

## BAB II

### TINJAUAN PUSTAKA DAN LANDASAN TEORI

#### 2.1 Penelitian Terdahulu

Beberapa penelitian mengenai pengenalan tulisan tangan telah banyak dilakukan oleh para peneliti, berbagai metode baik ekstraksi fitur maupun metode klasifikasi telah diujicobakan. Antara lain sebagai berikut:

Penelitian sebelumnya, tentang pengenalan citra huruf alphabet tulisan tangan dilakukan oleh Morwati[1]. Dengan metode *naïve bayes classifier*, Metode ini dalam penggunaan data petunjuk untuk mendukung keputusan pengklasifikasian, khususnya dalam proses klasifikasi dokumen. Metode ROI mendukung akurasi yang lebih baik untuk pengenalan huruf alphabet.

Penelitian kedua dirancang sebuah sistem pengenalan tulisan tangan aksara Lampung dilakukan oleh Eliza Hara[2]. Dengan metode deteksi tepi (*canny*) dapat mendeteksi tepian yang sebenarnya dengan tingkat kesalahan minimum. Didesain untuk menghasilkan citra tepian yang optimal. Metode JST, *backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan. Serta kemampuan jaringan member respon yang benar terhadap pola masukan yang serupa dengan pola yang dipakai.

Penelitian yang ketiga tentang pengenalan pola angka 0 sampai 9 menggunakan jaringan syaraf tiruan (JST) dilakukan oleh Wastiunamsih[3]. Dengan model jaringan koheren, JST dengan model jaringan koheren memiliki kemampuan untuk mengelompokan pola yang mirip ke dalam satu kelompok. Pada penelitian, peneliti berhasil mengenali pola angka 0 sampai 9 dengan tingkat

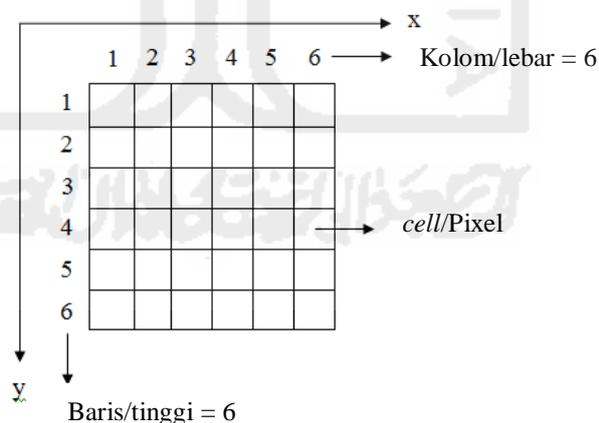
keberhasilan 100% dengan model jaringan koheren.

## 2.2 Citra

### 2.2.1 Pengertian Citra

Citra adalah suatu representasi (gambaran), kemiripan dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal yang dapat disimpan pada suatu media penyimpanan.

Citra digital merupakan citra yang disimpan dalam format digital (dalam bentuk file). Hanya citra digital yang dapat diolah menggunakan komputer. Jenis citra lain akan diolah dengan komputer harus diubah dahulu menjadi citra digital. Citra digital dibayangkan sebagai suatu matriks dimana baris dan kolomnya menggambarkan suatu titik didalam citra pada koordinat spasial  $x$  dan  $y$ . Gambar 2.1 menunjukkan posisi koordinat citra digital.



Gambar 2.1 Posisi koordinat citra digital

Citra digital berbentuk empat persegi panjang dan dimensi ukurannya dinyatakan sebagai tinggi x lebar (lebar x panjang). Citra digital yang tingginya  $N$ , lebarnya  $M$  dan memiliki  $L$  derajat leabuan dapat dianggap sebagai fungsi[4]:

$$f(x,y) \begin{cases} 0 \leq x \leq M \\ 0 \leq y \leq N \\ 0 \leq f \leq L \end{cases} \quad (2.1)$$

Dimana:

$N$  adalah jumlah baris (panjang atau tinggi matriks)

$M$  adalah jumlah kolom (lebar matriks)

$L$  adalah intensitas warna maksimal (derajat keabuan)

Citra digital yang berukuran  $N \times M$  dinyatakan dengan matriks yang berukuran  $N$  baris dan  $M$  kolom, dengan  $x$  dan  $y$  adalah koordinat  $(x,y)$ . Citra digital dapat ditulis dalam bentuk matriks sebagai berikut.

$$f(x, y) \approx \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M) \\ f(1,0) & f(1,1) & \dots & f(1,M) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (2.2)$$

### 2.2.2 Pengolahan Citra

Pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara. Pengolahan citra merupakan bagian penting yang mendasari berbagai aplikasi nyata, seperti dalam bidang biometrika, medis, geografi dan banyak bidang lainnya.

