berikut:

$$f(x) = \int_{-\infty}^{\infty} F(u) \exp[j2\pi ux] du \dots\dots\dots(2.2)$$

Kedua persamaan di atas disebut dengan pasangan transformasi Fourier. Jika  $f(x)$  adalah bilangan real, biasanya  $F(u)$  merupakan bilangan kompleks yang bisa diuraikan menjadi:

$$F(u) = R(u) + jI(u) \dots\dots\dots(2.3)$$

dimana  $R(u)$  dan  $I(u)$  adalah komponen real dan imajiner dari  $F(u)$ .

Persamaan di atas juga sering dituliskan sebagai:

$$F(u) = |F(u)| e^{j\phi u} \dots\dots\dots(2.4)$$

dimana  $|F(u)|$  adalah magnitude dari  $F(u)$ , yang diperoleh dari :

$$|F(u)| = [R^2(u) + I^2(u)]^{1/2} \dots\dots\dots(2.5)$$

$$\phi(u) = \tan^{-1}[I(u)/R(u)] \dots\dots\dots(2.6)$$

Fungsi magnitude  $|F(u)|$  disebut juga spektrum Fourier dari  $f(x)$ , dan  $\phi(u)$  disebut dengan sudut fase dari  $f(u)$ .

Jika  $f(x)$  dijadikan diskrit maka persamaan transformasi Fourier diskrit adalah:

$$f(x) = \frac{1}{N} \sum_{x=0}^{N-1} F(u) \exp[-j2\pi ux/N] \dots\dots\dots(2.7)$$

dan

$$f(x) = \sum_{u=0}^{N-1} F(u) \exp[j2\pi ux/N] \dots\dots\dots(2.8)$$

Karena pada pengolahan citra digital, data yang digunakan berbentuk digital/diskrit maka dapat digunakan kedua persamaan diatas untuk melakukan transformasi dan inverse transformasi Fourier.

### 2.1.3 Transformasi Fourier 2D

Transformasi Fourier 2D dari suatu fungsi  $f(x,y)$  didefinisikan sebagai berikut :

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \exp[-j2\pi(ux + vy)] dx dy \dots\dots(2.9)$$

Invers dari transformasi tersebut adalah

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) \exp[-j2\pi(ux + vy)] du dv \dots\dots(2.10)$$

Transformasi Fourier membawa suatu citra dari ruang spasial ke ruang frekuensi. Fungsi basis dari transformasi Fourier adalah berupa sinyal fungsi (sinyal) sinus. Melalui transformasi Fourier, suatu citra (sinyal atau fungsi) dapat dinyatakan sebagai penjumlahan sinyal sinus dan kosinus dengan amplitudo dan frekuensi yang bervariasi. Frekuensi ini domain pada suatu citra dapat diketahui melalui transformasi ini [18].

FFT (Fast Fourier Transform) adalah teknik perhitungan cepat dari DFT (Discrete Fourier Transform). Sama seperti FFT biasa menghasilkan dua vektor untuk input tunggal (sinus dan kosinus koefisien koefisien, atau bagian nyata dan imajiner), 2D-FFT menghasilkan untuk gambar output (real dan imajiner) untuk input tunggal. Untuk gambar dunia nyata (seperti yang diambil dengan kamera) biasanya diasumsikan bahwa bagian nyata adalah kecerahan dan bagian imajiner adalah 0 [18]. Untuk gambar  $f(x,y)$ , 2D FFT  $F(u,v)$  dapat ditemukan dengan menggunakan rumus :

$$F(u, v) = \frac{1}{MN} \sum_{u=0}^M \sum_{v=0}^N f(x, y) e^{-j2\pi(\frac{ux}{N} + \frac{vy}{N})} \dots\dots\dots(2.11)$$

Inversnya adalah

$$f(x, y) = \sum_{u=0}^M \sum_{v=0}^N F(u, v) e^{j2\pi(\frac{ux}{N} + \frac{vy}{N})} \dots\dots\dots(2.12)$$

## **2.2 Pengertian Daun**

### **2.2.1 Sejarah Daun**

Daun merupakan salah satu bagian dan organ yang peranannya sangat penting dalam tumbuhan. Kebanyakan daun berwarna putih karena didalam daun mengandung klorofil walupun tidak sedikit warna daun tidak berwarna hijau. Klorofil dalam daun berguna untuk menangkap energi cahaya matahari yang digunakan tanaman untuk proses fotosintesis. Daun bisa disebut sebagai organ yang sangat penting yang dipunyai oleh tanaman karena daun merupakan organ autotrof obligat yang artinya untuk melangsungkan hidupnya tumbuhan harus dipasok dengan energi sendiri dengan mengubah cahaya matahari menjadi energi dengan bantuan proses fotosintesis. Bentuk daun sangat beragam, namun biasanya berupa helaian, bisa tipis atau tebal [14].

