

BAB II

LANDASAN TEORI

2.1. Citra Candling

Citra *candling* telur merupakan citra yang didapatkan melalui proses *candling* (peneropongan) pada telur. Proses ini berupa analisa visual dalam peneropongan pada telur yang diberikan cahaya pada salah satu sisi didalam ruang gelap atau sering disebut *candling*. Pengambilan citra *candling* dilakukan dengan menggunakan lampu pijar 25-60 watt atau diarahkan pada cahaya. Hal ini bertujuan untuk mengetahui telur tersebut *fertil* atau *infertil* berdasarkan dari terlihatnya pembuluh darah (noktah merah) pada telur (Isnawati, 2016).

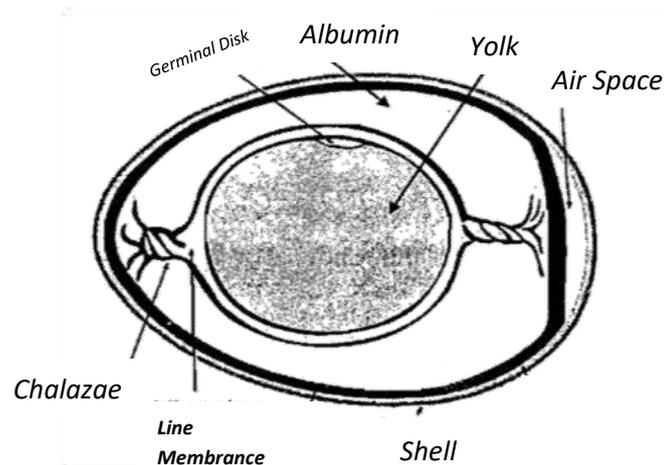
Adapun tahapan yang dilalui untuk mendapatkan citra dengan proses *candling* sebagai berikut :

- a) Siapkan kotak atau kardus berbentuk kubus yang dilapisi kertas hitam kemudian buat dua buah lubang lubang yang pertama ada bagian sisi bawah yang berfungsi tempat lampu atau senter *led* 25-60 watt. Lubang kedua berada di sisi atas yang berfungsi sebagai lubang camera untuk proses pembidikan menggunakan camera digital dengan ukuran 14 piksel.
- b) Langkah selanjutnya adalah proses peneropongan dengan meletakkan telur diatas lampu *led* dan pembidikan telur dapat dilakukan dengan memastikan hasil bidikan sesuai apa yang dibutuhkan. Adapun telur yang diteropong minimal hari ke 3-4 hari setelah pengeraman di mesin tetas, karena pada hari tersebut noktah merah mulai muncul.
- c) Setelah peneropongan dan pembidikan selesai, langkah berikutnya adalah memindahkan data gambar ke *computer* untuk di olah dan dianalisa cirinya.

Menurut Rohman Dijaya (2016), peneropongan telur pada usia 0 hari sampai hari ke 3 sulit untuk dideteksi, karena noktah atau embrio telur terlihat tipis atau samar-samar, sehingga akan sulit membedakan telur *fertil* dan *infertil*, akibatnya telur *fertil* dijadikan telur konsumsi atau sebaliknya. Waktu yang baik digunakan untuk melakukan *candling* dengan tujuan mengetahui kesuburan telur dilakukan pada hari ke 3-6 proses inkubasi (inkubasi dengan mesin tetas) dengan

melihat nokta merah pada telur, sedangkan waktu *candling* pada hari ke 12-18 proses inkubasi dilakukan untuk melihat pertumbuhan ataupun kematian dari *embrio*.

2.2. Telur Fertil dan Infertil



Gambar 2.1 Bagian-bagian telur (Faridah,2008)

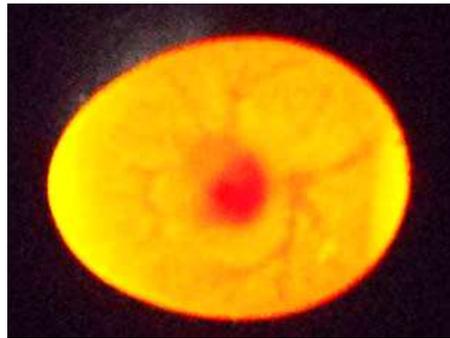
Menurut Paula Figoni (2008), telur memiliki beberapa komponen didalamnya yaitu:

- 1) Putih telur (Albumin)
- 2) Kuning telur (Yolk)
- 3) Kulit telur (Shell)
- 4) Rongga udara (*Air Cell*)
- 5) Chalazae (tali)
- 6) Zigot/sel embrio (*Germinal Disk*)
- 7) Dinding garis (*Line Membrane*)

