

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 mengikuti 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.

BAB II

LANDASAN TEORI

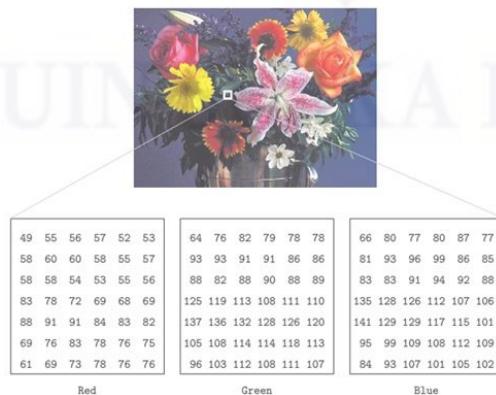
2.1 Citra

Secara harfiah, citra (*image*) adalah gambar pada bidang dua dimensi. Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dua dimensi tersebut. Sumber cahaya menerangi objek, objek tersebut memantulkan kembali sebagian dari berkas cahaya tersebut. Sehingga cahaya yang dipantulkan tersebut dapat diterima oleh alat optic seperti mata, kamera, scanner, dan lain-lain, sehingga bayangan objek tersebut direkam.(Munir, 2004)

Beberapa jenis citra berdasarkan nilai pixelnya adalah :

1. Citra warna

Pada citra warna masing-masing piksel memiliki warna tertentu, warna tersebut adalah merah (*red*), hijau (*green*), biru (*blue*). Jika pada masing-masing warna memiliki *range* dari 0-255, maka total warnanya adalah $255^3 = 16.581.375$ variasi warna berbeda pada gambar. Seperti terlihat pada gambar dibawah ini.(Kusumanto& Tompunu, 2011)



Gambar 2.1 Citra Warna

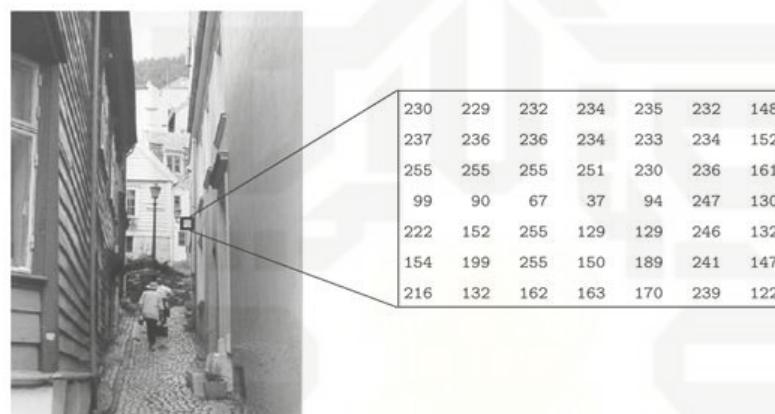
Hak Cipta Dilindungi Undang-Undang

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 mengikuti 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. Citra *Grayscale*

Pada citra *grayscale* merupakan sebuah citra yang gradasi warnanya mulai dari putih ke hitam. Rentang tersebut berarti bahwa setiap piksel dapat diwakili oleh 8 bit atau 1 byte. Salah satu fungsi dari citra grayscale ini adalah dalam dunia kedokteran (*X-ray*). Black and white merupakan hasil rata-rata dari citra warna.

Berikut ini merupakan contoh gambar dari citra *grayscale* sebagai berikut.

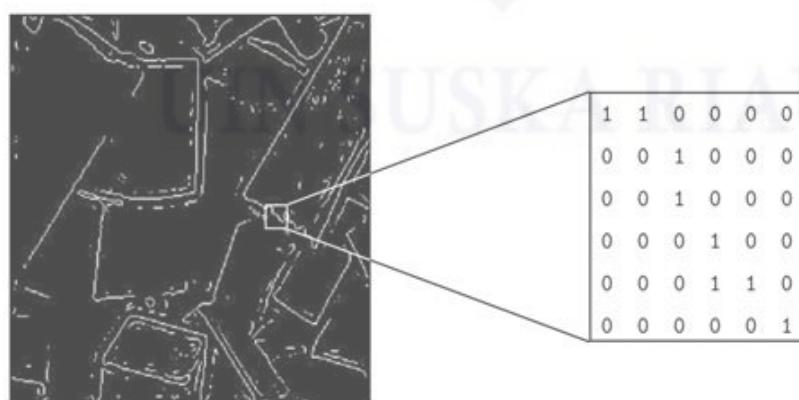


Gambar 2.2 Citra Grayscale(Kusumanto& Tompunu, 2011)