Dalam bentuk sederhana pengolahan citra bertujuan untuk melakukan perbaikan suatu citra menjadi citra yang berkualitas, seperti terlihat pada gambar 2.2, dimana pada gambar (a) memperlihatkan citra burung nuri tampak agak gelap. Setelah dilakukan perbaikan kontras, citra burung nuri lebih terang dan tajam diperlihatkan pada gambar (b).



Gambar 2.2 Proses pengolahan citra [4]  
 (a) citra asli tampak gelap, (b) citra setelah diperbaiki kontras lebih tajam dan terang

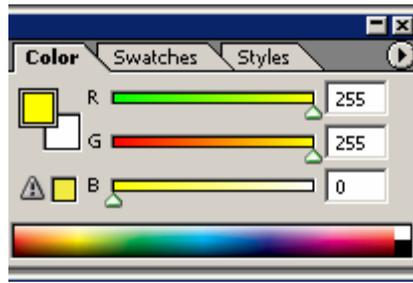
## 2.3 Pre-Proses Pengolahan Citra Digital

### 2.3.1 Binerisasi Citra

Dalam pengolahan citra digital, binerisasi citra melalui beberapa proses sebagai berikut:

#### 2.3.1.1 Citra RGB (*Red-Green-Blue*)

Setiap titik (pixel) pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar yaitu merah hijau biru → citra RGB (Red-Green-Blue). Banyaknya warna yang mungkin digunakan tergantung pada kedalaman pixel yang digunakan. Setiap komponen warna memiliki kedalaman pixel sendiri dengan nilai minimum 0 dan nilai maksimum 255 (8 bit). Hal ini menyebabkan setiap pixel pada citra RGB membutuhkan ruang penyimpanan 3 byte (1 byte = 8 bit). Berikut ini contoh gambar RGB pada photoshop.



Gambar 2.3 Contoh RGB pada photoshop

### 2.3.1.2 Citra Grayscale

Citra grayscale merupakan citra digital yang memiliki satu nilai pixel setiap pixelnya. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas warna yang dimiliki adalah dari hitam, keabuan dan putih. Tingkat keabuan disini ialah warna abu-abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Citra grayscale berikut memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan)[5].

Citra grayscale disimpan dalam format 8 bit untuk setiap sampel pixel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing R, G dan B menjadi citra grayscale dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B sehingga dapat dituliskan menjadi:

$$X = (R + G + B) / 3 \quad (2.3)$$

### 2.3.1.3 Citra biner

Citra biner adalah dengan setiap piksel hanya dinyatakan dengan sebuah nilai dari dua buah kemungkinan (yaitu nilai 0 dan 1). Nilai 0 menyatakan

warna hitam dan nilai 1 menyatakan warna putih. Citra jenis ini banyak dipakai dalam segmentasi citra, contohnya untuk mendeteksi tepi citra. Binerisasi citra adalah proses merubah citra kedalam bentuk biner (0 dan 1). Proses ini menyebabkan citra hanya memilih dua warna yaitu hitam dan putih.

## 2.3.2 Filtering Pada Citra

### 2.3.2.1 Mean Filtering

Ada beberapa jenis *mean filter* yang dapat digunakan untuk menghilangkan derau, berikut akan dijelaskan secara lebih rinci.