Telur *fertil* adalah telur yang mengalami perkembangan embrio, berawal dari sel benih betina yang sudah dibuahi yang dinamakan *zygot blastoderm* terdapat di kuning telur, selanjutnya *blastoderm* akan muncul darinya garis-garis warna merah yang merupakan petunjuk mulainya sistem sirkulasi darah, dan organ penting tubuh mulai terbentuk menjadi *embrio*. Perkembangan ini akan merubah komposisi warna kuning telur menjadi lebih gelap saat dilakukan prosesi *candling*. Sedangkan telur *infertil* adalah telur yang tidak mengalami perkembangan embrio

pada proses penetasannya, dan warna kuning telur lebih cerah saat *dicandling*. Banyak faktor yang mempengaruhi penetasan telur tapi 7.5 % dari 100% penetasan gagal menetas dikarenakan telur tersebut *infertil* (Lawrence,etal.,2008).

a. Telur Fertil



Gambar 2.2 Cir-ciri telur *fertil*

Gambar 2.2 di atas merupakan telur *fertil*, terlihat pada dinding telur muncul garis-garis merah atau adanya noktah merah yang merupakan pertanda adanya embrio *candling* pertama, ketika telur sudah dierami selama 3-4 hari. Telur yang subur kebanyakan bentuk telurnya oval sedangkan bentuk telur yang bulat atau terlalu lonjong merupakan telur abnormal sehingga tingkat *infertil* telur dengan bentuk tersebut tinggi.

b. Telur Infertil



Gambar 2.3 ciri-ciri telur *infertil*

Pada gambar 2.3 telur tidak tampak noktah merah, yang berarti tidak subur karena tidak ada *embrio* alias *infertil*. Gambar ini bisa dilihat ketika telur sudah dierami selama 3-4 hari. Terkadang yang terlihat bukan telur *infertil*, bukan juga telur *fertil*, tetapi seperti terlihat pada gambar. Ada titik darah (semacam noktah), tetapi tidak ada pembuluh darah. Ini menunjukkan telur yang sebenarnya *fertil*, tetapi *embrio* mati muda di dalam telur.

2.3. Pengolahan Citra dan Jenis Citra

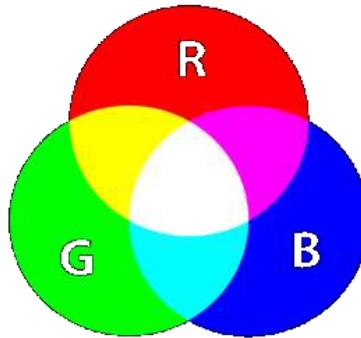
Pengolahan Citra (*Image Processing*) merupakan bidang yang berhubungan dengan proses transformasi citra atau gambar. Proses ini bertujuan untuk mendapatkan kualitas citra yang lebih baik. Sedangkan Pengenalan Pola (*Pattern Recognition*), bidang ini berhubungan dengan proses identifikasi obyek pada citra atau interpretasi citra. Proses ini bertujuan untuk mengekstrak informasi atau pesan yang disampaikan oleh gambar atau citra. Jenis Citra adalah 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 [PDP10].

2.3.1 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 [Novi, D.E.2012].



Gambar 2.4 Warna RGB

2.3.2 Citra Gray

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) [Putra,D.2010].



Gambar 2.5 Citra *Grayscale*

2.3.3 Citra biner

Citra biner adalah ciitra digital yang hanya memiliki dua kemungkinan nilai pixel yaitu hitam dan putih. Citra biner juga disebut sebagai citra B&W (*black dan white*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap pixel dari citra biner.

Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti segmentasi, pengambangan, morfologi, ataupun dithering [PDP10].



Gambar 2.6 Citra Biner

2.4 Pemrosesan Data Awal

2.4.1 Konversi Citra RGB 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:

$$\text{Citra Abu-Abu} = \text{rgb2gray}(a) \dots \dots \dots (2.1)$$

Dimana : a = gambar yang di konversi ke *gray*

2.4.2 Konversi Gray ke Biner

Citra biner (*binary image*) adalah citra yang setiap pikselnya hanya memiliki dua kemungkinan derajat keabuan yakni 0 dan 1. Proses pembineran dilakukan dg membulatkan keatas atau kebawah untuk

setiap nilai keabuan dari piksel yang berada diatas atau bawah harga ambang.