3. Citra Biner

Pada citra biner hanya terdiri dari warna hitam dan putih, karena hanya ada 2 warna pada setiap piksel. Maka hanya perlu 1 bit per piksel. Sehingga sangat efisien dalam hal penyimpanan. Gambar yang di representasikan dengan biner sangat cocok untuk teks, tanda tangan, dan arsitektur gambar.

Berikut ini merupakan contoh dari citra biner sebagai berikut.



Gambar 2.3Citra Biner (Kusumanto & Tompunu, 2011)



- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

2.2 Pengolahan Citra

Pengolahan citra menurut Gonzalez yang dikutip oleh SiskaFathia merupakan daerah yang ditandai dengan kebutuhan untuk pekerjaan eksperimental yang luas untuk menetapkan kelayakan dari solusi yang diusulkan untuk masalah yang diberikan. Sedangkan pengenalan pola menurut Handoyo, EricoDarmawan dan Lydia Wiguna Susanto adalah salah satu bidang studi yang berkaitan dengan pengolahan citra digital dibidang komputer.(Fathia, 2013)

Pengolahan citra digital menurut Sutoyo dan kawan-kawan yang dikutip oleh RD. Kusumantodan A.Tompunu adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar yang bergerak. Sedangkan yang dimaksud dengan digital adalah pengolahan citra yang menggunakan komputer. Secara matematis, citra merupakan fungsi kontinu dengan intensitas cahaya pada bidang dua dimensi. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numeric dengan nilai-nilai diskrit. Representasi dari fungsi nilai kontinyu menjadi nilai-nilai diskrit disebut digitalisasi citra.(Kusumanto& Tompunu, 2011)

2.2.1 Operasi Pengolahan Citra

Operasi-operasi yang dilakukan dalam mengolah citra digital banyak macamnya. Namun secara umum operasi pengolahan citra digital dapat diklasifikasikan dalam beberapa jenis, seperti berikut.

1. Perbaikan kualitas citra (*Image Enhancement*)

Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter pada citra. Dengan operasi ini, ciri-ciri khusus yang terdapat dalam citra dapat lebih ditonjolkan. Contoh operasi perbaikan citra.

- Perbaikan kontras
- Perbaikan tepian objek (*edge enhancement*)
- Penajaman (*sharpening*)
- Pemberian warna semu (*psuedocoloring*)

**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 mengikuti 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.

- e. Penapisan derau (*noise filtering*)

2. Restorasi Citra

Operasi ini bertujuan untuk menghilangkan kecacatan pada citra. Bedanya dengan perbaikan citra adalah penyebab degradasi gambar diketahui. Berikut adalah contoh restorasi citra.

- a. Penghilangan kesamaran (*deblurring*)

- b. Penghilangan derau (*noise*)

3. Kompresi Citra

Jenis operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang kompak atau solid sehingga memerlukan memori yang lebih sedikit. Hal yang penting yang harus diperhatikan dalam kompresi gambar adalah ketika gambar dikompres, maka gambar itu harus mempunyai kualitas gambar yang baik.

4. Segmentasi Citra

Jenis operasi ini bertujuan untuk memecah suatu citra kedalam beberapa segmen dengan suatu kriteria tertentu. Jenis ini berkaitan dekat dengan pengenalan pola.

5. Analisis Citra

Pada operasi ini proses yang dilakukan adalah menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsiya. Teknik analisis citra dengan cara mengekstraksi ciri tertentu untuk membantu dalam identifikasi objek.

Berikut adalah contoh analisis citra.