#### a. Arithmetic Mean Filter

Pada prinsipnya *arithmetic mean filter* sama dengan *mean filter*. *Arithmetic mean filter* bekerja dengan cara menggantikan nilai piksel sebuah citra yang terkena derau dengan nilai rata-rata nilai piksel tetangga dan dirinya sendiri yang berukuran  $m \times n$  [4]. Formula yang dapat digunakan adalah sebagai berikut:

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t) \quad (2.4)$$

Dimana:

$\hat{f}(x, y)$  = hasil *arithmetic mean filter*

$mn$  = jumlah baris ke  $m$  dan kolom  $n$

$g(s, t)$  = nilai citra pada kordinat *sub-image*

$S_{xy}$  = daerah yang diliputi oleh *filter*

$(s, t)$  = intensitas nilai pixel

Berikut ini contoh operasi *arithmetic mean filter*:

4	7	8
2	3	4
1	8	9

→

4	7	8
2	<b>5</b>	4
1	8	9

Gambar 2.4 Proses *arithmetic mean filter*

$$\hat{f}(x, y) = \frac{1}{9}(4 + 7 + 8 + 2 + 3 + 4 + 1 + 8 + 9) \approx 5,1 = 5$$

Filter ini cocok untuk menghilangkan derau Uniform dan Gaussian, tetapi akan sedikit mengaburkan citra.

#### **b. Harmonic Mean Filter**

Operasi pemfilteran dengan *harmonic mean filter* ini bekerja dengan baik untuk *salt noise*, tetapi gagal untuk *pepper noise*. Dapat bekerja dengan baik dengan tipe *noise* yang lain seperti *gaussian noise*, berikut ini rumus *harmonic mean filter*:

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}} \quad (2.5)$$

Dengan :

$\hat{f}(x, y)$  = hasil *harmonic mean filter*

$mn$  = jumlah baris ke  $m$  dan kolom  $n$

$(s, t)$  = intensitas nilai pixel

$g(s,t)$  = nilai citra pada kordinat *sub-image*

Contoh berikut operasi pada *harmonic mean filter*:

4	7	8	⇒	4	7	8
2	3	4		2	<b>3</b>	4
1	8	9		1	8	9

Gambar 2.5 Proses *harmonic mean filter*

$$\hat{f}(x,y) = \frac{3 \times 3}{\frac{1}{4} + \frac{1}{7} + \frac{1}{8} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{1} + \frac{1}{8} + \frac{1}{9}} \approx 3,17 = 3$$

Contoh berikut merupakan penerapan *harmonic mean filter* pada citra yang telah terdegradasi *gaussian noise*.

### c. *Contraharmonic Mean Filter*

*Contraharmonic mean filter* sangat cocok untuk mengurangi atau mengeliminasi efek *noise Salt and Pepper*. Filter ini dirumuskan dengan:

$$\hat{f}(x,y) = \frac{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s,t)^Q} \quad (2.6)$$

Dimana:

$$\hat{f}(x,y) = \text{hasil } \textit{contraharmonic mean filter}$$

$(s,t)$  = intensitas nilai pixel

$S_{xy}$  = region citra

$g(s,t)$  = nilai citra pada kordinat *sub-image*

$Q$  = order filter

Berikut ini contoh matriks citra 3x3 *contraharmonic mean filter*:

5	5	4		5	5	4
7	2	6	⇒	7	5	6
1	4	1		1	4	1

Gambar 2.6 Proses *contraharmonic mean filter*

Bila  $Q = 1$ , maka:

$$\hat{f}(x,y) = \frac{5^2 + 5^2 + 5^2 + 4^2 + 7^2 + 2^2 + 6^2 + 1^2 + 4^2 + 1^2}{5 + 5 + 4 + 7 + 2 + 6 + 1 + 4 + 1} \approx 4,9 = 5$$

Filter ini tidak bisa menghilangkan derau *salt and pepper* secara bersamaan. *Salt* hanya dapat dihilangkan bila  $Q$  bernilai negatif, sedangkan bila  $Q$  bernilai positif maka akan bekerja untuk derau *pepper*.  $Q$  disebut juga dengan orde filter. Misalkan nilai  $Q = 0$ , maka bisa dibuktikan bahwa rumusnya akan berubah menjadi rumus *arithmetic filter*. Sedangkan bila  $Q = -1$  filter akan berubah menjadi *contraharmonic filter*.