$$\text{Citra biner} = \text{im2bw}(a); \dots\dots\dots(2.2)$$

Dimana : a = gambar yang di konversi ke *biner*

2.4.3 Ekualisasi Histogram

Ekualisasi histogram adalah suatu tehnik perbaikan citra yang cara memanipulasi masing-masing piksel citra. Proses ini dilakukan dengan menggunakan metode CLAHE (Contrast-Limited Adaptive Histogram Equalization). Oleh karena itu *histeq* (*histogram Equalisasi*) disebut bekerja dibidang spasial, dengan *histogram equalisasi* kontras citra di *stretch* (diregangkan), sehingga titik atau pixel yang gelap semakin gelap sedangkan yang terang semakin terang.

$$\text{Citra Histogram} = \text{histeq}(a); \dots\dots\dots(2.3)$$

Dimana : a = gambar yang di konversi ke *biner*



Gambar 2.7. Perbandingan Image Sebelum Dan Setelah Dilakukan Ekualisasi Histogram

2.4.4 Segmentasi Citra

Segmentasi citra adalah pemisahan objek yang satu dengan objek yang lain dalam suatu citra atau antara objek dengan latar dalam sebuah citra. Dengan proses segmentasi tersebut, masing-masing objek pada citra dapat di ambil secara individu sehinga dapat di gunakan sebagai input proses lain. Ada dua macam segmentasi, yaitu *full segmentation*, dan *partial segmentation*. Full segmentation adalah pemisahan objek secara individu dari *background* dan diberi Id (label) pada tiap-tiap segment. *partial segmentation* adalah pemisahan sejumlah data dari background di mana data yang di

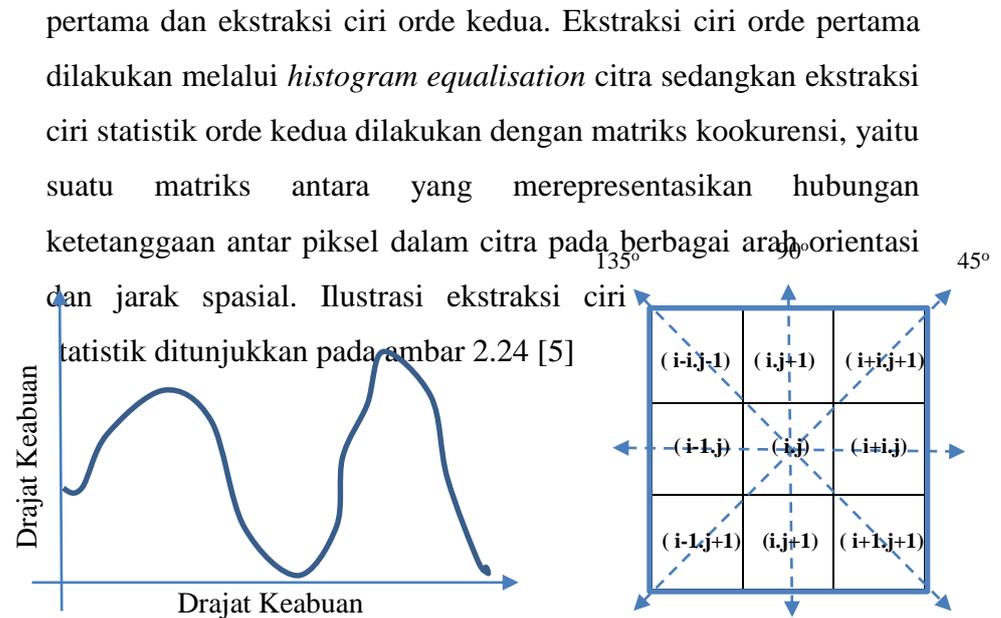
simpan hanya data yang di pisahkan saja hanya untuk mempercepat proses selanjutnya. [cahyan,2013]

Terdapat dua pendekatan utama dalam segmentasi citra yaitu didasarkan pada tepi (*edge-based*) dan didasarkan pada wilayah (*region-based*). Segmentasi didasarkan pada tepi membagi citra berdasarkan diskontinuitas di antara sub-wilayah (*sub-region*), sedangkan segmentasi yang didasarkan pada wilayah bekerjanya berdasarkan keseragaman yang ada pada sub-wilayah tersebut. Hasil dari segmentasi citra adalah sekumpulan wilayah yang melingkupi citra tersebut, atau sekumpulan kontur yang diekstrak dari citra (pada deteksi tepi).

2.5 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.