- a. Deteksi tepi (*edge detection*)

- b. Ekstraksi batas (*boundary*)

- c. Representasi daerah (*region*)

6. Rekonstruksi Citra

Pada operasi ini proses yang dilakukan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutip hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak mengikuti 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.

2.3 Pengenalan Pola

Pengenalan pola bertujuan untuk menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu objek dengan objek yang lain. (Munir, 2004)

Pendekatan ini menggunakan teori-teori ilmu peluang dan statistic. Ciri-ciri yang dimiliki oleh suatu pola ditentukan distribusi statistiknya. Pola yang berbeda memiliki distribusi yang berbeda pula. (Munir, 2004)

Sistem pengenalan pola dengan pendekatan statistic dapat dilihat pada skema gambar dibawah ini.



Gambar 2.4 Pendekatan Statistik(Munir, 2004)

2.3.1 *Pre-processing*

Tahap pre-processing merupakan tahap pertama yang dilakukan dalam pengenalan pola. Tahap pre-processing ini berguna untuk melakukan operasi-operasi pada citra untuk menghasilkan set data yang lebih konsisten. Pada tahap-tahap pengenalan pola ada beberapa proses yaitu konversi citra rgb ke grayscale, deteksi tepi canny, morfologi dilasi, dan resize.

Tahap pre-processing dilakukan untuk mempersiapkan data citra yang akan diolah pada proses *feature extraction*. Tahap yang dilakukan pertama pada proses pre-processing adalah sebagai berikut.

1. Cropping

Proses ini berfungsi untuk menghilangkan bagian citra yang tidak dibutuhkan dalam pengenalan pola, seperti bagian background yang kosong sehingga proses yang dilakukan lebih spesifik.



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 mengikuti 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. Konversi RGB ke *Grayscale*

Untuk melakukan proses konversi RGB ke grayscale dapat dilakukan dengan menggunakan rumus sebagai berikut.

$$\text{Grayscale} = (0,2989 * R) + (0,5870 * G) + (0,1141 * B) \dots \dots \dots \quad (2.1)$$

Keterangan :

R= merepresentasikan nilai merah pada citra

G= merepresentasikan nilai hijau pada citra

B= merepresentasikan nilai biru pada citra

3. Deteksi Tepi Canny

Metode *Canny* merupakan salah satu algoritma deteksi tepi modern. Deteksi tepi *Canny* ditemukan oleh Marr dan Hildreth yang meneliti pemodelan persepsi virtual manusia.

Ada beberapa kriteria dalam deteksi tepi paling optimum yang dapat dipenuhi oleh algoritma *Canny*.

a. Mendekripsi dengan baik (kriteria deteksi)

Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai dengan yang diinginkan.

b. Melokalisasi dengan baik (kriteria lokalisasi)

Dengan metode Canny dimungkinkan akan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.

c. Respon yang jelas (kriteria respon)

Hanya ada satu respon untuk setiap tepi. Sehingga mudah dideteksi dan tidak menimbulkan kerancuan dalam pengolahan citra selanjutnya.

Tahap-tahapan untuk melakukan deteksi tepi canny adalah sebagai berikut.

(Mazid Kamal, 2014)

a. *Smoothing* menggunakan *Filter Gaussian*

Proses ini berguna untuk mengurangi noise yang ada pada citra. (Oliveira, dkk 2016). Berikut rumus untuk menghitung gaussian filter.

b. *Finding Gradient*

Finding Gradient digunakan untuk mencari nilai tepi gambar dan arah tepi dengan melibatkan operator sobel sebagai *mask* untuk mencari nilai tepi. Mask sobel 3x3 digunakan untuk mencari nilai ketetanggaan arah x dan y.

Akar kuadrat dari hasil *mask* G_x dan G_y dihitung sebagai nilai magnitude

c. Tentukan Arah Tepian

Menentukan arah tepian yang ditemukan dengan menggunakan rumus sebagai berikut

Dari hasil yang didapatkan, maka nilai arah dapat ditentukan dengan rumus sebagai berikut.