### 2.3.2.2 Median Filtering

Prinsip kerja *median filter* melakukan proses rata-rata untuk mencari nilai tengah, dari titik penjumlahan dari semua tetangga. Filter ini menggunakan nilai median piksel-piksel di dalam jendela sebagai keluaran  $\hat{f}$ . Formula yang digunakan:

$$\hat{f}(x, y) = \text{median}_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (2.7)$$

Keterangan:

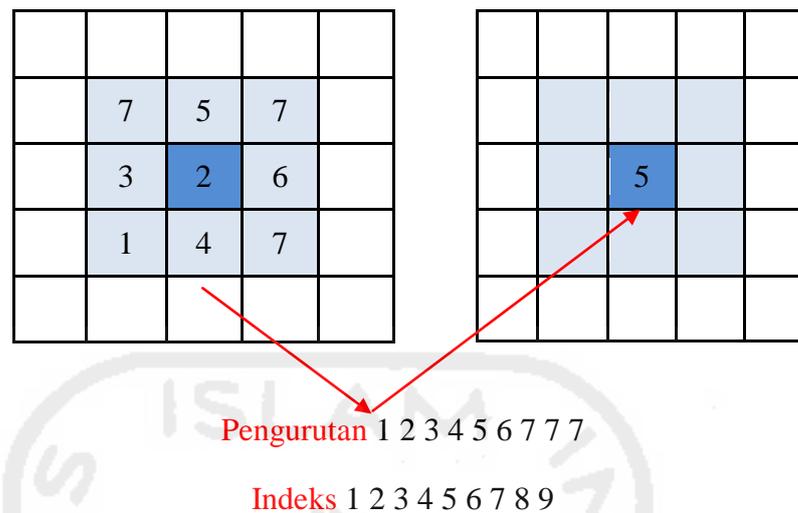
$\hat{f}(x, y)$  = hasil *median filter*

$(s, t)$  = intensitas nilai pixel

$S_{xy}$  = region citra

$g(s, t)$  = nilai citra pada kordinat *sub-image*

*Filter median* sangat terkenal karena untuk jenis *random noise* tertentu memberikan kemampuan pengurangan *noise* yang sangat baik. Berikut ini contoh gambar 2.7 pemilihan *kernel 3x3* dan mengurutkan nilai pixel.



Gambar 2.7 Pemilihan *kernel* 3x3 dan mengurutkan nilai pixel

### 2.3.2.3 Gaussian Filtering

Filter Gaussian adalah filter yang dipilih berdasarkan bentuk fungsi *gaussian*. Filter ini sangat baik untuk menghilangkan *noise* yang bersifat sebaran normal, yang banyak dijumpai pada citra hasil proses digitasi menggunakan kamera karena merupakan fenomena alamiah akibat sifat pantulan cahaya dan kepekaan sensor cahaya pada kamera itu sendiri.

Dari semua jenis filter diatas penelitian ini menggunakan *filter median* sebagai metode *filtering* pada citra. Pemilihan filter rata-rata dikarenakan filter ini mampu memperbaiki detail-detail kecil citra yang rusak akibat terkena derau (*noise*)

### 2.3.3 Resizing Pada Citra

Resizing pada citra adalah pengolahan citra dengan kegiatan memotong satu bagian dari citra. Rumus yang gunakan:

$$W = x_1 - x_2 \text{ dan } H = y_1 - y_2 \quad (2.8)$$

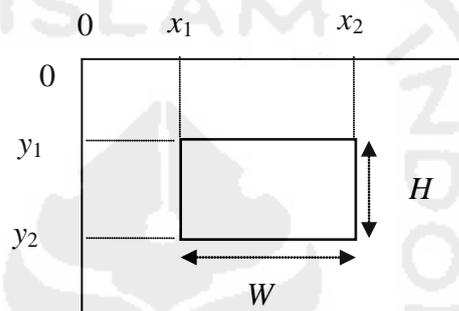
Dimana :