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 terdiri dari tiga macam metode yaitu metode statistik, metode spektral dan metode struktural. Metode GLCM termasuk dalam metode statistik dimana dalam perhitungan statistiknya menggunakan distribusi derajat keabuan (*histogram equalisation*) 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 statistik terdiri dari ekstraksi ciri orde



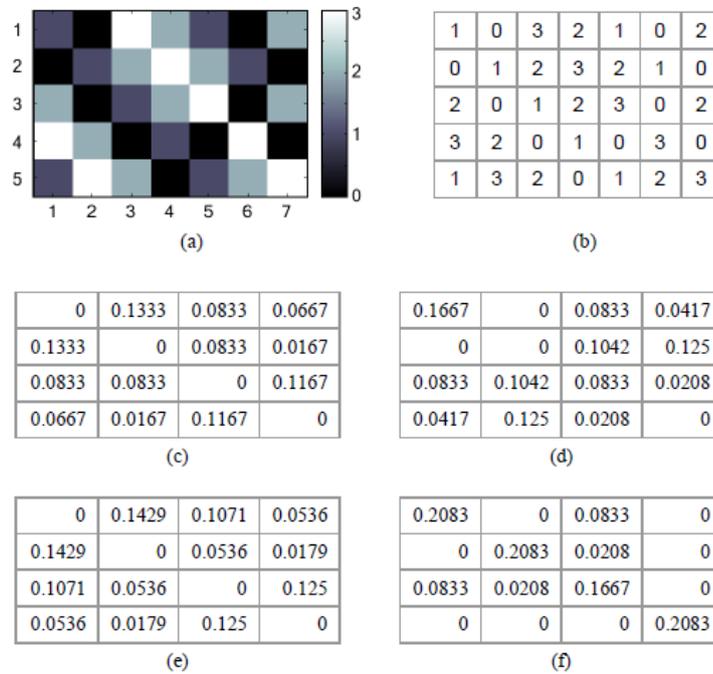
Gambar 2.8. Ilustrasi ekstraksi ciri statistik, (a) Histogram citra sebagai fungsi probabilitas kemunculan nilai intensitas pada citra, (b) Hubungan ketetangaan antar piksel sebagai fungsi orientasi dan jarak spasial

2.5.4 Co-cocurence Matrix

Co-occurrence berarti kejadian bersama, yaitu jumlah kejadian satu level nilai piksel bertetangga dengan satu level nilai piksel lain dalam jarak (d) dan orientasi sudut (θ) tertentu. Jarak dinyatakan dalam piksel dan orientasi dinyatakan dalam derajat. Orientasi dibentuk dalam empat arah sudut dengan interval sudut 45° , yaitu 0° , 45° , 90° , dan 135° . Sedangkan jarak antar piksel biasanya ditetapkan sebesar 1 piksel.

Co-occurrence Matrix merupakan matriks bujursangkar dengan jumlah elemen sebanyak kuadrat jumlah level intensitas piksel pada citra. Setiap titik (p,q) pada co-occurrence matrix berorientasi θ berisi peluang kejadian piksel bernilai p bertetangga

dengan piksel bernilai q pada jarak d serta orientasi θ dan $(180-\theta)$ [DNE12].



Gambar 2.9 Ilustrasi pembuatan matriks kookurensi

- (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 memperoleh matriks kookurensi tersebut, Langkah selanjutnya yakni mencari nilai rata-rata dari sudut 0° , 45° , 90° , dan 135° , hasil perhitungan dapat dilihat pada gambar 2.10

0.0938	0.0690	0.0893	0.0405
0.0690	0.0521	0.0655	0.0399
0.0893	0.0655	0.0625	0.0656
0.0405	0.0399	0.0656	0.0521

Gambar 2.10 Perhitungan nilai rata-rata

Berikut adalah nilai matriks i dan j yang nantinya akan digunakan sebagai pelengkap perhitungan.

i			
1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

j			
1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

(a)
(b)

Gambar 2.11 (a) nilai Matriks variable i (b) nilai matriks variable j

Setelah memperoleh nilai rata-rata, 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* [PCD4].

