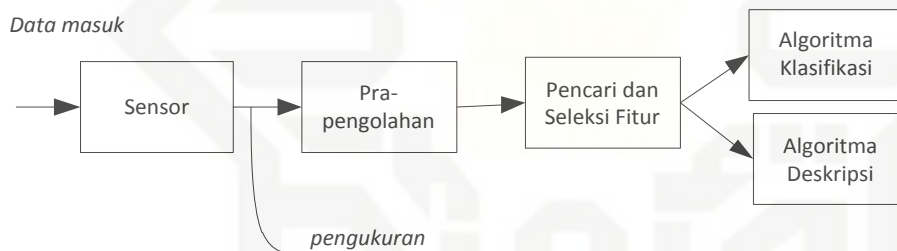


BAB II LANDASAN TEORI

2.1. Pengenalan Pola

Pengertian dari pengenalan pola (*pattern recognition*) adalah suatu bidang ilmu untuk menggambarkan atau mengklasifikasikan sesuatu berdasarkan pengukuran kuantitatif fitur (ciri) atau sifat utama dari suatu objek. Sedangkan untuk pengertian dari pola adalah suatu entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama. Contoh dari pola adalah sidik jari, motif batik, dan lain sebagainya. Pola dapat juga berupa kumpulan hasil pengukuran atau pemantauan dan bisa dinyatakan dalam notasi vektor atau matriks. Struktur dari sistem pengenalan pola dapat dilihat pada gambar 2.1 dibawah ini (Putra, 2010):



Gambar 2. 1 Struktur Sistem Pengenalan Pola

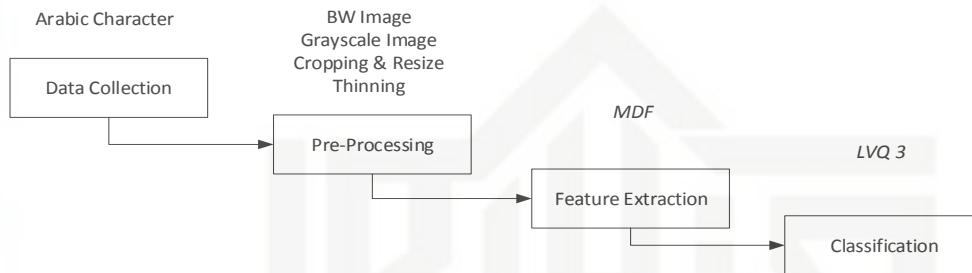
Keterangan gambar 2.1 sebagai berikut:

1. Sensor berfungsi sebagai penangkap objek dari dunia nyata yang selanjutnya akan diubah menjadi sinyal digital melalui proses digitalisasi.
2. Pra-pengolahan berfungsi untuk mempersiapkan citra atau sinyal agar dapat menghasilkan ciri yang lebih baik untuk tahap selanjutnya. Pada tahap ini sinyal dari informasi ditonjolkan dan sinyal pengganggu (*derau*) diminimalisasi.
3. Pencari dan seleksi fitur sebagai penemu dari karakteristik pembeda yang akan mewakili sifat utama sinyal dan sekaligus mengurangi dimensi sinyal menjadi sekumpulan bilangan yang lebih sedikit tetapi representatif.
4. Algoritma klasifikasi berfungsi untuk mengelompokkan fitur ke dalam kelas yang sesuai.

5. Algoritma deskripsi berfungsi untuk memberikan deskripsi pada sinyal.

2.2. Pengenalan Pola Karakter

Pada penelitian ini, proses pengenalan pola karakter yang akan dilakukan dapat dilihat pada gambar 2.2 berikut (Shatil, 2006):



Gambar 2. 2 Proses Pengenalan Pola Karakter Huruf Hijaiyah

2.2.1. Data Collection

Tahap ini merupakan tahap pertama dalam pengenalan pola karakter. *Data collection* digunakan sebagai data masukan dalam pengenalan pola. Pola masukan berupa sebuah citra, dimana pola dari citra tersebut adalah pola huruf Hijaiyah.

2.2.2. Pre-processing

Tahap kedua setelah *data collection* adalah *pre-processing*. Tahap inilah dilakukan pengolahan citra (*image processing*). Hal yang dilakukan dalam proses ini adalah konversi *RGB* ke dalam tipe *grayscale*, lalu dikonversi kembali kedalam tipe biner. Selanjutnya dilakukan proses *cropping* dan *resize* untuk mendapatkan ukuran citra yang diinginkan lalu lakukan proses *thinning*, dimana dalam proses ini piksel pada citra bertipe biner akan dikikis menjadi piksel yang tipis.

2.2.3. Feature Extraction

Tahap ini dilakukan setelah tahap *pre-processing* selesai. Tahap *feature extraction* atau ekstraksi ciri dilakukan untuk mencari sebuah vektor dalam sebuah citra. Tahap ini dilakukan dengan menggunakan metode ekstraksi ciri *Modified Direction Feature* (MDF)

2.2.4. Classification

Tahap ini merupakan tahap akhir dalam pengolahan citra yaitu *classification*. Tahap ini dilakukan proses klasifikasi masukan yang telah melalui

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

tahapan sebelumnya sehingga menghasilkan vektor untuk digunakan sebagai data uji pada metode LVQ 3.

2.3. Huruf Hijaiyah

Bahasa Arab (*al-lughah al-'Arabiyyah*) atau secara mudahnya (*'Arrabi*) adalah sebuah bahasa semitik yang muncul di daerah yang sekarang termasuk wilayah Arab Saudi. Walaupun bahasa ini dianggap sebagai bahasa asing oleh bangsa Indonesia, bahasa Arab tidak asing ditelinga mereka, terutama umat Islam.

Sebagaimana diketahui bahwa Islam merupakan agama wahyu yang diturunkan oleh Allah Swt kepada Nabi Muhammad Saw dengan perantara Malaikat Jibril As. Wahyu yang diturunkan terakhir kepada Nabi Muhammad Saw sebagai utusan terakhir, dihimpun menjadi kitab suci Al-Qur'an yang berbahasa Arab sebagaimana yang disebutkan dalam Surat Yusuf (12) ayat 2 dan Surat Al-Syu'ara (26) ayat 192-196. Begitu pula dengan Hadits yang merupakan penjelasan dan penafsiran Al-Qur'an, dihimpun dan disusun dalam bahasa Arab "Cintailah Bahasa Arab karena tiga hal: aku adalah orang Arab; Al-Qur'an berbahasa Arab; dan bahas ahli surga adalah Arab" mengutip dari Hadits Muslim dari Ibnu Abbas. Dengan demikian, dapat disimpulkan bahwa sumber pokok agama Islam yaitu AL-Qur'an dan Hadits keduanya berbahasa Arab (Hermawan, 2013).