$x_1, y_1$  adalah koordinat titik pojok kiri atas

$x_2, y_2$  adalah koordinat titik pojok kanan atas

$W$  adalah hasil pemotongan  $x_1 - x_2$

$H$  adalah hasil pemotongan  $y_1 - y_2$



Gambar 2.8 Proses resizing

### 2.3.4 Normalisasi Pada Citra

Pada proses normalisasi gambar akan berputar dengan derajat sudut dalam arah berlawanan di sekitar titik pusatnya. Citra digital memiliki  $L$  derajat keabuan, nilai 0 sampai 1 (misalnya pada citra dengan kuantisasi derajat keabuan 8 bit, nilai derajat keabuan 0 sampai 255)[5]. Berikut ini rumus untuk menghitung normalisasi:

$$h_i = \frac{n_i}{n} \quad (2.9)$$

Dimana:

$h_i$  = hasil normalisasi

$i = 0, 1, \dots, L-1$

$n_i$  = jumlah pixel yang dimiliki derajat keabuan  $i$

$n$  = jumlah seluruh citra di dalam pixel

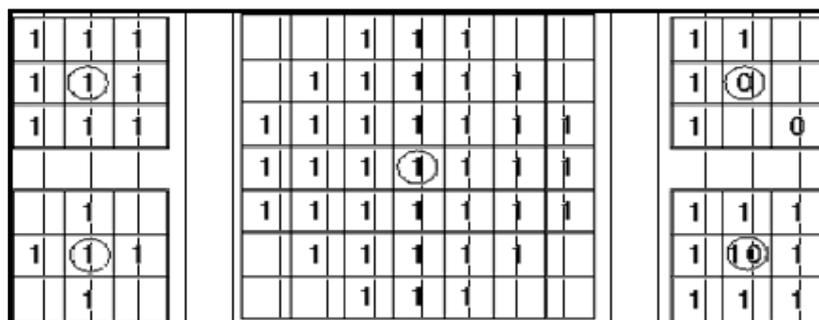
### 2.3.5 Morfologi Pada Citra

Kata morfologi (*morphology*) secara sederhana dapat diartikan sebagai bentuk dan struktur suatu objek. Morfologi adalah untuk memperbaiki bentuk objek agar dapat menghasilkan fitur-fitur yang lebih akurat ketika analisis dilakukan terhadap objek. Jadi operasi morfologi bukanlah operasi yang mempunyai tujuan analisis, melainkan suatu operasi yang bertujuan untuk memfasilitasi analisis yang dilakukan terhadap objek tersebut.

Ada beberapa komponen penting yang perlu diketahui pada pengolahan citra, seperti struktur elemen (*kernel*) dan dilasi, yaitu:

#### 1) Struktur elemen

Struktur elemen atau biasa disebut *kernel*, berisi pola yang menghususkan koordinat dari beberapa titik yang memiliki relatifitas yang sama ke suatu pusat (*origin*). Biasanya direpresentasikan menggunakan koordinat *kartesian* untuk setiap elemen sebagai kotak-kotak kecil. Contoh berbagai ukuran dari beberapa struktur elemen dapat dilihat pada Gambar 2.9.



Gambar 2.9 Contoh ukuran dari beberapa struktur elemen [5]

## 2) Dilasi

Operasi dilasi dilakukan untuk memperbesar ukuran segmen objek dengan menambahkan lapisan di sekeliling objek. Terdapat 2 cara untuk melakukan operasi ini, yaitu dengan cara mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik objek atau lebih mudahnya set setiap titik yang tetangganya adalah titik objek menjadi titik objek. Cara kedua yaitu dengan mengubah semua titik di sekeliling titik batas menjadi titik objek atau lebih mudahnya set semua titik tetangga sebuah titik objek menjadi titik objek.

Suatu objek (citra input) dinyatakan dengan  $A$  dan SE dinyatakan dengan  $B$  serta  $B_x$  menyatakan translasi  $B$  sedemikian hingga pusat  $B$  terletak pada  $x$ . Operasi dilasi  $A$  dengan  $B$  dapat dinyatakan sebagai berikut.[6]

$$A \oplus B = \{x \mid B_x \cap A \neq \emptyset\} \quad (2.10)$$

Dengan:

$A$  menyatakan citra biner yang dikenakan operasi pada objeknya

$B$  menyatakan citra yang melakukan operasi perubahan bentuk objeknya

$\oplus$  menyatakan citra bilangan XOR

$x$  menyatakan hasil operasi dilasi  $A$  oleh  $B$

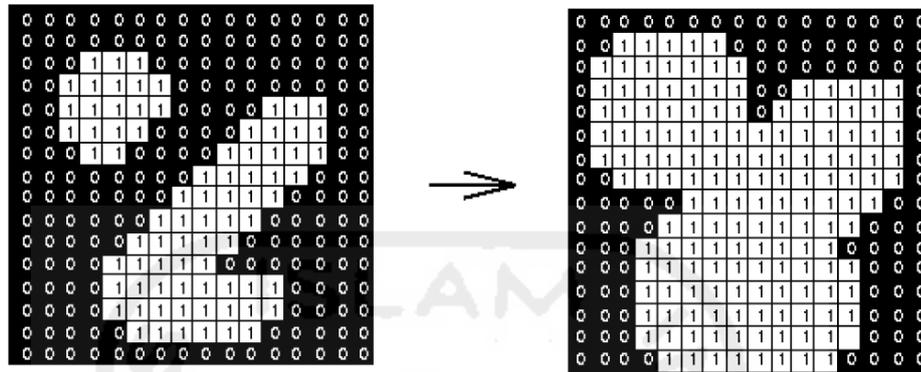
$B_x$  menyatakan translasi  $B$  terletak pada  $x$

$\cap$  menyatakan irisan dari himpunan  $A$  dan  $B$

$\emptyset$  menyatakan himpunan kosong

Proses dilasi dilakukan dengan membandingkan setiap pixel citra input dengan nilai pusat SE dengan cara melapiskan SE dengan citra sehingga pusat SE tepat dengan posisi pixel citra yang diproses.

Gambar 2.10 berikut ini contoh proses dilasi dengan menggunakan SE berukuran 3 x 3 dengan elemen SE bernilai 1.



Gambar 2.10 Proses dilasi menggunakan SE bernilai 1 [6]

## 2.4 Pengenalan Pola

Pola adalah entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama. Pola bisa merupakan kumpulan hasil pengukuran atau pemantauan dan bisa dinyatakan vektor atau matriks. Secara umum pengenalan pola (*pattern recognition*) adalah suatu ilmu untuk mengklasifikasikan atau menggambarkan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu objek [7].

Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang baik adalah ciri yang memiliki daya pembeda yang tinggi sehingga pengenalan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan baik. Pengenalan pola ini bersifat *conceptually driven processing* yang berarti bahwa proses dimulai dari pembentukan konsep pada objek yang dijumpai. Ada banyak aplikasi yang dijadikan implementasi dari pengenalan pola, diantaranya pengenalan wajah manusia, pengenalan gambar mata, pengenalan penyakit

berdasarkan gejala-gejala yang ditemukan pada objek dan lain-lain. Salah satu metode ekstraksi fitur pada pengenalan pola yang banyak digunakan adalah metode zernike moment.

## 2.5 Metode Zernike Moment

Moment *Zernike* diperkenalkan oleh F. Zernike dalam bukunya berjudul *Phisica* yang diterbitkan pada tahun 1934. Penerapan moment *Zernike* untuk pengolahan citra diperkenalkan pertama kali oleh M.R Teague pada tahun 1980. Bila dilihat dari sisi perhitungan, *Zernike* moment melibatkan perhitungan yang lebih kompleks. Namun *Zernike* moment telah dibuktikan sebagai salah satu metode ekstraksi fitur atau ciri yang mengalami rotasi.

*Zernike* moment termasuk pada *region-based shape descriptor*. Jenis moment ini dikenal sangat efisien pada penggunaannya untuk pengenalan pola sebab sifat ortogonalitas pada *Zernike polynomials* dalam hasil ekstraksi ciri yang dibentuk serta memiliki properti yang tidak tergantung pada rotasi citra. Moment *Zernike* didasarkan pada *polynomial Zernike* yang bersifat orthogonal terhadap lingkaran  $x^2 + y^2 \leq 1$ , yang dinyatakan sebagai berikut:

$$V_{pq}(x, y) = U_{pq}(r \cos\theta, r \sin\theta) = R_{pq}(r) \cdot \exp(jq\theta) \quad (2.11)$$