1. *Angular Second Moment (ASM)*

Menunjukkan ukuran sifat homogenitas citra.

$$ASM = \sum_i \sum_j \{p(i,j)\}^2 \dots\dots\dots(2.4)$$

dimana $p(i,j)$ merupakan menyatakan nilai pada baris i dan kolom j pada matriks kookurensi. Berikut adalah perhitungan nilai ASM

0.0938	0.0690	0.0893	0.0405
0.0690	0.0521	0.0655	0.0399
0.0893	0.0655	0.0625	0.0656
0.0405	0.0399	0.0656	0.0521

0.0088	0.0048	0.0080	0.0016
0.0048	0.0027	0.0043	0.0016
0.0080	0.0043	0.0039	0.0043
0.0016	0.0016	0.0043	0.0027

(a)
(b)

Gambar 2.12 Ilustrasi perhitungan ASM

Keterangan:

- (a) Adalah nilai dari penjumlahan masing-masing sudut dibagi dengan banyaknya sudut (nilai rata-rata)
- (b) Adalah nilai dari masing-masing *pixel* yang sudah dipangkat 2. Jika nilai (b) dijumlahkan, maka hasil yang diperoleh yakni nilai ASM = 0.0672

2. 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. Berikut adalah adalah perhitungan nilai CON

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

.....(2.5)

(i-j) ² = k			
0	1	4	9
1	0	1	4
4	1	0	1
9	4	1	0

k*rata-rata			
0	0.0690	0.3571	0.3643
0.0690	0	0.0655	0.1595
0.3571	0.0655	0	0.0656
0.3643	0.1595	0.0656	0

(a) (b)

Gambar 2.13 Ilustrasi perhitungan nilai *contrast*

Keterangan:

- (a) Hasil pengurangan nilai dari variable *i* dengan nilai variable *j* kemudian dikuadratkan, perhitungan ini digunakan sebagai nilai dari variable *k*
- (b) Hasil perkalian dari nilai variable *k* dengan nilai dari variable rata-rata. Sehingga jika dikerjakan sesuai dengan rumus 2.11 maka nilai yang diperoleh untuk CON = 2.1622

3. *Correlation*

Menunjukkan ukuran ketergantungan linear derajat keabuan citra sehingga dapat memberikan petunjuk adanya struktur linear dalam citra. Berikut adalah perhitungan nilai COR

$$COR = \frac{\sum_i \sum_j (ij) \cdot p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \dots\dots\dots(2.6)$$

Keterangan:

- μ_x : adalah nilai rata-rata elemen kolom pada matriks $p(i,j)$
- μ_y : adalah nilai rata-rata elemen baris pada matriks $p(i,j)$
- σ_x : adalah nilai standar deviasi elemen pada kolom $p(i,j)$
- σ_y : adalah nilai standar deviasi elemen pada baris $p(i,j)$

0.0938	0.0690	0.0893	0.0405
0.0690	0.0521	0.0655	0.0399
0.0893	0.0655	0.0625	0.0656
0.0405	0.0399	0.0656	0.0521

(a)

μ_x	2.3865	σ_x	1.1038
μ_y	2.3865	σ_y	1.1038
$\mu_x * \mu_y$	5.6952	$\sigma_x * \sigma_y$	1.2184

(b)

$i*j$			
1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

(c)

$(i*j)*a$			
0.0938	0.1381	0.2679	0.1619
0.1381	0.2083	0.3929	0.3190
0.2679	0.3929	0.5625	0.7875
0.1619	0.3190	0.7875	0.8333

(d)

Gambar 2.14 Ilustrasi perhitungan nilai *corelation*

Keterangan:

- (a) Adalah nilai dari penjumlahan masing-masing sudut dibagi dengan banyaknya sudut (nilai rata-rata)
- (b) Adalah nilai rata-rata dan standar deviasi
- (c) Adalah nilai perkalian matriks i dengan j

- (d) Adalah nilai dari matriks (c) dikalikan dengan nilai dari matriks (a).
Sehingga jika dikerjakan sesuai dengan rumus 2.12 maka nilai yang diperoleh untuk $COR = 0.1127$

4. Variance

Menunjukkan variasi elemen-elemen matriks kookurensi. Citra dengan transisi derajat keabuan kecil akan memiliki variansi yang kecil pula. Berikut adalah perhitungan nilai VAR

$$VAR = \sum_i \sum_j (i - \mu_x)(j - \mu_y)p(i, j) \dots\dots\dots(2.7)$$

$i - \mu_x$				$j - \mu_y$			
-1.3865	-1.3865	-1.3865	-1.3865	-1.3865	-1.3865	-1.3865	-1.3865
-0.3865	-0.3865	-0.3865	-0.3865	-0.3865	-0.3865	-0.3865	-0.3865
0.6135	0.6135	0.6135	0.6135	0.6135	0.6135	0.6135	0.6135
1.6135	1.6135	1.6135	1.6135	1.6135	1.6135	1.6135	1.6135
(a)				(b)			
$(i - \mu_x) * (j - \mu_y)$							
1.9223	0.5358	-0.8506	-2.2371	0.1802	0.0370	-0.0760	-0.0905
0.5358	0.1494	-0.2371	-0.6236	0.0370	0.0078	-0.0155	-0.0249
-0.8506	-0.2371	0.3764	0.9900	-0.0760	-0.0155	0.0235	0.0650
-2.2371	-0.6236	0.9900	2.6035	-0.0905	-0.0249	0.0650	0.1356
(c)				(d)			