Gambaran dua dimensi daun digunakan sebagai pembeda bagi bentuk-bentuk daun. Bentuk dasar daun membulat, dengan variasi cuping menjari atau menjadi elips dan memanjang. Bentuk ekstremnya bisa meruncing panjang. Daun juga bisa bermodifikasi menjadi duri (misalnya pada kaktus), dan berakibat daun kehilangan fungsinya sebagai organ fotosintetik. Daun tumbuhan sukulen atau xerofit juga dapat mengalami peralihan fungsi menjadi organ penyimpan air [14].

### **2.2.2 Bentuk-Bentuk Tulang Daun**

Salah satu cara untuk mengidentifikasi tumbuhan biasanya kita juga menggunakan susunan tulang daun. Tulang daun terdiri dari 4 bagian [13].

- **Tulang Daun Menyirip**

Tulang daun menyirip berbentuk seperti susunan sirip ikan. Contohnya adalah daun mangga, daun jambu, daun nangka, dan daun rambutan.



**Gambar 1.2.** Daun Menyirip

- Tulang Daun Menjari

Bentuk tulang daun menjari seperti susunan jari-jari tangan. Contohnya adalah daun pepaya, daun singkong, kapas dan daun jarak.



**Gambar 1.3.** Tulang Daun Menjari

- Tulang Daun Melengkung

Bentuk tulang daun melengkung seperti garis-garis lengkung. tulang daun jenis ini dapat kita temukan pada berbagai tumbuhan di lingkungan sekitar kita. Ujung-ujung tulang daun melengkung terlihat menyatu. Contohnya adalah daun sirih, daun gadung, dan daun genjer.



**Gambar 1.4.** Daun Melengkung

- Tulang Daun Sejajar

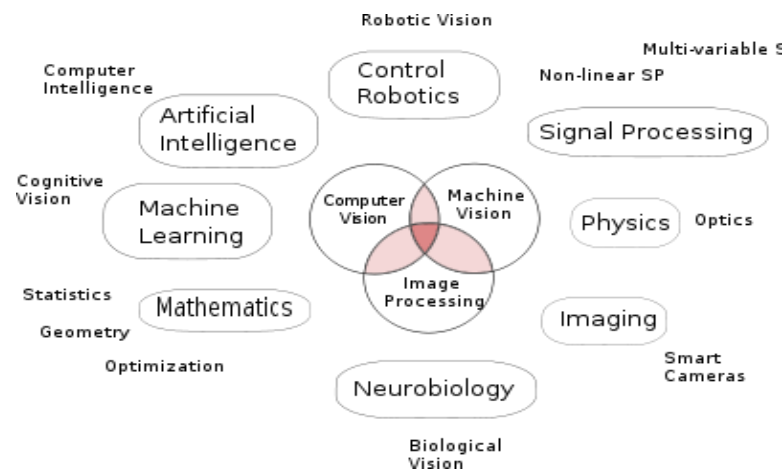
Bentuk tulang daun sejajar seperti garis-garis lurus yang sejajar. Tiap-tiap ujung tulang daun ini menyatu. Contohnya adalah tebu, padi, dan semua jenis rumput.



**Gambar 1.5** Daun Sejajar

### **2.3 Computer Vision**

Computer Vision didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali obyek yang diamati atau diobservasi. Arti dari Computer Vision adalah ilmu dan teknologi mesin yang melihat, di mana mesin mampu mengekstrak informasi dari gambar yang diperlukan untuk menyelesaikan tugas tertentu. Sebagai suatu disiplin ilmu, visi komputer berkaitan dengan teori di balik sistem buatan bahwa ekstrak informasi dari gambar. Data gambar dapat mengambil banyak bentuk, seperti urutan video, pandangan dari beberapa kamera, atau data multi-dimensi dari scanner medis. Computer Vision berusaha untuk menerapkan teori dan model untuk pembangunan sistem [12]. Pada Computer Vision terdapat kombinasi antara Pengolahan Citra dan Pengenalan Pola yang hubungannya dapat dilihat pada gambar 2.6



**Gambar 1.6** Kombinasi Pengolahan Citra dan Pengenalan Pola

Pengolahan Citra (Image Processing) merupakan bidang yang berhubungan dengan proses transformasi citra atau gambar. Proses ini bertujuan untuk mendapatkan kualitas citra yang lebih baik. Sedangkan Pengenalan Pola (Pattern Recognition), bidang ini berhubungan dengan proses identifikasi obyek pada citra atau interpretasi citra. Proses ini bertujuan untuk mengekstrak informasi atau pesan yang disampaikan oleh gambar atau citra [12].