Penulisan dalam bahasa Arab menggunakan huruf Arab atau yang sering dikenal dengan huruf Hijaiyah. Huruf Hijaiyah ini terdiri dari 28 huruf, dimana cara penulisannya adalah dari kanan ke kiri. Beberapa karakter huruf hijaiyah dapat dilihat pada tabel 2.1 berikut (Anggraini, 2015):

Tabel 2. 1 Karakter Huruf Hijaiyah

No	Tunggal	Awal	Tengah	Akhir
1	ا	-	-	ـا
2	ب	بـ	بـ	ـب
3	ت	تـ	تـ	ـت
4	ث	ثـ	ثـ	ـث

5	ج	ح	خ	ج
6	د	ذ	د	د
7	ر	ز	ز	ر
8	س	.	.	س
9	ش	.	.	ش
10	ص	.	.	ص
11	ض	.	.	ض
12	ط	ط	ط	ط
13	ظ	ظ	ظ	ظ
14	ق	ق	ق	ق
15	ك	ك	ك	ك
16	گ	گ	گ	گ
17	خ	خ	خ	خ
18	د	د	د	د
19	ذ	ذ	ذ	ذ
20	ر	ر	ر	ر
21	ز	ز	ز	ز

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

22	ق	ق	ق	ق
23	ك	ك	ك	ك
24	ه	ه	ه	ه
25	م	م	م	م
26	ن	ن	ن	ن
27	و	-	-	و
28	ي	ي	ي	ي
Total Huruf	28	22	22	28

2.4. Citra Digital

Menurut Putra pada tahun 2010 citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real ataupun kompleks yang direpresentasikan dengan deretan bit tertentu. Sedangkan menurut Mauridhi dan Arif pada tahun 2010 citra digital merupakan susunan dari sejumlah nilai tingkat keabuan yang dikenal dengan piksel (*pixel*) pada posisi tertentu.

Citra digital dapat dinyatakan sebagai duatu fungsi dua dimensi $f(x,y)$, dengan x ataupun y adalah posisi koordinat sedangkan f adalah amplitudo pada posisi (x,y) yang dikenal sebagai intensitas atau *grayscale*. Bentuk nilai dari intensitas adalah diskrit mulai dari 0 – 255 (Purnomo & Muntasa, 2010).

2.4.1. Citra Berwarna (RGB)

Citra berwarna atau dikenal dengan citra RGB adalah jenis citra yang menyajikan warna dalam bentuk komponen R (*red* atau merah), G (*green* atau

hijau), dan B (*blue* atau biru). Masing-masing komponen dari warna tersebut menggunakan 8 bit, yang bernilai berkisar antara 0 – 255 dari rentang nilai komponen warna tersebut, dapat dikatakan bahwa kemungkinan warna yang bisa disajikan mencapai $256 \times 256 \times 256$ atau 16.777.216 warna (Purnomo & Muntasa, 2010).

2.4.2. Citra *Grayscale*

Citra *grayscale* menangani gradasi warna hitam dan putih yang pastinya akan menghasilkan efek warna abu-abu. Pada citra ini warna dinyatakan dengan intensitas. Intensitas citra berkisar 0 – 255 dengan jumlah bitnya adalah 8. Nilai 0 menyatakan hitam dan 255 menyatakan putih. Nilai intensitas dari citra *grayscale* tidak akan melebihi 255 ataupun tidak mungkin kurang dari 0 (Purnomo & Muntasa, 2010).

2.4.3. Citra Biner

Citra biner merupakan citra digital masing-masing nilainya memiliki dua kemungkinan yaitu hitam dan putih, sehingga citra ini dikenal sebagai citra B&W (*black and white*) atau citra monokrom. Nilai intensitas dari citra ini berkisar 0 dan 1, dimana nilai 0 menyatakan hitam dan nilai 1 menyatakan putih (Putra, 2010).

2.5. Pengolahan Citra Digital

Meskipun sebuah citra kaya akan informasi, sebuah citra kerap kali mengalami penurunan (*degradasi*), seperti adanya cacat (*noise*) pada citra, warnanya terlalu kontras, kurang tajam, kabur (*blur*), dan sebagainya. Permasalahan tersebut tentu saja mengakibatkan sulitnya citra tersebut diinterpretasikan karena berkurangnya informasi yang disampaikan oleh citra tersebut.

Solusi yang perlu dilakukan pada citra tersebut adalah dengan melakukan manipulasi citra tersebut menjadi citra lain yang memiliki kualitas yang lebih baik dari sebelumnya. Adapun ilmu yang membahas tentang hal ini adalah pengolahan citra (*image processing*).

Pengolahan citra adalah suatu proses pemrosesan citra dimana data yang dimasukkan berupa sebuah citra (Munir, 2004).

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.5.1. Konversi RGB ke Grayscale

Untuk mengubah suatu citra menjadi citra biner, terlebih dahulu citra tersebut harus dikonversi kedalam citra *grayscale*. Salah satu persamaan yang digunakan untuk konversi RGB ke *Grayscale* adalah (Kadir & Susanto, 2013):

$$Grayscale = (0.2989 * R) + (0.5870 * G) + (0.1141 * B) \dots\dots\dots(2. 1)$$

Keterangan:

R = merepresentasikan nilai warna merah pada citra

G = merepresentasikan nilai warna hijau pada citra

B = merepresentasikan nilai warna biru pada citra

2.5.2. Konversi Grayscale ke Biner

Sebelum mengubah citra *grayscale* menjadi citra biner, terlebih dahulu carilah nilai dari *threshold* dari histogram nilai *grayscale*. Persamaan yang digunakan pada proses *thresholding* citra *grayscale* sebagai berikut (Putra, 2010):

$$g(x, y) = \begin{cases} 1, & \text{jika } f(x, y) \geq T \\ 0, & \text{jika } f(x, y) < T \end{cases} \dots\dots\dots(2. 2)$$

Keterangan:

g(x,y) = citra biner dari citra *grayscale* f(x,y)

f(x,y) = citra *grayscale*


T = nilai ambang (*threshold*)

Dalam keterangan diatas bahwa f(x,y) merupakan merupakan citra *grayscale*, g(x,y) merupakan citra biner dari citra *grayscale* f(x,y), dan T merupakan nilai ambang (*threshold*). Nilai T memegang peranan penting dalam proses *thresholding*. Kualitas yang dihasilkan oleh citra biner sangat bergantung pada nilai T yang digunakan. Nilai T ini didapat dari nilai citra yang berkisar antara 0 – 255. Pada proses pengenalan pola ini, nilai *thresholding* digunakan untuk mendapatkan citra biner yang bernilai 0 dan 1.