Dengan:

$V_{pq}(x,y)$  adalah titik koordinat

$U_{pq}$  adalah sudut antara  $\cos\theta$  dan  $\sin\theta$

$R_{pq}$  adalah *polynomial* radial orthogonal

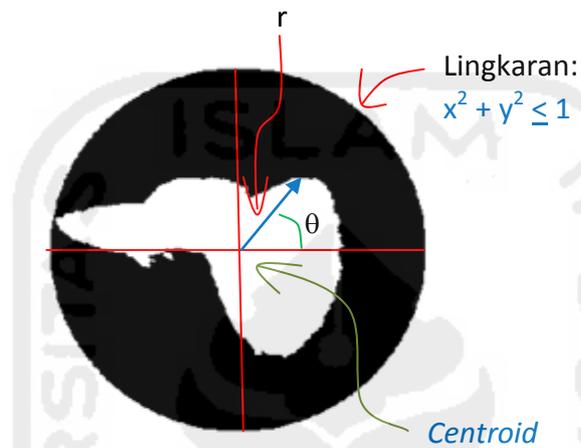
$r$  adalah radius dari (y,x) ke pusat massa (centroid)

$\theta$  adalah sudut antara r dan sumbu x

$j$  adalah bilangan ganjil mulai dari satu

$q$  adalah bilangan bulat mulai dari nol

pada yang dilihat pada gambar 2.11 dimana citra dalam lingkaran yang memenuhi persamaan  $x^2 + y^2 \leq 1$ .



Gambar 2.11 Citra dalam lingkaran yang memenuhi persamaan  $x^2 + y^2 \leq 1$  [7]

Pada momen *Zernike* terdapat moment spasial orde  $(m,n)$ , didefinisikan sebagai berikut:

$$M_{ij} = \sum_{x=1}^M \sum_{y=1}^N x^i y^j I(x, y)$$

(2.12)

Dimana:

$i, j = 0, 1, 2, \dots$ , dengan  $i, j$  menyatakan orde moment

$M$  = jumlah kolom dalam citra

$N$  = jumlah baris pada citra

$x$  = ordinat pixel

$y$  = absis pixel

$I(x,y)$  = intensitas pixel pada posisi  $(x,y)$

## 2.6 Klasifikasi Jaringan Syaraf Tiruan (JST)

Jaringan syaraf tiruan (*artificial neural network*) adalah sistem pemroses informasi dengan karakteristik dan performa yang mendekati syaraf biologis, sama seperti otak yang memproses suatu informasi. Jaringan syaraf tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klarifikasi karena proses pembelajaran.

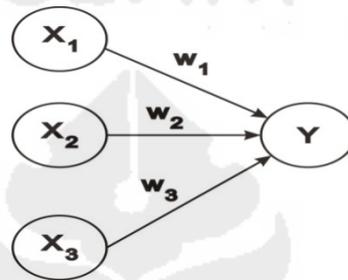
Jaringan syaraf tiruan merupakan sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi. Jaringan syaraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa:[8]

1. Pemroses informasi terjadi pada banyak elemen sederhana (neuron).
2. Sinyal dikirim melalui neuron-neuron melalui penghubung-penghubung.
3. Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal.
4. Penentuan output, setiap neuron menggunakan fungsi aktivasi dikenalkan pada jumlah output yang diterima. Kemudian besar output tersebut dibandingkan batas ambang.

Jaringan syaraf ditentukan oleh:

1. Pola hubungan antar neuron (arsitektur jaringan).
2. Metode penentuan bobot penghubung (metode *training/learning/algorithm*).
3. Fungsi aktivasi.

Sebagai contoh seperti gambar 2.12 perhatikan neuron Y:



Gambar 2.12 Neuron [8]

Y menerima input dari neuron  $x_1$ ,  $x_2$ ,  $x_3$  dengan bobot masing-masing  $w_1$ ,  $w_2$ ,  $w_3$ .

Ketika inplus neuron yang ada dijumlahkan :

$$net = x_1w_1 + x_2w_2 + x_3w_3 \quad (2.13)$$