## 2.4 Jenis Citra

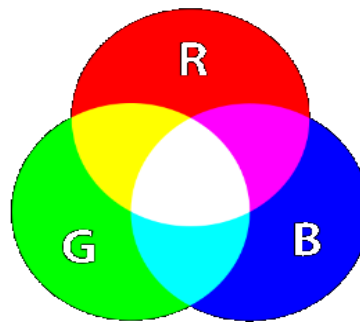
Nilai suatu *pixel* memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0 – 255. Citra dengan penggambaran seperti ini digolongkan kedalam citra integer. Berikut adalah jenis-jenis citra berdasarkan nilai *pixel*nya [10].

### 2.4.1 Citra RGB

RGB sering disebut sebagai warna additive. Hal ini karena warna dihasilkan oleh cahaya yang ada. Beberapa alat yang menggunakan color model RGB antara lain; mata manusia, projector, TV, kamera video, kamera digital, dan alat-alat yang menghasilkan cahaya. Proses pembentukan cahayanya adalah dengan mencampur

ketiga warna tadi. Skala intensitas tiap warnanya dinyatakan dalam rentang 0 sampai 255.

Ketika warna Red memiliki intensitas sebanyak 255, begitu juga dengan Green dan Blue, maka terjadilah warna putih. Sementara ketika ketiga warna tersebut mencapai intensitas 0, maka terjadilah warna hitam, sama seperti ketika berada di ruangan gelap tanpa cahaya, yang tampak hanya warna hitam. Hal ini bisa dilihat ketika menonton di bioskop tua di mana proyektor yang digunakan masih menggunakan proyektor dengan 3 warna dari lubang yang terpisah, bisa terlihat ketika film menunjukkan ruangan gelap, cahaya yang keluar dari ketiga celah proyektor tersebut berkurang [2].



**Gambar 1.7** Warna RGB

#### **2.4.2 Citra GrayScale**

Citra grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian RED=GREEN=BLUE. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan dan putih. Tingkat keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Citra grayscale berikut memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan) [10].





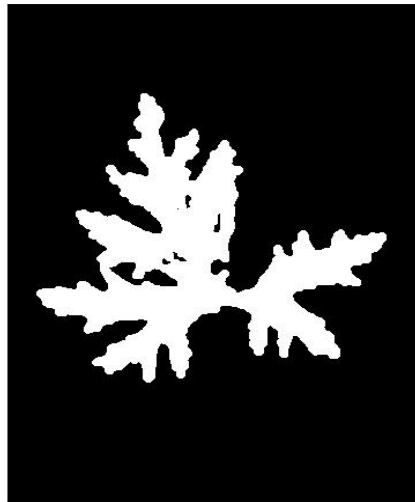
**Gambar 1.8** Citra Grayscale

### **2.4.3 Citra Biner**

Citra biner (binary image) adalah citra yang hanya mempunyai dua nilai derajat keabuan: hitam dan putih. Meskipun saat ini citra berwarna lebih disukai karena memberi kesan yang lebih kaya daripada citra biner, namun tidak membuat citra biner mati. Pada beberapa aplikasi citra biner masih tetap dibutuhkan, misalnya citra logo instansi (yang hanya terdiri atas warna hitam dan putih), citra kode batang (bar code) yang tertera pada label barang, citra hasil pemindaian dokumen teks, dan sebagainya. Bab ini akan memaparkan beberapa konsep dan teknik pengolahan citra biner. Alasan penggunaan citra biner adalah karena ia memiliki sejumlah keuntungan sebagai berikut:

1. Kebutuhan memori kecil karena nilai derajat keabuan hanya membutuhkan representasi 1 bit. Kebutuhan memori untuk citra biner masih dapat berkurang secara berarti dengan metode pemampatan run-length encoding (RLE). Metode RLE akan dijelaskan kemudian.

2. Waktu pemrosesan lebih cepat dibandingkan dengan citra hitam-putih karena banyak operasi pada citra biner yang dilakukan sebagai operasi logika (AND, OR, NOT, dll) daripada operasi aritmetika bilangan bulat.