2.5.3. Cropping & Resize


Cropping merupakan suatu proses pemotongan dari bagian citra untuk meningkatkan konsentrasi citra yang diinginkan. Proses ini dilakukan guna mendapatkan area yang hanya berfokus pada objek citra agar dapat diproses dengan baik pada tahap sebelumnya.

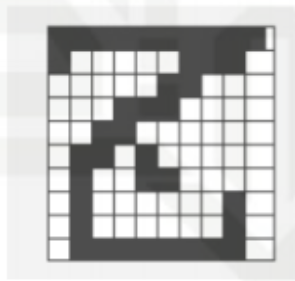
- Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
 2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Resize berguna untuk mengubah ukuran besarnya citra dalam piksel. Proses ini dilakukan agar citra memiliki ukuran piksel yang sama. Contoh dari tahapan ini adalah pada karakter  (jim) yang dilakukan oleh Andrian pada tahun 2012. Proses *cropping* yang dilakukan adalah citra inputan asli yang berukuran matriks citra 100x100 piksel akan dipotong sesuai dengan ukuran citra yang diinginkan kemudian di *resize* menjadi matriks citra berukuran 10x10 piksel.





Gambar 2. 3 Karakter  (jim) dengan Ukuran Matriks Citra 100x100 piksel

Pola karakter  (jim) kemudian dipetakan ke dalam ukuran matriks citra 10x10 piksel. Dapat dilihat pada gambar 2.4 dibawah ini, matriks yang diberi tanda hitam memiliki nilai biner 1 dan yang berwarna putih akan diberi nilai biner 0.



Gambar 2. 4 Karakter  (jim) Ukuran Matriks Citra 10x10 piksel

Dari tabel diatas, dapat dilihat nilai biner 1 merupakan hasil konversi biner dari pola karakter  (jim) dengan bobot matriks [i,j] 10x10. Tampilan bilangan biner dari 0-1 pada pola karakter  (jim) dapat dilihat pada gambar 2.5 berikut:


```
1111111111
1000001110
0000011000
0001110000
0011000000
0100100000
0100110000
0100000010
0100000010
0111111110
```

Gambar 2. 5 Representasi Biner Pola Karakter ج (jim)

Representasi bilangan biner pada pola karakter ج (jim) dalam bentuk vektor dapat dilihat pada gambar dibawah ini:

e.1111111111100000111000000110000001100000011000000010010000001001100000100000010010000001001111111110

Gambar 2. 6 Representasi Bilangan Biner dalam Vektor

2.5.4. Thinning (Penipisan)

Thinning berguna untuk mengubah bentuk asli citra biner menjadi citra yang menampilkan batas-batas objek/*foreground* hanya setebal satu piksel. Piksel pada objek biner akan direduksi menjadi piksel yang bernilai sama dengan nilai piksel *background*. Proses *thinning* secara iteratif akan menghapus piksel-piksel objek biner yang memiliki nilai transisi dari 0 ke 1. Keluaran berupa citra biner dengan ketebalan satu piksel.(Putra, 2010).

Proses pada *thinning* menggunakan *contour point* dari citra. *Contour point* merupakan piksel yang memiliki nilai 1 dan memiliki paling sedikit satu 8-tetangga yang bernilai 0.

Gambar 2.7 menunjukkan piksel 8-tetangga untuk proses *thinning* metode Zhang Shuen :

P9	P2	P3
P8	P1	P4
P7	P6	P5

Gambar 2. 7 Piksel 8-tetangga

Langkah-langkah dari proses *thinning* sebagai berikut:

1. Beri tanda *contour point* (P1) untuk dihapus apabila telah memenuhi semua kondisi berikut:

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

- (1) $2 \leq N(P1) \leq 6$
- (2) $S(P1) = 1$
- (3) $P2 \cdot P4 \cdot P6 = 0$
- (4) $P4 \cdot P6 \cdot P8 = 0$

$N(P1)$ merupakan jumlah tetangga dari $P1$ yang bernilai 1, sedangkan $S(P2)$ merupakan jumlah transisi dari 0 ke 1 dalam urutan $P2$ ke $P3$, $P3$ ke $P4$, dan seterusnya.

2. Memiliki langkah yang sama dengan langkah 1, namun untuk kondisi (3) dan (4) berubah menjadi:

- (1) $2 \leq N(P1) \leq 6$
- (2) $S(P1) = 1$
- (3) $P2 \cdot P4 \cdot P8 = 0$
- (4) $P2 \cdot P6 \cdot P8 = 0$

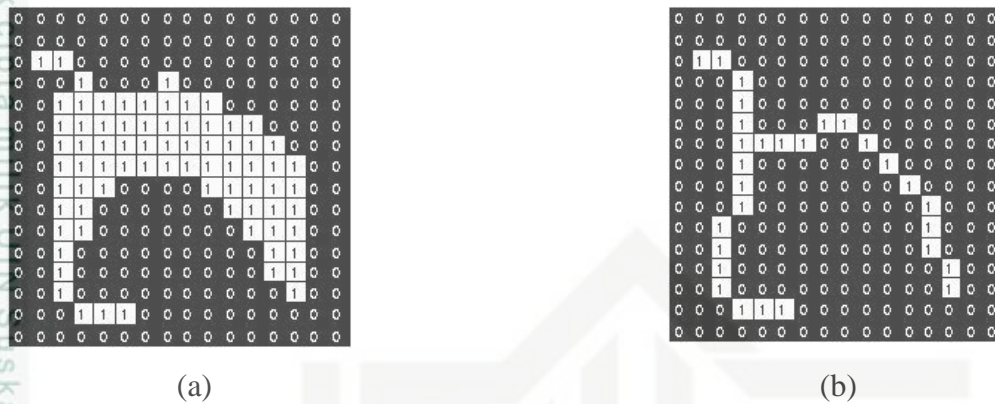
Berikut merupakan penjelasan untuk langkah-langkah *thinning*:

1. Untuk langkah 1, beri tanda pada semua piksel 8-tetangga yang memenuhi kondisi (1) sampai (4). Piksel yang diberi tanda tidak akan dihapus hingga semua proses penandaan piksel selesai. Tujuannya untuk mencegah terjadinya perubahan struktur data.
2. Hapus piksel yang telah diberi tanda.
3. Untuk langkah 2, beri tanda pada piksel yang belum dihapus pada langkah 1. Penghapusan harus memenuhi kondisi (1) sampai (4).
4. Hapus piksel yang telah diberi tanda (Damayanti & Setiawan, 2013).