Gambar 2.15 Ilustrasi perhitungan nilai *variance*

Keterangan:

- (a) Adalah hasil pengurangan antara matriks pada variable i dengan nilai pada variable μ_x
 (b) Adalah hasil pengurangan antara matriks pada variable j dengan nilai pada variable μ_y
 (c) Adalah hasil perkalian dari (a) dengan (b)
 (d) Adalah hasil perkalian antara nilai pada hasil (c) dengan nilai rata-rata awal

5. *Inverse Difference Moment*

Menunjukkan kehomogenan citra yang berderajat keabuan sejenis. Citra homogen akan memiliki harga IDM yang besar. Berikut adalah perhitungan nilai IDM

$$IDM = \sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j) \dots\dots\dots(2.8)$$

1+(i-j) ²			
1	2	5	10
2	1	2	5
5	2	1	2
10	5	2	1

$\frac{1}{1 + (i - j)^2} p(i, j)$			
0.0938	0.0345	0.0179	0.0040
0.0345	0.0521	0.0327	0.0080
0.0179	0.0327	0.0625	0.0328
0.0040	0.0080	0.0328	0.0521

(a)
(b)

Gambar 2.16 Ilustrasi perhitungan nilai IDM

Keterangan:

- (a) Adalah hasil 1 ditambah dengan matriks variable *i* dikurang dengan matriks variable *j* kemudian dikuadratkan
- (b) Adalah hasil dari 1 dibagi dengan hasil (a) kemudian dikalikan dengan nilai rata-rata awal. Sehingga jika dikerjakan sesuai dengan rumus 2.14 maka nilai yang diperoleh untuk IDM = 0.5203

6. *Entropy*

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

$$ENT_2 = - \sum_i \sum_j p(i, j) \cdot {}^2\log p(i, j) \dots\dots\dots(2.9)$$

$-\sum_i p(i,j)$			
-0.0938	-0.0690	-0.0893	-0.0405
-0.0690	-0.0521	-0.0655	-0.0399
-0.0893	-0.0655	-0.0625	-0.0656
-0.0405	-0.0399	-0.0656	-0.0521

${}^2\log p(i,j)$			
-3.4150	-3.8563	-3.4854	-4.6268
-3.8563	-4.2630	-3.9329	-4.6482
-3.4854	-3.9329	-4.0000	-3.9296
-4.6268	-4.6482	-3.9296	-4.2630

(a)

(b)

$$-\sum_i p(i,j) * {}^2\log p(i,j)$$

0.3202	0.2663	0.3112	0.1873
0.2663	0.2220	0.2575	0.1854
0.3112	0.2575	0.2500	0.2579
0.1873	0.1854	0.2579	0.2220

(c)

Gambar 2.17 Ilustrasi perhitungan nilai *Entropy*

Keterangan:

- (a) Adalah hasil dari nilai rata-rata awal dengan ditambah min
- (b) Adalah hasil dari ${}^2\log p(i,j)$
- (c) Adalah hasil perkalian antara nilai hasil pada (a) dengan nilai hasil pada (b).

2.6 Square Euclidean

Square euclidean digunakan untuk menentukan perhitungan jarak terdekat nilai vektor ciri citra uji dengan citra acuan. Nilai *square euclidean* yang mendekati nilai nol, akan menunjuk pada citra tertentu. Nilai vektor ciri citra masukan yang memiliki nilai vektor ciri yang sama dengan vektor ciri citra tertentu akan memiliki nilai *square euclidean* yang mendekati nol. Rumus menghitung *square euclidean* sebagai berikut :

Rumus menghitung jarak euclidian

$$d_i = \sqrt{\sum_{j=1}^n (X_{ij} - P_j)^2} \dots\dots\dots(2.10)$$

Keterangan:

- d_i = Jarak sampel
 X_{ij} = Data sampel pengetahuan
 P_j = Data input variabel ke-j
 n = Jumlah sample