**Gambar 1.9** Citra Biner

## 2.5 Pemrosesan Data Awal (*pre-processing*)

### 2.5.1 Konversi Citra ke Grayscale

Merubah citra RGB menjadi citra grayscale adalah salah satu contoh proses pengolahan citra menggunakan operasi titik. Untuk mengubah citra RGB menjadi citra grayscale adalah dengan menghitung rata-rata nilai intensitas RGB setiap pixel penyusun tersebut. Rumusan matematis yang digunakan adalah:

$$\text{Citra Abu-Abu} = \text{rgb2gray}(a)$$

Dimana :

$a$  = gambar yang di konversi ke *gray*

### 2.5.2 Operasi Pointwise

Karena citra digital adalah matriks, maka operasi-operasi aritmetika matriks juga berlaku pada citra. Operasi matriks yang dapat dilakukan adalah:

1. Penjumlahan atau pengurangan antara dua buah citra A dan B:

$$C(x, y) = A(x, y) \pm B(x, y),$$

2. Perkalian dua buah citra:

$$C(x, y) = A(x, y) B(x, y),$$

3. Penjumlahan/pengurangan citra A dengan skalar c:

$$B(x, y) = A(x, y) \pm c,$$

4. Perkalian/pembagian citra A dengan sebuah skalar c:

$$B(x, y) = c \times A(x, y)$$

Perkalian antar matriks dapat dilihat pada contoh sederhana berikut:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \times \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}$$

Namun untuk penelitian ini, digunakan Operasi *Pointwise* atau perkalian element per element untuk citra Grayscale [2.7] dan Citra Biner [2.8]. Bila kita ingin melakukan perkalian pointwise, ada beberapa kebingungan yang bisa muncul. Pada kasus pointwise, kita ingin mengalikan matriks secara elemen per elemen, jadi mereka harus memiliki dimensi yang sama. Sebagai contoh, dua matriks 5 x 8 dapat dikalikan secara pointwise, walaupun keduanya tidak bisa melakukan perkalian matriks biasa. Untuk melakukan perkalian pointwise pada MATLAB, kita menggunakan operator “point-star”  $A .* B$ . Misalnya bila A dan B keduanya adalah matriks 3 x 2 maka

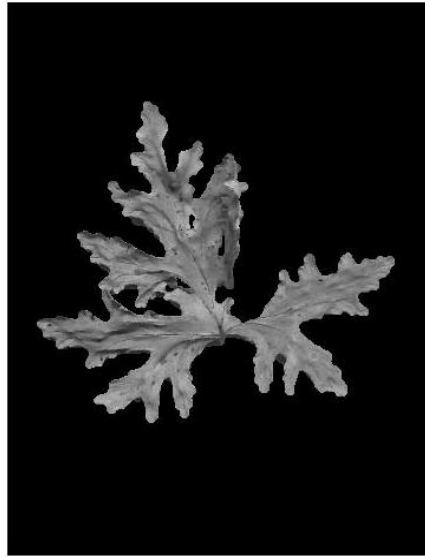
$$C = A .* B = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} .* \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} = \begin{bmatrix} a_{11} * b_{11} & a_{12} * b_{12} \\ a_{21} * b_{21} & a_{22} * b_{22} \\ a_{31} * b_{31} & a_{32} * b_{32} \end{bmatrix}$$

Untuk selanjutnya, perkalian semacam ini kita sebut dengan istilah perkalian array. Perhatikan bahwa perkalian array bersifat komutatif karena kita akan memperoleh hasil yang sama bila kita menghitung  $D = B .* A$ .

Di dalam MATLAB, bila sebuah “titik” digunakan dengan operator aritmetik, maka ia akan mengubah definisi operator tersebut ke operasi pointwise. Jadi operator  $./$  berarti pembagian pointwise,  $.^$  berarti

pemangkatan pointwise. Misalnya,  $xx = (0,9).^n$  akan menghasilkan suatu vector yang nilainya sama dengan  $(0,9)^n$  untuk  $n = 0, 1, 2, \dots, 49$ .

Untuk hasil Citra dari perkalian antar elemen dapat dilihat pada gambar dibawah ini



**Gambar 1.10** Citra Hasil Operasi Pointwise

## 2.6 Analisis Bentuk

Pengertian bentuk adalah teknik untuk mempresentasikan bentuk objek yang akan diteliti, sebuah representasi yang baik akan dapat menggambarkan karakteristik intrinsic dari sebuah shape secara eksplisit. Representasi sebuah shape juga harus invariant terhadap rotasi, scaling dan transformasi.

Ekstraksi fitur bentuk pada skripsi ini terdapat 4 fitur, meliputi:

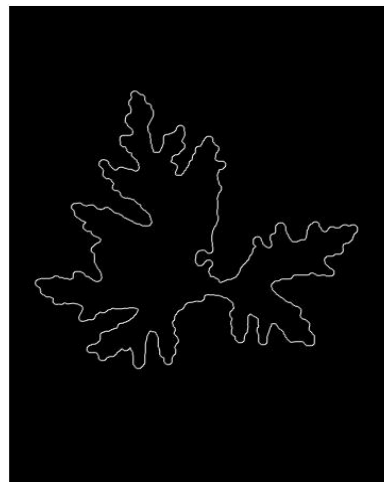
- Aspek Rasio

Pengertian aspek rasio (aspect ratio) suatu citra adalah angka yang menunjukkan perbandingan panjang dan lebar suatu bidang gambar yang dinyatakan sebagai panjang/lebar. Aspek rasio paling umum digunakan dalam dunia digital untuk menyatakan proporsi format tampilan image digital dan proporsi layar (screen) digital. Karena besaran panjang dan lebar suatu gambar atau tampilan memiliki satuan yang sama (seperti pixel, cm, inch, dsb) maka rasio aspek tidak memiliki satuan. Misalnya

sebuah image foto digital memiliki panjang 400 pixel dan lebar 300 pixel maka rasio aspek citra tersebut adalah 4:3.