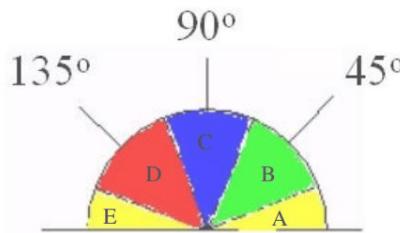
jika $0 \leq \theta \geq 22.5$ atau $157.5 \leq \theta \geq 180$ maka nilai_arah = 0

jika $22.5 \leq \theta \geq 67.5$ maka nilai_arah = 45

jika $67.5 \leq \theta \geq 112.5$ maka nilai_arah = 90

jika $112.5 \leq \theta \geq 157.5$ maka nilai_arah = 135(2.7)

Ilustrasi gambar hasil dari proses pencarian nilai arah gradien adalah seperti Gambar 2.2 berikut.



Gambar 2.5 Ilustrasi Gambar arah(Oliveira, dkk 2016)

d. Non Maximum Supremassion

Proses ini berfungsi untuk memperkecil garis tepi yang muncul sehingga menghasilkan garis tepian yang lebih ramping. Membuang potensi gradient disuatu piksel dari kandidat edge jika piksel tersebut bukan merupakan maksimal lokal dari arah edge diposisi piksel tersebut.

Jika nilai arah $(i, j) = 0$ maka $G(i, j) = \max(G(i, j), (i, j + 1), (i, j - 1))$

Jika nilai arah $(i, j) = 45$ maka $G(i, j) = \max(G(i, j), (i + 1, j - 1), (i - 1, j + 1))$

Jika nilai arah $(i, j) = 90$ maka $G(i, j) = \max(G(i, j), (i + 1, j), (i - 1, j))$

Jika nilai arah $(i, j) = 135$ maka $G(i, j) = \max(G(i, j), (i + 1, j + 1), (i - 1, j - 1))$

e. Hysteresis thresholding

Hasil dari langkah non-maximal suppression adalah citra yang berisi kandidat edge, serta intensitas dari kekuatan edge diposisi tersebut. Langkah terakhir adalah thresholding atau klasifikasi tiap piksel apakah termasuk kategori edge atau tidak. Sederhananya hysteresis thresholding merupakan klasifikasi dua buah nilai *high-threshold* dan *low-threshold*. Suatu piksel disahkan sebagai piksel edge jika nilainya lebih besar atau sama dengan *high-threshold* atau jika piksel tersebut memiliki intensitas kekuatan edge yang lebih besar dari *low-threshold* dan terhubung dengan piksel yang nilainya lebih besar dari nilai *high-threshold*.

$$T_{low} = 0,075$$

$$T_{High} = 0,175$$

$$T_{Low} = T_{Low} * \max(G)$$

$$T_{High} = T_{High} * \max(G)$$

Jika $G_{(i,j)} < T_{Low}$ maka $T = 0$

naka $T = 1$

Hasil dari nilai yang bernilai 1 akan dibaca sebagai tepi dari citra yang diproses.

4. Morfologi Dilasi

a. Dilasi

Dilasi merupakan proses dari pengolahan citra yaitu proses penebalan dari pada piksel-piksel citra yang bernilai 1. Dilasi mempunyai hukumkomutatif, yaitu $A \oplus B = B \oplus A$ (2.10)

Berikut ini ditunjukkan contoh proses dan hasil dilasi. Pada diagram pergeseran, daerah yang berwarna kelabu menunjukkan posisi awal dari objek yang akan dikenakan dilasi. Perhatikan bahwa $A(0,0)$ adalah A itu sendiri. Pada contoh ini diperoleh $B = \{(0,0),(-1,1),(1,-1),(-1,-1)\}$. Koordinat-koordinat yang tampak pada definisi tersebut merupakan koordinat dimana A di translasikan (digeser).

x/y	1	2	3	4	5
1					
2		●	●		
3		●	●		
4		●	●		
5		●	●	●	
6			●	●	
7					

A

x/y	-1	0	1
-1	●		●
0		●	
1	●		●

B

Untuk proses dilasi dapat dilihat pada proses berikut dengan menggunakan rumus diatas.