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Contoh *thinning* dapat dilihat pada gambar 2.8 berikut:



Gambar 2. 8 Contoh *thinning*

- (a) Citra biner asal
- (b) Citra *thinning*

2.6. Modified Direction Feature (MDF)

MDF merupakan metode ekstraksi ciri pengembangan dari metode *Direction Feature* (DF), yaitu dengan menggabungkan metode *Direction Feature* (DF) dan *Transition Feature* (TF). MDF ini akan menghasilkan vektor ciri dimana vektor ciri tersebut didasarkan pada nilai transisi dan pelabelan nilai arah yang terdapat pada piksel *foreground* citra.

Untuk mencari nilai vektor ciri yang akan diproses, terlebih dahulu carilah nilai arah (DF) dan nilai transisi (TF). Nilai TF dan DF diambil dari 4 segmen arah yaitu, kiri ke kanan, kanan ke kiri, atas ke bawah, dan bawah ke atas (Riansyah, Nurhasanah, & Dewi, 2017).

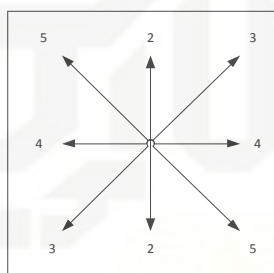
2.6.1. Menentukan nilai arah (*Direction Feature* (DF))

Pada DF yang dilakukan adalah melakukan pencarian nilai *feature* berdasarkan *label* arah dari sebuah piksel *foreground* (piksel karakter) Setiap piksel *foreground* (piksel karakter) pada citra akan diberi label (*feature* arah) dengan cara menelusuri tetangga dari masing-masing piksel *foreground* searah perputaran jarum jam. Nilai arah dapat dilihat pada tabel 2.2 berikut (Riansyah, Nurhasanah, & Dewi, 2017):

Tabel 2. 2 Nilai Label Arah pada DF

Arah	Nilai	Bentuk
Vertikal	2	
Diagonal Kanan	3	/
Horizontal	4	—
Diagonal Kiri	5	\

Pelabelan arah pada DF dapat dilihat pada gambar 2.9 berikut:



Gambar 2. 9 Pelabelan Arah pada DF

Langkah-langkah pelabelan arah pada masing-masing piksel sebagai berikut:

1. Lakukanlah pengecekan secara raster dari kiri ke kanan
2. Jika ditemukan sebuah piksel *foreground*, lakukanlah pengecekan dengan melihat tetangga dari piksel tersebut. Contoh pelabelan dapat dilihat pada tabel matriks ketetanggaan piksel berikut:

Tabel 2. 3 Matriks Ketetanggaan Penentuan Nilai Label

X1	X2	X3
X8	O	X4
X7	X6	X5

3. Jika O merupakan sebuah *foreground*, maka nilai arah O didapatkan dengan melakukan pengecekan secara berurutan dari X1-X8. Pengecekan akan berhenti jika sudah ditemukannya X pertama yang merupakan *foreground*. Ubahlah nilai O menjadi nilai arah berdasarkan aturan dibawah ini:

- 1) Jika berada pada posisi X1 atau X5 maka nilai arah adalah 5
- 2) Jika berada pada posisi X2 atau X6 maka nilai arah adalah 2
- 3) Jika berada pada posisi X3 atau X7 maka nilai arah adalah 3

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

4) Jika berada pada posisi X4 atau X8 maka nilai arah adalah 4

2.6.2. Menentukan nilai transisi (*Transition Feature (TF)*)

Nilai transisi (TF) merupakan nilai dari pembagian antara posisi dari transisi dengan panjang atau lebar dari citra. Transisi merupakan posisi dimana terjadinya piksel dari *background* berubah menjadi piksel *foreground* namun tidak sebaliknya. Mencari nilai transisi dilakukan secara transversal dari empat arah, yaitu kanan ke kiri, kiri ke kanan, atas ke bawah dan bawah ke atas. Nilai *longitude transition (LT)* pada TF didapatkan melalui pembagian antara posisi transisi dengan panjang ataupun lebar dari citra. Nilai transisi dari masing-masing arah akan bernilai antara 0-1, dimana nilai tersebut selalu menurun.

Jumlah transisi tidak boleh melebihi jumlah transisi maksimal, jika ditemukan jumlah transisi melebihi jumlah transisi maksimal maka tidak dilakukan perhitungan pada transisi tersebut. Sebaliknya, jika ditemukan jumlah transisi kurang dari jumlah transisi maksimal maka berikan nilai 0 pada transisi sisanya. Berikut adalah persamaan untuk perhitungan nilai LT jika pemindaian dilakukan dari arah kiri ke kanan dan dari atas ke bawah (Riansyah, Nurhasanah, & Dewi, 2017):

$$LT_i = 1 - \left(\frac{x_i}{Max_i} \right) \dots\dots\dots(2. 3)$$

Sedangkan persamaan untuk perhitungan nilai LT jika pemindaian dilakukan dari arah kanan ke kiri dan dari bawah ke atas:

$$LT_i = \left(\frac{x_i}{Max_i} \right) \dots\dots\dots(2. 4)$$

Pada kedua persamaan tersebut, *x_i* merupakan indeks piksel yang dikaji dihitung dari awal pencarian, sedangkan *Max_i* merupakan jumlah piksel maksimal dalam satu baris atau kolom segmen citra piksel citra yang mengikuti aturan berikut:

1. Jika pemindaian yang dilakukan dari kiri ke kanan dan dari kanan ke kiri maka nilai *Max_i* adalah lebar citra
2. Jika pemindaian yang dilakukan dari atas ke bawah dan dari bawah ke atas maka nilai *Max_i* adalah panjang citra

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Selain mencari nilai TF yang dihasilkan dari perhitungan masing-masing LT, perhitungan *direction transition* (DT) pada DF juga dilakukan untuk setiap transisi yang terjadi pada TF. Apabila pada TF ditemukan suatu transisi, maka DF untuk *foreground* yang bersangkutan juga akan dihitung.