- Perimeter

Perimeter merupakan bagian terluar dari suatu objek bersebelahan dengan piksel-piksel dari latar belakang. Nilai perimeter suatu objek dapat dicari dengan menghitung banyaknya piksel yang merupakan piksel-piksel yang berada pada perbatasan dari objek tersebut. Contoh Gambar berikut merupakan hasil perimeter dari suatu object daun.



**Gambar 1.2** Perimeter Image

- Indeks Kebulatan

Faktor kebulatan dapat digunakan untuk menggolongkan bentuk objek yang dihubungkan dengan bentuk bundar dan memanjang seperti banyak dijumpai pada jenis buah-buahan. Lingkaran adalah bentuk rasio paling bundar dengan nilai resiko sekitar 1, untuk bujur sangkar nilainya  $\frac{\pi}{4}$  semakin mengecil ketika objek berbentuk memanjang (elips atau kotak).

- Faktor Kekompakan (Compactness)

Salah satu pengolahan citra biner adalah Compactness (kekompakan). Compactness bertujuan untuk mengidentifikasi bentuk dan ukuran object yang sama tetapi dengan profik tepi yang berbeda. Dalam penghitungan compactness suatu citra diperhatikan area dan

perimeter citra. Area (**A**) merupakan jumlah pixel penyusun objek. satuan dari area adalah pixel. Perimeter (**P**) adalah panjang perbatasan object tersebut. Besar perimeter merupakan jumlah pixel yang terdapat pada batas object tersebut [21].

## 2.7 Analisis Tekstur

Tekstur merupakan karakteristik intrinsik dari suatu citra yang terkait dengan tingkat kekasaran (roughness), granularitas (granulation), dan keteraturan (regularity) susunan struktural piksel. Aspek tekstural dari sebuah citra dapat dimanfaatkan sebagai dasar dari segmentasi, klasifikasi, maupun interpretasi citra. Tekstur dapat didefinisikan sebagai fungsi dari variasi spasial intensitas piksel (nilai keabuan) dalam citra. Berdasarkan strukturnya, tekstur dapat diklasifikasikan dalam dua golongan [20] :

- Makrostruktur

Tekstur makrostruktur memiliki perulangan pola lokal secara periodik pada suatu daerah citra, biasanya terdapat pada pola-pola buatan manusia dan cenderung mudah untuk direpresentasikan secara matematis.

- Mikrostruktur

Pada tekstur mikrostruktur, pola-pola lokal dan perulangan tidak terjadi begitu jelas, sehingga tidak mudah untuk memberikan definisi tekstur yang komprehensif.

Analisis tekstur bekerja dengan mengamati pola ketetanggaan antar piksel dalam domain spasial. Dua persoalan yang seringkali berkaitan dengan analisis tekstur adalah

- Ekstraksi Ciri

Ekstraksi ciri merupakan langkah awal dalam melakukan klasifikasi dan interpretasi citra. Proses ini berkaitan dengan kuantisasi karakteristik citra ke dalam sekelompok nilai ciri yang sesuai. Dalam praktikum ini kita akan mengamati metoda ekstraksi ciri statistik orde pertama dan kedua, serta mengenali performansi masing-masing skema dalam mengenali citra dengan karakteristik tekstural yang berlainan.

- Segmentasi Citra

Segmentasi citra merupakan proses yang bertujuan untuk memisahkan suatu daerah pada citra dengan daerah lainnya. Berbeda dengan pada citra non-tekstural, segmentasi citra tekstural tidak dapat didasarkan pada intensitas piksel per piksel, tetapi perlu mempertimbangkan perulangan pola dalam suatu wilayah ketetanggaan lokal. Dalam praktikum ini kita akan mencoba menerapkan filter Gabor untuk melakukan segmentasi citra tekstural berdasarkan perulangan pola lokal pada orientasi dan frekuensi tertentu.