Berikut adalah contoh perhitungan menggunakan *Square Euclidean*:

$$\text{Hitung} = \text{SQRT} = (\text{ASM_Latih} - \text{ASM_Uji})^2 + (\text{CON_Latih} - \text{CON_Uji})^2 + (\text{COR_Latih} - \text{COR_Uji})^2 + (\text{VAR_Latih} - \text{VAR_Uji})^2 + (\text{IDM_Latih} - \text{IDM_Uji})^2 + (\text{ENT_Latih} - \text{ENT_Uji})^2$$

2.4.1 Metode K-Nearest Neighbor

Metode komputasi statistik telah diperkenalkan untuk mengekstrak informasi tutupan lahan secara digital di citra penginderaan jauh. Beberapa diantaranya mampu bekerja tanpa didasarkan pada *training area*, akan tetapi beberapa metode menuntut adanya daerah sampel atau *training area* untuk mengekstrak informasi tutupan lahan tersebut, metode seperti ini disebut *supervised classification*. Algoritma KNN terlebih dahulu mengidentifikasi daerah sampel sebelum melakukan penunjukkan kelas. Dalam proses identifikasi daerah-daerah sampel ini sebagian algoritma akan mengambil beberapa parameter statistik tertentu dan masing-masing kelas daerah sampel, seperti ini rata-rata, variasi, dan sebagainya. Algoritma-algoritma seperti ini dikenal sebagai algoritma klasifikasi parametric.

Algoritma klasifikasi parametrik biasanya disertai dengan sejumlah asumsi tertentu, yang sepenuhnya telah ditentukan sebelum proses k berlangsung. Misalnya sebagian *classier* berasumsi bahwa data (nilai spektral) homogen atau terdistribusi menurut pola tertentu. Dikarenakan algoritma klasifikasi parametrik berdasarkan atas asumsi tertentu yang dibuat sebelum proses klasifikasi, algoritma-algoritma ini dinilai memiliki keterbatasan jika kemudian asumsi yang digunakan tidak sesuai dengan keadaan atau kondisi data yang sebenarnya. Sejumlah ahli penginderaan jauh menyatakan bahwa seringkali asumsi dasar yang digunakan oleh algoritma

klasifikasi parametrik tertentu tidak sesuai dengan kondisi data yang sebenarnya.

Algoritma klasifikasi non-parametrik dinilai lebih siap untuk dihadapkan pada berbagai kondisi data. Algoritma klasifikasi non-parametrik memiliki asumsi dasar yang relative lebih sedikit dibandingkan algoritma klasifikasi parametrik, lebih jauh lagi algoritma ini tidak memerlukan parameter statistik tertentu dan daerah sampel, seperti rata-rata atau variasi.

Algoritma klasifikasi non-parametrik yang memiliki formulasi statistik paling sederhana dan paling mudah diimplementasikan adalah (*K-Nearest Neighbour*). K-NN masuk kategori *supervised algorithm* karena label data tetap dipakai dalam proses *training*, yaitu optimisasi nilai k (dan juga saat klasifikasi). Yang membedakan adalah *neural network*, misalnya, pasangan data-label dipakai untuk menentukan parameter *vector (weights)* dalam proses *training*. Setelah proses *training* selesai, *training* data itu dibuang, tidak diperlukan lagi dalam proses klasifikasi. Adapun K-NN termasuk kategori "*memory based method*", yaitu seluruhnya atau sebagian data *training* disimpan dan dipakai dalam proses klasifikasi. Ciri dan *memory based method* proses *trainingnya* berlangsung cepat, sedangkan proses klasifikasinya lambat.

Algoritma K-NN (*K-Nearest Neighbor*) sensitif terhadap keberadaan *noise & irrelevant* input, dibandingkan dengan *neural network*. Dalam proses *training neural network (perceptron)*, *weight* untuk atribut yang tidak memiliki kontribusi terhadap klasifikasi akan di-adjust agar nilainya mendekati 0, tetapi dalam K-NN, tidak ada perbedaan bobot untuk tiap atribut.

Data pembelajaran diproyeksikan ke ruang berdimensi banyak, dimana masing-masing dimensi merepresentasikan fitur dan data. Ruang ini dibagi menjadi bagian-bagian berdasarkan klasifikasi data pembelajaran. Sebuah titik pada ruang ini ditandai kelas c jika kelas c merupakan klasifikasi yang paling banyak ditemui pada k buah tetangga terdekat titik tersebut. Dekat atau jauhnya tetangga biasanya dihitung berdasarkan jarak Euclidean.