Nilai dari DT dari DF ini diambil dari pembagian anatra nilai arah dengan nilai pembagi dimana nilai pembagi yang digunakan adalah 10. Nilai 10 ini digunakan untuk mendapatkan rentang nilai antara 0-1. Persamaan DT sebagai berikut:

$$DT_i = \left(\frac{\text{Nilai_Arah}(DF)_i}{10} \right) \dots\dots\dots(2. 5)$$

Setelah semua nilai dari DF dan TF dari 4 arah ditemukan, selanjutnya dengan melakukan normalisasi vektor ciri yang didapat pada setiap arah pencarian dengan persamaan:

$$nrFeature \times nrTransitions \times nrVectors \times nrMatrixHeight(Width) \dots(2. 6)$$

Keterangan:

- nrFeature* = 2 (TF dan DF)
- nrTransitions* = jumlah transisi yang digunakan
- nrVectors* = 4 (jumlah arah pencarian)
- nrMatrixHeight(Width)* = jumlah ukuran normalisasi matriks

2.7. Jaringan Syaraf Tiruan

Jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk melakukan simulasi proses pembelajaran pada otak manusia tersebut. Sistem pemrosesan dalam jaringan syaraf tiruan dirancang dengan menirukan cara kerja dari otak manusia dalam menyelesaikan suatu permasalahan dengan melakukan suatu proses pembelajaran. Pembelajaran atau pengenalan kegiatan dari data masa lalu dapat dilakukan oleh jaringan syaraf tiruan sehingga membuat jaringan syaraf tiruan mempunyai kemampuan untuk memberikan suatu keputusan pada data yang belum pernah dipelajari (Kusumadewi, 2003).

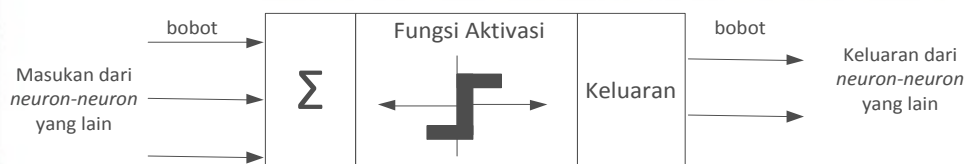
Menurut Medsker dan Liebowitz pada tahun 1994 dalam Turban pada tahun 2001, hubungan antara jaringan syaraf biologis dengan jaringan syaraf tiruan dapat dilihat pada tabel 2.4 berikut (Desiani & Arhami, 2006):

Tabel 2. 4 Hubungan Jaringan Syaraf Biologis dengan Jaringan Syaraf Tiruan

Jaringan Syaraf Biologis	Jaringan Syaraf Tiruan
Soma	Node (simpul)
Dendrit	Masukan
Axon	Keluaran
Sinapsis	Bobot
Kecepatan lambat	Kecepatan cepat
Terdiri dari banyak neuron (10^9)	Beberapa neuron

Jaringan syaraf tiruan ini sebagai tercipta sebagai turunan dari model matematis dari kesadaran manusia atau pemahaman manusia yang berdasarkan pada asumsi sebagai berikut:

1. Neuron merupakan elemen sederhana tempat terjadinya pemrosesan sautu informasi
2. Aliran sinyal diantara sel syaraf atau *neuron* melalui suatu sambungan penghubung.
3. Setiap sambungan penghubung memiliki bobot yang saling terhubung yang akan menambahkan atau mengurangi sinyal yang ditransmisi.
4. Adanya fungsi aktivasi pada setiap sel syaraf atau *neuron* yang diterapkan terhadap jumlah bobot dari data masukan untuk menentukan hasil keluarannya. Hasil keluaran akan dibandingkan dengan nilai *window* atau *threshold* (T. Sutojo, 2011).



Gambar 2. 10 Model Matematis Jaringan Syaraf Tiruan

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.7.1. Paradigma Pembelajaran Jaringan Syaraf Tiruan

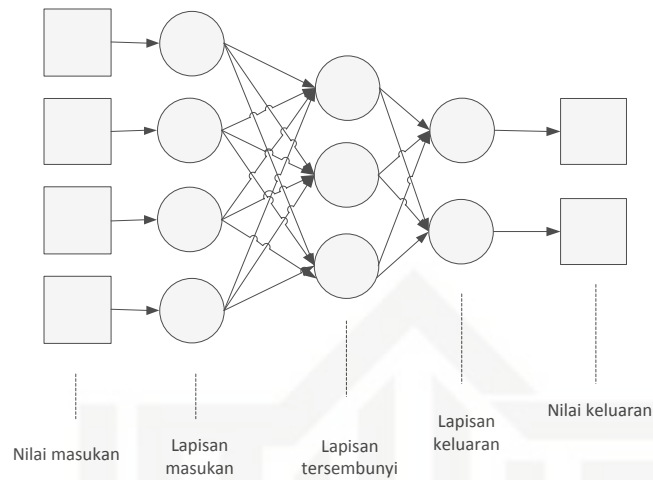
Jaringan syaraf tiruan memiliki 2 paradigma dalam model pembelajaran atau pelatihan, yaitu:

1. *Supervised learning* (pembelajaran terawasi) yang memiliki pengertian kumpulan masukan yang berusaha untuk membentuk target keluaran yang sudah diketahui sebelumnya (mengacu pada data *learning*). Pembelajaran ini menggunakan sekumpulan pasangan data masukan dan data keluaran yang diharapkan.
2. *Unsupervised learning* (pembelajaran tak terawasi). Jaringan mengorganisasikan dirinya untuk membentuk vektor-vektor masukan tanpa harus menggunakan data atau contoh-contoh pelatihan, umumnya ada pada umpan balik.. jaringan ini hanya menggunakan sekumpulan pasangan data masukan tanpa mempunyai contoh data keluaran yang diharapkan (Desiani & Arhami, 2006).

2.7.2. Arsitektur Jaringan Syaraf Tiruan

Suatu model jaringan syaraf tiruan ditentukan oleh hubungan antar *neuron* atau bisa disebut juga dengan arsitektur jaringan. *Neuron* layer merupakan lapisan-lapisan tempat berkumpulnya *neuron-neuron*. Lapisan-lapisan penyusun jaringan syaraf tiruan dibagi menjadi tiga bagian, yaitu:

1. Lapisan Masukan (*Input Layer*). Unit-unit dalam lapisan masukan bertugas sebagai penerima pola masukan dari luar yang merepresentasikan suatu permasalahan.
2. Lapisan Tersembunyi (*Hidden Layer*). Nilai keluaran dari dari unit-unit tersembunyi tidak dapat diamati secara langsung .
3. Lapisan Keluaran (*Output Layer*). Unit-unit keluaran ini merupakan solusi dari jaringan syaraf tiruan terhadap permasalahan yang dihadapi (T. Sutojo, 2011).