x/y	1	2	3	4	5
1					
2					
3			●	●	
4			●	●	
5			●	●	
6			●	●	●
7				●	●

A(1,1)

x/y	1	2	3	4	5
1			●	●	
2			●	●	
3			●	●	
4			●	●	●
5				●	●
6					
7					



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 mengikuti 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.

x/y	1	2	3	4	5
1					
2					
3	●	●			
4	●	●			
5	●	●			
6	●	●	●		
7		●	●		

A(1,-1)

x/y	1	2	3	4	5
1	●	●			
2	●	●			
3	●	●	●		
4	●	●	●	●	
5		●	●		
6					
7					

A(-1,-1)

x/y	1	2	3	4	5
1	●	●	●	●	
2	●	●	●	●	
3	●	●	●	●	●
4	●	●	●	●	●
5	●	●	●	●	●
6	●	●	●	●	●
7		●	●	●	●

A \oplus B

b. Fill

Proses fill merupakan proses pengisian nilai pada citra yang bernilai 0 yang dikelilingi oleh citra nilai 1.

pada gambar pertama terdapat beberapa piksel yang bernilai 0 dimana piksel tersebut terletak ditengah-tengah dari piksel yang bernilai 1, proses fill itu sendiri akan mengubah nilai 0 pada citra dimana nilai 0 tersebut dikelilingi oleh citra bernilai 1. Hasilnya dapat dilihat pada gambar kedua.

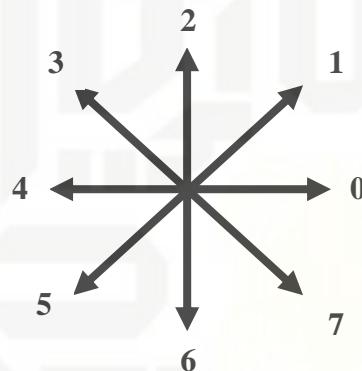
c. Resize

Proses resize merupakan proses perubahan ukuran piksel pada citra. Misalnya sebuah citra berukuran 100x100 diubah menjadi citra yang berukuran 10x10.

2.3.2 Feature Extraction Chain Code

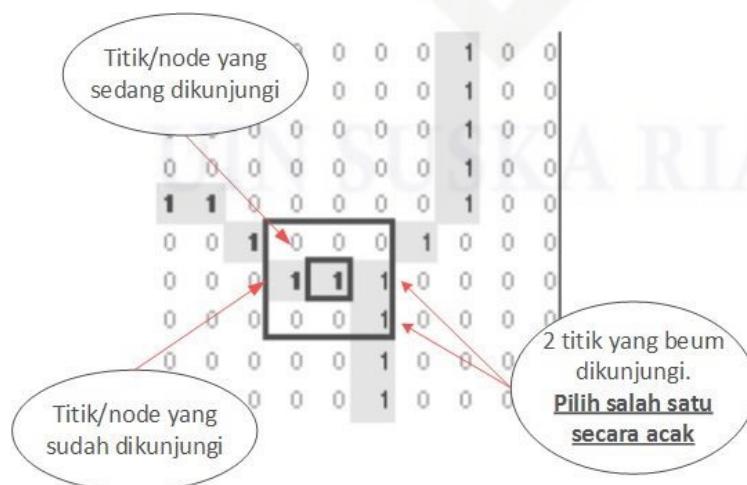
Untuk mengenali suatu pola dari karakter huruf didalam citra, dibutuhkan adanya ciri-ciri khusus. Setiap objek pasti mempunyai ciri-ciri yang berbeda

dengan karakter lain. Ciri khusus yang bagus adalah ciri yang memiliki daya pembeda yang tinggi. Sehingga pengelompokan berdasarkan ciri yang dimiliki dapat menghasilkan keakuratan yang tinggi. Ekstraksi ciri atau ekstraksi fitur adalah proses pengambilan ciri-ciri suatu objek didalam citra untuk membedakan objek satu dengan yang lainnya. Sebelum dilakukan ekstraksi fitur perlu dilakukan preprocessing pada citra seperti binerisasi, thinning, normalisasi. Chain code adalah algoritma yang melakukan penelusuran pixel-pixel objek dengan panduan arah mata angin. Seperti yang ditunjukkan pada gambar dibawah ini.