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

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

Ketepatan algoritma K-NN ini sangat dipengaruhi oleh ada atau tidaknya fitur-fitur yang tidak relevan. atau jika bobot fitur tersebut tidak setara dengan relevansinya terhadap idasifikasi. Riset terhadap algoritma ini sebagian besar membahas bagaimana memilih dan memberi bobot terhadap fitur, agar performa - klasifikasi menjadi lebih baik. Algoritma K-NN ini memiliki konsistensi yang kuat. Ketika jumlah data mendekati tak hingga, algoritma ini menjamin *error rate* yang tidak lebih dari dua kali Bayes error rate (*error rate* minimum untuk distribusi data tertentu).

Konsep dasar pada 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 *metode Co-occurrence Matrix* yaitu data baru yang akan di uji di kurangi dengan semua data pelatihan yang di sebut 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, apakah masuk ke kelas pohon kepala sawit muda atau pohon kepala sawit tua

7. Cari jumlah kelas dan tetangga yang terdekat dan tetapkan kelas tersebut sebagai kelas data yang akan dievaluasi

8. Selesai.

K-Nearest Neighbour berasumsi bahwa pixel-pixel yang saling berdekatan satu sama lain (dikenal sebagai pixel-pixel bertetangga) di dalam feature space akan tergolong ke dalam kelas yang sama. Dalam proses klasifikasinya, *K-Nearest Neighbour* akan mengukur jarak spektral (spectral distance) setiap pixel yang ada pada citra terhadap piksel-piksel yang berada di bawah daerah sample.

K-Nearest Neighbour dihadapkan pada satu masalah utama, yaitu lamanya waktu proses klasikasi. Hal ini dikarenakan *K-Nearest Neighbour* mengukur jarak *spektral* setiap *pixel* ke semua *pixel* yang ada dalam daerah sample. Waktu proses klasifikasi akan semakin lama seiring semakin banyaknya jumlah saluran (band), bertambah besarnya ukuran daerah sampel dan bertambahnya jumlah kelas yang akan diklasikasikan.

Kemungkinan metode yang dapat dilakukan untuk mengefisiensikan waktu eksekusi *K-Nearest Neighbour* adalah dengan memodifikasi algoritma pengukur jarak *spektral* pada algoritma ini. Umumnya *K-Nearest Neighbour* menggunakan *Euclidean Distance* sebagai metode pengukur jarak spektral.

Akan tetapi, metode pengukur jarak *spektral* lainnya seperti *Manhattan Distance* juga dapat digunakan dalam *K-Nearest Neighbour*. *Manhattan Distance* memiliki bentuk formulasi yang lebih sederhana dibandingkan dengan *Euclidean Distance*. Dilihat dari bentuk persamaan matematisnya, sudah dapat diprediksi bahwa algoritma pengukur jarak spektral ini akan memiliki waktu kalkulasi yang lebih cepat dibanding *Euclidean Distance*.

2.8 Penelitian terdahulu

Beberapa penelitian yang pernah dilakukan untuk mendeteksi telur *infertil* menggunakan teknik yang berbeda dijabarkan dalam tabel 2.1

Tabel 2.1 Penelitian terdahulu

NO	Judul	Pengarang	Tahun	Kelebihan
1	<i>Detecting Fertility of Hatching es using machine vision and histogram characterization method</i>	Das. K Evans .M.D.	1992	Tingkat akurasi 96 hingga 100% pada hari keempat pengeraman dan 88 hingga 90% pada hari ketiga
2	<i>The identification of white fertile eggs prior to incubation based on machine vision and least square supportvector machine</i>	Zhihui Zhu, Meihu Ma	2011	Tingkat akurasi yang dihasilkan sebesar 92.5 % dari 100 sampel yang dideteksi
	<i>Egg Embryo Development Detection with</i>	Kurt C. Lawrence, DouglasSmith, William	2008	pertama dan 92 % dihari ke Tingkat

	<i>Hyperspectral Imaging</i>	<i>R.Windham, Gerald W.Heitschmidt, Bosoon Park</i>		<i>akurasi yang dihasilkan 91.7 % sehari</i>
4	<i>Aplikasi mesin visi dalam pendeteksian fertilitas telur</i>	<i>Faridah Nopriadi, Alfa antariksa</i>	2008	<i>Tingkat akurasi pendeteksian sebesar 92.36 % dari hasil pengujian menggunakan 144 sampel</i>
5	<i>Deteksi Telur Infertil Menggunakan Jaringan Saraf Tiruan Multilayer Perceptron</i>	<i>Muhammad Zaen Nawawi</i>	2014	<i>pengenalan menggunakan data yang dilatihkan. Tingkat pengenalan yang dicapai adalah 98%.</i>