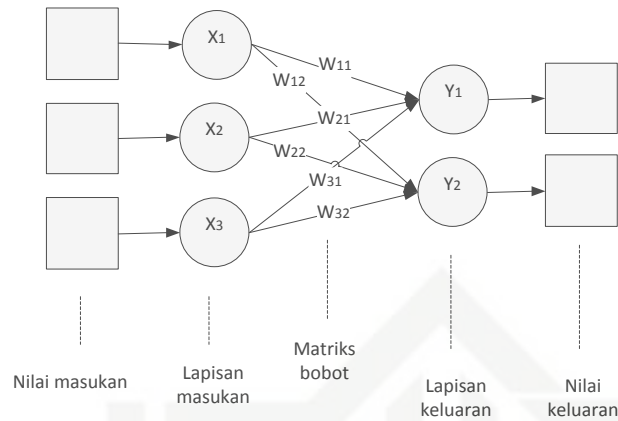
Gambar 2. 11 Tiga Lapisan pada Jaringan Syaraf Tiruan

Faktor yang paling penting dalam melakukan penentuan kelakuan suatu *neuron* adalah fungsi aktivasi dan pola dari bobot. *Neuron* akan memiliki fungsi aktivasi yang sama jika setiap lapisan *neuron* tersebut sama. Jika *neuron* dalam suatu lapisan (lapisan tersembunyi) dihubungkan dengan *neuron* pada lapisan lain (lapisan keluaran) maka setiap *neuron* pada lapisan tersebut (lapisan tersembunyi) harus dihubungkan dengan setiap lapisan pada lapisan lainnya (lapisan keluaran) (Kusumadewi, 2003).

Menurut Kusumadewi pada tahun 2003, beberapa jenis arsitektur jaringan pada jaringan syaraf tiruan, yaitu:

1. Jaringan dengan Lapisan Tunggal (*Single Layer Net*)

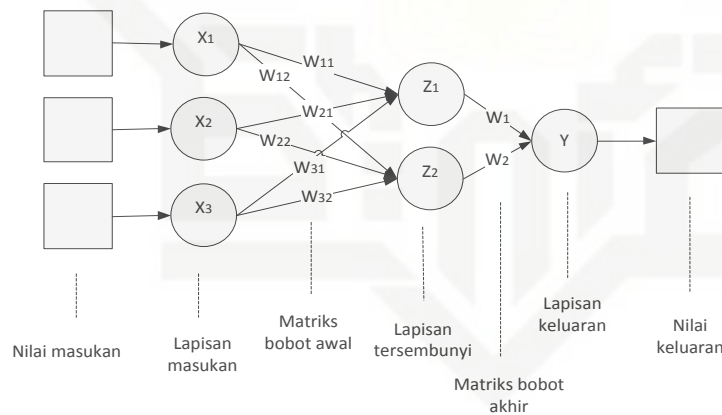
Jaringan dengan lapisan ini hanya mempunyai satu lapisan masukan dan satu lapisan keluaran yang bobot-bobotnya saling terhubung. Masukan yang diterima pada jaringan ini akan langsung diolah menjadi keluaran tanpa harus melewati jaringan tersembunyi.



Gambar 2. 12 Jaringan dengan Lapisan Tunggal (Single Layer Net)

2. Jaringan dengan Lapisan Banyak (Multilayer Net)

Jaringan dengan lapisan ini memiliki 3 buah lapisan yaitu, lapisan masukan, lapisan tersembunyi, dan lapisan keluaran. Pembelajaran pada algoritma ini rumit, dikarenakan jaringan ini mampu menyelesaikan suatu permasalahan yang lebih sulit daripada jaringan dengan lapisan tunggal.



Gambar 2. 13 Jaringan dengan Lapisan Banyak (Multilayer Net)

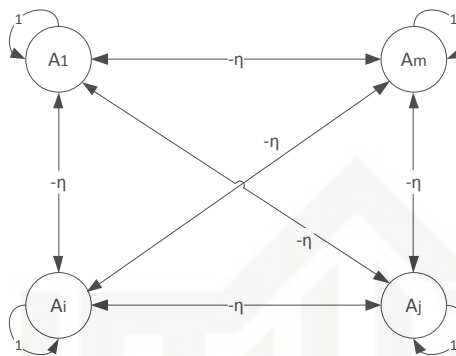
3. Jaringan dengan Lapisan Kompetitif (Competitive Layer)

Jaringan dengan lapisan ini mempunyai bobot yang sudah ditentukan dan tidak mempunyai proses pelatihan. Kegunaan dari jaringan ini adalah untuk mengetahui *neuron* pemenang dari sekumpulan *neuron* yang ada. Pada jaringan dengan lapisan kompetitif ini, sekumpulan *neuron* akan bersaing untuk mendapatkan haknya menjadi *neuron* aktif. Nilai dari bobot masing-

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

masing *neuron* untuk dirinya sendiri bernilai 1 dan untuk *neuron* yang lain akan bernilai random negatif.

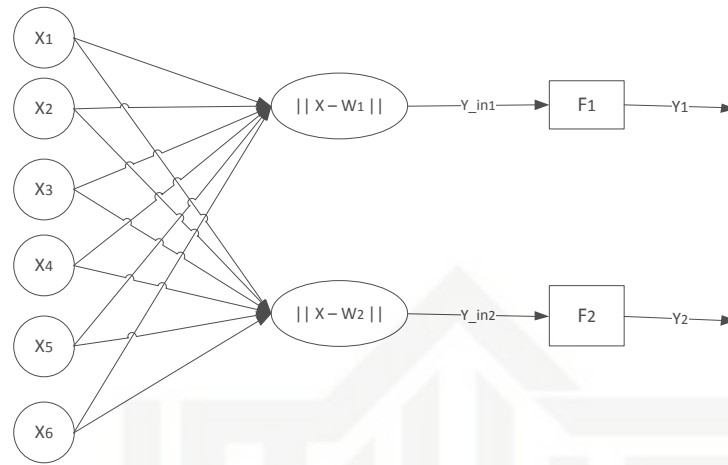


Gambar 2. 14 Jaringan dengan Lapisan Kompetitif (*Competitif Layer Net*)

2.8. Learning Vector Quantization (LVQ)

Learning Vector Quantization (LVQ) adalah jenis *Neural Network* yang dikembangkan oleh Teuvo Kohonen pada tahun 1989. LVQ merupakan algoritma klasifikasi prototipe *supervised* dengan aturan *Competitive Learning* versi dari algoritma *Kohonen Self-Organizing Map* (SOM). Proses pembelajaran pada metode LVQ ini dilatih secara signifikan supaya menghasilkan performansi yang lebih cepat dibandingkan algoritma lain (Fausett, 1994).