Gambar 2.6Arah Mata Angin Sebagai Panduan(Fitri Amalia Hadi, Tjokorda Agung Budi W, Kurniawan Nur Ramadhani, 2015)

Dengan mekanisme melakukan penelusuran piksel-piksel, teknik chain code dapat digunakan untuk menemukan struktur pembentuk dari suatu objek. Illustrasi proses chain code dapat dilihat pada gambar dibawah ini.



Gambar 2.7Ilustrasi kondisi pemilihan titik tetangga secara acak

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.

Hasil dari ekstraksi menggunakan chain code adalah vektor ciri yang berisi informasi urutan kode chain code pembentuk huruf. Dari mekanisme tersebut maka urutan dari kode chain kode setiap huruf akan memiliki panjang yang berbeda, sehingga perlunya dilakukan proses normalisasi untuk menyamakan panjang dari chain kode agar dapat digunakan sebagai input pada klasifikasi.

Proses normalisasi yang dilakukan adalah yaitu merubah chain code menjadi matriks yang berisi nilai dan frekuensi. Menghapus nilai dengan frekuensi 1, dengan menerapkan rumus normaliasi untuk mencari frekuensi yang diinginkan, dan membangun ulang chain code sesuai frekuensi yang baru. Nilai frekuensi yang baru kode ke i dapat ditetapkan dengan rumus normalisasi sebagai berikut.

jika $V_{awal} > V_{akhir}$, maka
 normalisasi = $V_{awal} - V_{akhir}$

Keterangan :

V_{awal} : Vektor awal hasil chain code

Vakhir : Vektor setelah normalisasi

Jp : Jarak pengurangan

2.3.3 Classification

Klasifikasi adalah proses pengelompokan data sesuai dengan kelas yang sudah ditentukan. Pada tahap ini objek akan diklasifikasi berdasarkan kesamaan fitur objek yang sudah diekstraksi pada tahapan sebelumnya. Klasifikasi menggunakan jaringan syaraf tiruan dengan metode Learning Vector Quantization 2:1.

2.4 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan merupakan teknik komputasi yang tepat karena teknik komputasi ini dikembangkan berdasarkan cara kerja sistem syaraf biologis manusia sehingga memiliki kemampuan untuk mempelajari dan mengenali



- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

sesuatu, walaupun terdapat gangguan dalam pengenalan seperti noise.(Husen et al., 2015)

Zekson Arizona Matondang berpendapat jaringan syaraf tiruan merupakan salah satu sistem pemrosesan infomasi yang di desain dengan meniru cara kerja otak manusia dalam menyelesaikan suatu masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya. Ada banyak teknik yang dapat digunakan dalam implementasi jaringan syaraf tiruan yaitu *perceptron*, *backpropagation*, *fuzzy*, *LVQ*. (Matondang, 2013)

M.F Andrijasa mengatakan bahwa jaringan syaraf tiruan merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran otak manusia.

Jaringan syaraf tiruan tercipta sebagai suatu generalisasi model matematis dari pemahaman yang didasarkan pada asumsi berikut.

- Pemrosesan informasi terjadi pada elemen sederhana pada neuron.
- Isyarat mengalir diantara sel syaraf / neuron melalui suatu sambungan penghubung.
- Setiap sambungan penghubung memiliki bobot yang bersesuaian.
- Setiap sel syaraf akan merupakan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya.(Jumarwanto, Hartanto, & Prastiyanto, 2009)

2.4.1 Arsitektur Jaringan Syaraf Tiruan

Berikut ini beberapa arsitektur jaringan yang sering dipakai dalam jaringan syaraf tiruan.

1. Jaringan Lapisan Tunggal (*Single Layer Network*)

Dalam jaringan ini, sekumpulan *neuron* dihubungkan langsung dengan sekumpulan outputnya. Dalam beberapa model, misalnya *perceptron* hanya ada sebuah unit *neuron output*.



2. Jaringan Lapisan Jamak (*Multi Layer Network*)

Jaringan layar jamak adalah perluasan dari jaringan lapisan tunggal. Dalam jaringan ini, selain unit input dan output, ada unit-unit lain (yang disebut layar tersembunyi atau *hidden layer*).