### 2.7.1 Ekstraksi Ciri Statistik

Analisis tekstur lazim dimanfaatkan sebagai proses antara untuk melakukan klasifikasi dan interpretasi citra. Suatu proses klasifikasi citra berbasis analisis tekstur pada umumnya membutuhkan tahapan ekstraksi ciri, yang dapat terbagi dalam tiga macam metode berikut:

- Metode statistik

Metode statistik menggunakan perhitungan statistik distribusi derajat keabuan (histogram) dengan mengukur tingkat kontras, granularitas, dan kekasaran suatu daerah dari hubungan ketetanggaan antar piksel di dalam citra.

Paradigma statistik ini penggunaannya tidak terbatas, sehingga sesuai untuk tekstur-tekstur alami yang tidak terstruktur dari sub pola dan himpunan aturan (mikrostruktur).

- Metode spektral

Metode spektral berdasarkan pada fungsi autokorelasi suatu daerah atau *power distribution* pada domain transformasi Fourier dalam mendeteksi periodisitas tekstur.

- Metode struktural

Analisis dengan metode ini menggunakan deskripsi primitif tekstur dan aturan sintaktik. Metode struktural banyak digunakan untuk pola-pola makrostruktur.

### 2.7.1.1 Ekstraksi Ciri Orde Pertama

Ekstraksi ciri orde pertama merupakan metode pengambilan ciri yang didasarkan pada karakteristik histogram citra. Histogram menunjukkan probabilitas kemunculan nilai derajat keabuan piksel pada suatu citra. Dari nilai-nilai pada histogram yang dihasilkan, dapat dihitung beberapa parameter ciri orde pertama, antara lain *adalah mean, skewness, variance, kurtosis, dan entropy*.

#### a) Mean

Menunjukkan ukuran dispersi dari suatu citra

$$\mu = \sum_n f_n P(f_n) \dots\dots\dots(2.13).$$

dimana  $f_n$  merupakan suatu nilai intensitas keabuan, sementara  $p(f_n)$  menunjukkan nilai histogramnya (probabilitas kemunculan intensitas tersebut pada citra).

#### b) Variance

Menunjukkan variasi elemen pada histogram dari suatu citra.

$$\sigma^2 = \sum_n (f_n - \mu)^2 p(f_n) \dots\dots\dots(2.14)$$

#### c) Kurtosis

Menunjukkan tingkat keruncingan relatif kurva histogram dari suatu citra.

$$\alpha_4 = \frac{1}{\sigma^4} \sum_n (f_n - \mu)^4 p(f_n) - 3 \dots\dots\dots(2.15)$$

#### d) Skewness

Menunjukkan tingkat kemencengan relatif kurva histogram dari suatu citra.

$$\alpha_4 = \frac{1}{\sigma^4} \sum_n (f_n - \mu)^4 p(f_n) \dots\dots\dots(2.16)$$

### 2.7.1.2 Ekstraksi Ciri Orde Dua

Pada beberapa kasus, ciri orde pertama tidak lagi dapat digunakan untuk mengenali perbedaan antar citra. Pada kasus



seperti ini, kita membutuhkan pengambilan ciri statistik orde dua.

Salah satu teknik untuk memperoleh ciri statistik orde dua adalah dengan menghitung probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan orientasi sudut tertentu. Pendekatan ini bekerja dengan membentuk sebuah matriks kookurensi dari data citra, dilanjutkan dengan menentukan ciri sebagai fungsi dari matriks antara tersebut.

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 dengan interval sudut  $45^\circ$ , yaitu  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ , dan  $135^\circ$ . Sedangkan jarak antar piksel biasanya ditetapkan sebesar 1 piksel.

Matriks kookurensi merupakan matriks bujursangkar dengan jumlah elemen sebanyak kuadrat jumlah level intensitas piksel pada citra. Setiap titik  $(p,q)$  pada matriks kookurensi berorientasi  $\theta$  berisi peluang kejadian piksel bernilai  $p$  bertetangga dengan piksel bernilai  $q$  pada jarak  $d$  serta orientasi  $\theta$  dan  $(180-\theta)$ .

Setelah memperoleh matriks kookurensi tersebut, kita dapat menghitung ciri statistik orde dua yang merepresentasikan citra yang diamati. Haralick et al mengusulkan berbagai jenis ciri tekstural yang dapat diekstraksi dari matriks kookurensi. Dalam modul ini dicontohkan perhitungan 6 ciri statistik orde dua, yaitu *Angular Second Moment*, *Contrast*, *Correlation*, *Variance*, *Inverse Difference Moment*, dan *Entropy*.