LVQ merupakan suatu metode yang berguna untuk melakukan suatu proses pembelajaran terhadap lapisan kompetitif yang terawasi. Lapisan kompetitif ini akan belajar untuk melakukan klasifikasi vektor-vektor masukan secara otomatis. Kelas-kelas yang akan dihasilkan oleh lapisan kompetitif ini hanya bergantung dari jarak antara vektor-vektor masukan. Jika kedua vektor masukan mendekati sama, maka lapisan kompetitif akan mengelompokkan vektor-vektor tersebut kedalam kelas yang sama. (Kusumadewi, 2002 dikutip oleh Ramadona, dkk 2004)



Gambar 2. 15 Arsitektur dari *Learning Vector Quantization (LVQ)*

Pada gambar 2.15 menunjukkan bahwa jaringan pada LVQ dengan 6 buah unit (*neuron*) pada lapisan masukan, 2 unit (*neuron*) pada lapisan keluaran. Proses yang akan terjadi pada masing-masing *neuron* adalah mencari jarak antara suatu vektor masukan ke bobot yang berkaitan (W_1 dan W_2). Pada W_1 yang merupakan vektor bobot yang dapat menghubungkan masing-masing dari *neuron* pada lapisan masukan ke *neuron* pertama pada lapisan keluaran, sedangkan pada W_2 yang merupakan vektor bobot yang dapat menghubungkan masing-masing *neuron* pada lapisan masukan ke *neuron* kedua lapisan keluaran. Fungsi aktivasi F_1 akan memetakan y_{in1} ke Y_1 apabila $|X - W_1| < |X - W_2|$. Hal ini juga akan terjadi pada F_2 . Fungsi aktivasi linear $y=x$ (Kusumadewi, Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK, 2004).

2.8.1. Learning Vector Quantization 3 (LVQ 3)

Learning Vector Quantization 3 atau biasa disebut dengan LVQ 3 merupakan salah satu variasi dari algoritma LVQ setelah algoritma LVQ 2 dan LVQ 2.1. Dalam algoritma ini, dua vektor yang paling dekat atau berdekatan diperbolehkan melakukan pembelajaran selama vektor dari masukan memenuhi kondisi berikut (Fausett, 1994):

$$\text{Min} \left[\frac{dc1}{dc2}, \frac{dc2}{dc1} \right] > (1-\epsilon) (1+\epsilon) \dots \dots \dots (2. 7)$$

Langkah-langkah algoritma LVQ 3, sebagai berikut:

1. Tentukan X sebagai vektor-vektor pelatihan. ($X_1, \dots, X_i, \dots, X_n$)

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2. T merupakan kategori atau kelas yang benar untuk vektor-vektor pelatihan.
3. W_j merupakan vektor dari bobot pada unit keluaran ke- j ($W_{1j}, \dots, W_{ij}, \dots, W_{nj}$)
4. C_j merupakan kategori atau kelas yang merepresentasikan oleh unit keluaran ke- j
5. Tentukan nilai dari *learning rate* (α). Nilai dari α adalah $0 < \alpha < 1$.
6. Penurunan tingkat pembelajaran menggunakan nilai pengurangan dari *learning rate* (α) 0.1.
7. Tentukan jarak minimum menggunakan jarak *euclidean* dengan persamaan (2.8).

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \dots\dots\dots(2.8)$$
8. Nilai minimal dari tingkat pembelajaran yang masih diperbolehkan menggunakan nilai minimal dari *learning rate* (Min. α).
9. Nilai yang digunakan sebagai daerah yang harus dipenuhi untuk melakukan perbaharuan vektor pemenang (W_{c1}) dan *runner-up* (W_{c2}) apabila berada pada kelas yang berbeda. Nilai ini menggunakan nilai *window* (ϵ). Persamaannya dapat dilihat pada persamaan (2.7) .
10. Apabila telah memenuhi kondisi dari *window* (ϵ) dan bernilai *TRUE*, vektor referensi yang tak masuk ke dalam kelas yang sama dengan vektor x akan diperbaharui atau dimodifikasi dengan persamaan:

$$W_{c1}(t + 1) = W_{c1}(t) - \alpha(t)[x(t) - W_{c1}(t)] \dots\dots\dots(2.9)$$

$$W_{c2}(t + 1) = W_{c2}(t) + \alpha(t)[x(t) - W_{c2}(t)] \dots\dots\dots(2.10)$$
 Sebaliknya, bernilai *FALSE*. Bobot akan dimodifikasi menjadi:

$$W_{c1}(t + 1) = W_{c1}(t) - \beta(t)[x(t) - W_{c1}(t)] \dots\dots\dots(2.11)$$

$$W_{c2}(t + 1) = W_{c2}(t) + \beta(t)[x(t) - W_{c2}(t)] \dots\dots\dots(2.12)$$
 Nilai pada $\beta(t)$ merupakan hasil perkalian dari $\epsilon\alpha(t)$. Nilai ϵ atau nilai *window* yang digunakan adalah 0.2 (Fausett, 1994).
11. Setelah itu, akan menghasilkan bobot akhir (W). Yang selanjutnya bobot-bobot ini akan digunakan untuk melakukan proses pengujian (Budianita & Prijodiprodjo, 2013).

2.9. Persentase Akurasi

Persentase dari keberhasilan dari sistem yang akan dibangun dihitung dengan menggunakan persamaan *confusion matrix accuracy*. *Confusion matrix accuracy* merupakan sebuah tabel yang menyatakan jumlah data uji benar diklasifikasikan dan jumlah data uji salah diklasifikasikan. Contoh *confusion matrix accuracy* untuk biner ditunjukkan pada tabel 2.5 berikut (Indriani, 2014):

Tabel 2. 5 Confusion Matrix Accuracy untuk Klasifikasi Biner

		Kelas Prediksi	
		1	0
Kelas Sebenarnya	1	TP	FN
	0	FP	TN

Keterangan:

True Positive (TP) merupakan jumlah dokumen dari kelas 1 yang benar dan diklasifikasikan sebagai kelas 1.

True Negative (TN) merupakan jumlah dokumen dari kelas 0 yang benar dan diklasifikasikan sebagai kelas 0.

False Positive (FP) merupakan jumlah dokumen dari kelas 0 yang salah dan diklasifikasikan sebagai kelas 1.