3. Jaringan Kompetitif (*Competitive layer Network*)

Jaringan model ini mirip dengan jaringan lapisan tunggal dan jamak. Hanya saja, ada neuron output yang memberikan sinyal pada unit input atau *feed loop*.

2.4.2 Pemodelan Jaringan Syaraf Tiruan

Jaringan syaraf tiruan terdiri atas beberapa neuron yang ada hubungannya dengan neuron-neuron tersebut. Neuron adalah sebuah unit pemroses informasi yang menjadi dasar dari pengoperasian jaringan syaraf tiruan. Syaraf adalah sebuah unit pemroses informasi dengan tiga elemen dasar yaitu :

1. Satu set link yang berhubungan.
2. Sebuah penjumlah untuk menghitung besarnya penambahan pada sinyal masukan.
3. Sebuah fungsi aktivasi untuk membatasi banyaknya keluaran pada syaraf.(Jumarwanto et al., 2009)

Proses pembelajaran dalam jaringan syaraf tiruan diklasifikasikan menjadi 2 bagian :

1. Pembelajaran terawasi atau *supervised learning*

Merupakan metode pembelajaran pada jaringan syaraf tiruan disebut terawasi jika *output* yang diharapkan telah diketahui sebelumnya. Pada proses pembelajaran, suatu pola input akan diberikan kesuatu neuron pada lapisan input. Pola ini akan dirambatkan disepanjang jaringan syaraf hingga sampai ke neuron pada lapisan *output*. Lapisan *output* ini membangkitkan pola *output* yang nantinya akan dicocokkan dengan pola *output* targetnya. Apabila terjadi perbedaan antara pola *output* hasil pembelajaran dengan pola target, maka disinilah muncul *error*. Apabila nilai *error* masih besar, maka perlu dilakukan pembelajaran lain.

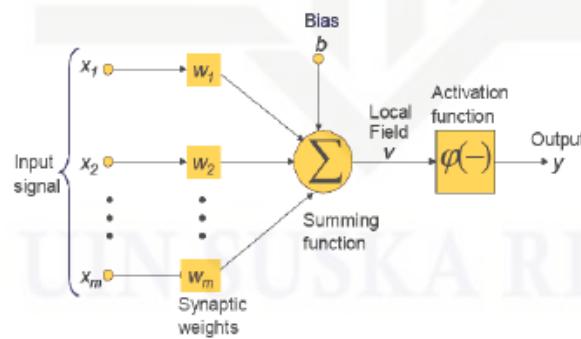
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak mengikuti kepentingan 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. Pembelajaran tidak terawasi atau *unsupervised learning*

Merupakan metode yang memerlukan *output* target. Pada metode ini tidak dapat ditentukan hasil yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai output yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk klasifikasi atau pengelompokan suatu pola. (Kusumadewi, 2003)

Dalam lapisan kompetitif, proses pembelajaran dilakukan secara terawasi. Input akan bersaing untuk dapat masuk ke dalam suatu kelas. Hasil dari lapisan kompetitif ini berupa kelas, yang kemudian akan dihubungkan dengan lapisan output oleh fungsi aktivasi. Fungsi aktivasi yang digunakan adalah fungsi linear dengan tujuan kelas yang diperoleh pada lapisan output sesuai dengan kelas yang dimasukkan ke lapisan output. Fungsi aktivasi yang digunakan dalam LVQ adalah fungsi aktivasi linear, tujuannya agar diperoleh keluaran yang sama dengan masukan, sesuai dengan rumus linear yaitu $x=y$.

Sebagian besar jaringan syaraf tiruan melakukan penyesuaian bobot-bobotnya selama menjalani pelatihan. Gambar dibawah ini merupakan model dari jaringan syaraf tiruan.



Gambar 2.8 Model Jaringan Syaraf Tiruan(Jumarwanto et al., 2009)

Jaringan syaraf tiruan memiliki kelebihan antara lain :

1. Belajar *Adaptive*

Merupakan kemampuan untuk mempelajari bagaimana melakukan pekerjaan berdasarkan data yang diberikan untuk pelatihan atau pengalaman awal.