#### a. *Angular Second Moment*

Menunjukkan ukuran sifat homogenitas citra.

$$ASM = \sum_i \sum_j \{ p(i, j) \}^2 \dots\dots\dots (2.17)$$

b. *Contrast*

Menunjukkan ukuran 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.

$$CON = \sum_k k^2 \left[ \sum_i \sum_j p(i, j) \right]_{|i-j|=k} \dots\dots\dots(2.18)$$

c. *Correlation*

Menunjukkan ukuran ketergantungan linear derajat keabuan citra sehingga dapat memberikan petunjuk adanya struktur linear dalam citra.

$$COR = \frac{\sum_i \sum_j (ij) \cdot p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \dots\dots\dots(2.19)$$

d. *Variance*

Menunjukkan variasi elemen-elemen matriks kookurensi. Citra dengan transisi derajat keabuan kecil akan memiliki variansi yang kecil pula.

$$VAR = \sum_i \sum_j (i - \mu_x)(j - \mu_y) p(i, j) \dots\dots\dots(2.20)$$

e. *Inverse Different Moment*

Menunjukkan kehomogenan citra yang berderajat keabuan sejenis. Citra homogen akan memiliki harga IDM yang besar.

$$IDM = \sum_i \sum_j \frac{1}{1+(i-j)^2} p(i, j) \dots\dots\dots(2.21)$$

f. *Entropy*

Menunjukkan ukuran ketidakteraturan bentuk. Harga ENT besar untuk citra dengan transisi derajat keabuan merata dan bernilai kecil jika struktur citra tidak teratur (bervariasi).

$$ENT_2 = - \sum_i \sum_j p(i, j) \cdot {}^2 \log p(i, j) \dots\dots\dots(2.22)$$

## 2.8 Euclidean

Euclidean digunakan untuk menentukan perhitungan jarak terdekat nilai vector ciri citra uji dengan citra acuan. Nilai Euclidean yang mendekati nilai nol, akan menunjukkan pada citra tertentu. Nilai vector ciri citra masukan yang memiliki nilai vector ciri yang sama dengan vector ciri citra tertentu akan memiliki nilai Euclidean yang mendekati nol.

Rumus menghitung Euclidean sebagai berikut:

$$d_j = \sqrt{\sum_{i=1}^n (X_{ij} - P_j)^2} \dots\dots\dots(2.23)$$

Keterangan :

$d_j$  : jarak sampel

$X_{ij}$  : data sampel pengetahuan

$P_j$  : data input varial ke-j

$N$  : jumlah sample

Berikut adalah contoh perhitungan menggunakan Euclidean:

Hitung = SQRT(data\_Latih – data\_Uji)<sup>2</sup>.

## 2.9 Metode K-Nearest Neighbor

Algoritma *k-nearest neighbor* (k-NN atau KNN) adalah sebuah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat dengan objek tersebut.

*K-Nearest Neighbor* berdasarkan konsep '*learning by analogy*'. Data *learning* dideskripsikan dengan atribut numerik *n*-dimensi. Tiap data *learning* merepresentasikan sebuah titik, yang ditandai dengan *c*, dalam ruang *n*-dimensi. Jika sebuah data *query* yang labelnya tidak diketahui diinputkan, maka *K-Nearest Neighbor* akan mencari *k* buah data *learning* yang jaraknya paling dekat dengan data *query* dalam ruang *n*-dimensi. Jarak antara data *query* dengan data *learning* dihitung dengan cara mengukur jarak antara titik yang merepresentasikan data *query* dengan semua titik yang merepresentasikan data *learning* dengan rumus Euclidean Distance.

Pada fase training, algoritma ini hanya melakukan penyimpanan vektor-vektor fitur dan klasifikasi data training sample. Pada fase klasifikasi, fitur-fitur yang sama dihitung untuk testing data (yang klasifikasinya tidak diketahui). Jarak dari vektor baru yang ini terhadap seluruh vektor training sample dihitung dan sejumlah  $k$  buah yang paling dekat diambil. Titik yang baru klasifikasinya diprediksikan termasuk pada klasifikasi terbanyak dari titik-titik tersebut.

Sebagai contoh, untuk mengestimasi  $p(x)$  dari  $n$  training sample dapat memusatkan pada sebuah sel disekitar  $x$  dan membiarkannya tumbuh hingga meliputi  $k$  samples. Samples tersebut adalah KNN dari  $x$ . Jika densitasnya tinggi di dekat  $x$ , maka sel akan berukuran relatif kecil yang berarti memiliki resolusi yang baik. Jika densitas rendah, sel akan tumbuh lebih besar, tetapi akan berhenti setelah memasuki wilayah yang memiliki densitas tinggi.

Nilai  $k$  yang terbaik untuk algoritma ini tergantung pada data. Secara umum, nilai  $k$  yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi semakin kabur. Nilai  $k$  yang bagus dapat dipilih dengan optimasi parameter, misalnya dengan menggunakan *cross-validation*. Kasus khusus dimana klasifikasi diprediksikan berdasarkan *training data* yang paling dekat (dengan kata lain,  $k = 1$ ) disebut algoritma *nearest neighbor*.