False Negative (FN) merupakan jumlah dokumen dari kelas 1 yang salah dan diklasifikasikan sebagai kelas 0.

$$Akurasi = \frac{TP+TN}{TP+FN+FP+TN} \times 100\% \dots \dots \dots (2. 13)$$

2.10. Penelitian Terkait

Penelitian lain yang terkait dengan penelitian yang akan diteliti oleh penulis dapat dilihat pada tabel 2.6 berikut:

Tabel 2. 6 Beberapa Penelitian Terkait

No	Nama Peneliti	Judul Penelitian	Metode Penelitian	Hasil Penelitian
1	Cheng Lin Liu, In Jung Kim, Jin H. Kim (1997)	<i>High Accuracy Handwritten Chinese Character Recognition</i>	<i>Modified Direction Featutre (MDF) dan Learning Vector</i>	Tingkat akurasi yang dihasilkan pada pengenalan pola karakter cina dengan 100 citra uji sebesar 95.73%.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

		<i>by Improved Feature Matching Method</i>	<i>Quantization 3 (LVQ 3)</i>	
2	M. Blumenstein, X.Y. Liu dan B. Verma (2003)	<i>A Modified Direction Feature for Cursive Character Recognition</i>	<i>Modified Direction Featutre (MDF), Direction Featutre Extraction (DFE), Multi Layer Perceptron (MLP) dan Radial Basis Function (RBF)</i>	Pada penelitian ini dilakukan dengan membandingkan metode ekstraksi ciri MDF dan DF serta metode klasifikasi jaringan syaraf tiruan <i>Multi Layer Perceptron (MLP)</i> dan <i>Radial Basis Function (RBF)</i> dalam pengenalan karakter tulis tangan yang berasal dari database CEDAR menghasilkan tingkat akurasi yang dihasilkan oleh MDF lebih baik yaitu sebesar 80.83% untuk huruf besar dan 70.22% pada huruf kecil. Sedangkan untuk hasil akurasi pada DFE sebesar 77.32% untuk huruf besar dan 69.73% untuk huruf kecil.
2	Seong-Whan Lee & Hee-Heon Song (2007)	<i>Optimal Design of Reference Models Using Simulated Annealing Combined an Improved LVQ3.</i>	<i>Learning Vector Quantization 2.1 (LVQ 2.1) dan Learning Vector Quantization 3 (LVQ 3)</i>	Pada penelitian ini melakukan perbandingan antara LVQ 2.1 dan LVQ 3 untuk kasus pengenalan pola karakter huruf Korea (Hangul). Hasil penelitian tersebut yaitu pada jumlah data latih 20 menyatakan bahwa LVQ 2.1 memiliki tingkat error 16% sedangkan tingkat error LVQ 3 12%, pada jumlah data latih 30 menyatakan

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

				bahwa LVQ 2.1 memiliki tingkat error 14% sedangkan tingkat error LVQ 3 12%, dan pada jumlah data latih 50 menyatakan bahwa LVQ 2.1 memiliki tingkat error 12% sedangkan tingkat error LVQ 3 10%. Hasil akurasi LVQ 2.1 sebesar 88% dan LVQ 3 sebesar 90%.
3	Maharani Dessy Wuryandari dan Irawan Afrianto (2012)	Perbandingan Metode Jaringan Syaraf Tiruan <i>Backpropagation</i> dan <i>Learning Vector Quantization</i> Pada Pengenalan Wajah	<i>Backpropagation</i> dan <i>Learning Vector Quantization</i>	Tingkat kecocokan pengenalan wajah dengan <i>backpropagation</i> sebesar 37.33% dengan rata-rata waktu pengenalan 130 milisecond dan pada LVQ sebesar 37.63% dengan rata-rata waktu pengenalan 32 milisecond
4	Welli Andrian (2012)	Analisa dan Implementasi Metode <i>Kohonen Neural Network</i> untuk Pengenalan Karakter Huruf Arab	<i>Kohonen Neural Network</i>	Dengan total sampel pengujian 310 karakter, mempunyai persentase <i>error</i> 43,64% untuk kategori huruf arab diawal, 22.73% untuk huruf tunggal (<i>isolated</i>), 24% ditengah kata (<i>middle</i>), dan 8,57% diakhir kata (<i>end</i>).
5	Tjokorda Agung BW, I Gede Rudy Hermanto, Retno Novi D (2009)	Pengenalan Huruf Bali Menggunakan Metode <i>Modified Direction Feature</i>	<i>Modified Direction Feature</i> (MDF) dan <i>Learning Vector Quantization</i> (LVQ)	Penelitian ini menghasilkan akurasi data latih sebesar 88.98% dan data uji sebesar 81.84%. Tingkat akurasi tersebut dipengaruhi oleh ukuran normalisasi,

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

		(MDF) dan <i>Learning Vector Quantization</i> (LVQ)		jumlah transisi, dan pembagian wilayah. Dimana dalam penelitian ini konfigurasi MDF yang menghasilkan akurasi terbaik adalah ukuran normalisasi 100x50, jumlah transisi 4, dan tidak dilakukannya pembagian gambar menjadi beberapa bagian.
6	Elvia Budianita, Widodo Prijodiprojjo (2013)	Penerapan <i>Learning Vector Quantization</i> (LVQ) untuk Klasifikasi Status Gizi Anak	<i>Learning Vector Quantization</i> (LVQ)	Algoritma LVQ 3 memiliki akurasi yang lebih baik daripada LVQ 1, penggunaan parameter <i>window</i> (ϵ) pada algoritma LVQ 3 memberikan hasil yang baik yaitu mampu meningkatkan performansi akurasi dalam klasifikasi jika dibandingkan dengan algoritma LVQ 1 tanpa parameter <i>window</i> (ϵ). Hal ini dibuktikan pada pengujian yaitu nilai parameter yang digunakan pada LVQ 3 yang terdiri dari <i>learning rate</i> (α) = 0.05, nilai minimal <i>learning rate</i> (Min. α) = 0.02, nilai pengurangan α = 0.1, dan nilai dari <i>window</i> (ϵ) = 0.2, nilai-nilai ini sudah cukup efektif dan efisien karena telah sesuai target yang ingin dicapai secara menyeluruh (nilai akurasi

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

				mencapai 100%). Pengujian selanjutnya dilakukan dengan pengujian jumlah data latih. Pengujian ini menghasilkan rata-rata akurasi pada LVQ 3 adalah 95.2% sedangkan nilairata-rata akurasi pada LVQ 1 adalah 88%
7	Keumala Anggraini (2015)	Penerapan Jaringan Syaraf Tiruan Metode <i>Learning Vector Quantization</i> (LVQ) untuk Pengenalan Karakter Huruf Hijaiyah	<i>Learning Vector Quantization</i> (LVQ)	Persentase <i>error</i> yang dihasilkan sebesar 23.57% untuk kategori huruf arab tunggal, 34.55% untuk kategori huruf arab diawal, 37.27% untuk huruf arab ditengah kata (<i>middle</i>) dan diakhir kata (<i>end</i>) 20%. Dengan rata-rata persentase <i>error</i> 28%. Dan akurasi yang dihasilkan sebesar 76.43% untuk kategori huruf arab tunggal, 65.45% untuk kategori huruf arab diawal, 62.73% untuk huruf arab ditengah kata (<i>middle</i>) dan diakhir kata (<i>end</i>) 80%. Dengan rata-rata persentase akurasi 72%.