Ketepatan algoritma KNN sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap klasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur agar performa klasifikasi menjadi lebih baik.

KNN memiliki beberapa kelebihan yaitu ketangguhan terhadap *training data* yang memiliki banyak *noise* dan efektif apabila *training data*-nya besar. Sedangkan, kelemahan KNN adalah KNN perlu menentukan nilai dari parameter  $k$  (jumlah dari tetangga terdekat), *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*.

Konsep dasar dari KNN adalah mencari jarak terdekat antara data yang akan dievaluasi dengan K tetangga terdekatnya dalam data pelatihan. Jumlah kelas yang paling banyak dengan jarak terdekat tersebut akan menjadi kelas dimana data data evaluasi tersebut berada.

Algoritma KNN :

1. Mulai
2. Tentukan parameter K
3. Hitung jarak antara data yang akan dievaluasi dengan semua pelatihan.

Jarak ini didapatkan dari data yang telah di ekstraksi dengan metode *Analisis Frekuensi, Tekstur dan bentuk* yaitu data pelatihan dikurangi data baru yang akan di uji yang disebut dengan jarak *Euclidian*.

4. Urutkan jarak yang terbentuk (urut naik)  
Setelah diketahui jarak setiap data pelatihan, maka data di urutkan atau di sorting mulai dari data terkecil sampai data yang paling besar lalu cari jarak yang paling kecil (terdekat) dengan data pelatihan.
5. Tentukan jarak terdekat sampai urutan K
6. Pasangkan kelas yang bersesuaian  
Setelah di tentukan K terdekatnya berapa, maka bisa di ketahui beberapa kelas yang akan di tujukan pada data uji tersebut.
7. Cari jumlah kelas dan tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi
8. Selesai.[Syamani, 2008]

## 2.10 Normalisasi Data

Setelah mendapatkan data-data yang dibutuhkan dari semua jenis daun, langkah selanjutnya adalah menormalisasi beberapa data guna proses klasifikasi kedepan. Normalisasi adalah suatu teknik untuk mengorganisasi data ke dalam tabel-tabel untuk memenuhi kebutuhan pemakai di dalam suatu

organisasi, atau proses penskalaan nilai atribut dari data sehingga bisa jatuh pada range tertentu. Sebagai contoh: Misalnya berkenaan dengan pencatatan tingkat kematian penduduk di Indonesia per-bulannya berdasarkan jenis umur. Secara sederhana, disana ada 3 dimensi data, yaitu bulan (1-12), umur (0-150 misalnya), dan jumlah kematian (0-jutaan). Kalau kita bentangkan range masing-masing dimensi, maka kita akan mendapatkan ketidakseimbangan range pada dimensi yang ketiga, yaitu jumlah kematian [22].

Normalisasi yang akan digunakan dalam penelitian ini menggunakan metode min-max normalization. Keuntungan dari metode ini adalah keseimbangan nilai perbandingan antar data saat sebelum dan sesudah proses normalisasi. Tidak ada data bias yang dihasilkan oleh metode ini. Kekurangannya adalah jika ada data baru, metode ini akan memungkinkan terjebak pada "out of bound" error. Rumus metode min-max normalization dapat dilihat dibawah ini.

$$s' = \frac{s - \{s_k\}}{\max\{s_k\} - \min\{s_k\}} \dots\dots\dots(2.24)$$

## 2.11 Penelitian Sebelumnya

1. Penelitian yang dilakukan oleh **Pedro Felipe Barros Silva** yang berjudul *Development of a System for Automatic Plant Species Recognition* dari **Departemen Matematika Fakultas Ilmu Pengetahuan Universitas Porto**. Nilai akurasi metode KNN yang digunakan dalam penelitian tersebut adalah sebesar 75.36% dengan membandingkan skenario pengujian yang sama dalam skripsi ini.
2. Penelitian yang dilakukan oleh tim gabungan dari ITS yang diwakili oleh **R.V. Hari Ginardi** beserta **Handayani Tjandrasa** dan dari Universitas Muhammadiyah Gresik yang diwakili oleh **Soffiana Agustin** dengan judul penelitian *Identification of Oil Palm Plantation in Ikonos Images using Radially Averaged Power Spectrum Values*. Penambahan fitur frekuensi pada objek citra dapat membantu meningkatkan akurasi sampai 28%.

3. ***KLASIFIKASI UMUR LAHAN PERKEBUNAN KELAPA SAWIT MENGGUNAKAN METODE LOCAL BINARY PATTERN (LBP).***

Skripsi oleh **Muhammad Subchan** yang menggabungkan LBP sebagai ekstraksi ciri fitur dengan metode K-NN sebagai klasifikasi umur kelapa sawit. Dengan tingkat keakurasian pada K=3 mencapai 87%.