2. *Self-Organization*

Kemampuan untuk membuat organisasi sendiri atau representasi dan informasi yang diterimanya selama waktu belajar.

3. *Real-Time Operation*

Perhitungan pada jaringan syaraf tiruan dapat dilakukan secara parallel, sehingga perangkat keras yang dirancang dan di produksi secara khusus dapat mengambil keuntungan dari kemampuan ini.

Jaringan syaraf tiruan juga memiliki kekurangan antara lain :

1. Tidak efisien dalam perhitungan angka numerik dengan tingkat presisi yang tinggi.
2. Tidak efisien jika digunakan untuk melakukan operasi algoritma aritmetrik, operasi logika, dan simbolis.

2.5 Learning Vector Quantization (LVQ)

Learning vector Quantization merupakan sebuah metode jaringan syaraf tiruan untuk mengklasifikasi yang banyak digunakan. Metode ini merupakan sebuah generalisasi dari kinerja otak manusia atau sistem saraf yang berdasarkan atas asumsi pemrosesan yang terjadi pada elemen yang disebut neuron. Sinyal mengalir diantara sel saraf melalui suatu hubungan dimana setiap sambungan memiliki bobot yang bersesuaian. Bobot tersebut akan digunakan untuk menggandakan sinyal yang dikirim melalui sinyal. Setiap sel saraf akan menerapkan fungsi aktivasi terhadap sinyal hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan sinyal keluarannya.

Learning Vector Quantization termasuk jenis Neural Network yang dikembangkan oleh Teuvo Kohonen tahun 1989. LVQ merupakan algoritma klasifikasi *prototype supervised* dengan aturan *Competitive Learning* versi dari algoritma Kohonen *Self-Organizing Map* (SOM). Model pembelajaran LVQ dilatih secara signifikan agar lebih cepat dibandingkan algoritma lain seperti *Backpropagation Neural Network*. (Arifianto, Sarosa, & Setyawati, 2014)

LVQ adalah suatu metode yang dilakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar

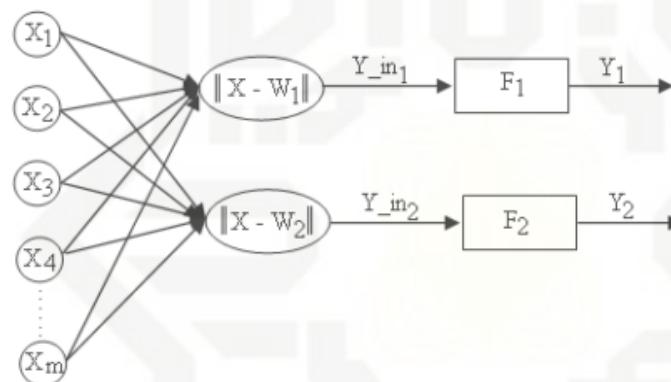
- Hak Cipta Dilindungi Undang-Undang**
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak mengikuti 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.

untuk mengklasifikasikan vector-vektor input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vector-vektor input. Jika 2 vektor input mendekati sama, maka lapisan kompetitif akan meletakkan ke 2 vektor tersebut pada kela yang sama. (Kusumadewi, 2003)

2.5.1 Arsitektur Jaringan LVQ

LVQ merupakan jaringan syaraf tiruan dengan tipe arsitektur jaringan lapisan tunggal umpan-maju yang terdiri dari unit masukan.

Berikut adalah gambar dari arsitektur jaringan LVQ.



Gambar 2.9 Arsitektur Jaringan LVQ (Kusumadewi, 2003)

Proses pembelajaran pada LVQ ini bertujuan untuk mencari nilai bobot yang sesuai untuk mengelompokkan vektor-vektor inputan kedalam kelas yang sesuai dengan inialisasi bobot awal pada saat pembentukan jaringan. Berikut ini adalah parameter yang akan digunakan pada LVQ.(Hidayati& Warsito, n.d.)

1. Alfa (*Learning Rate*)

Alfa didefinisikan sebagai tingkat pembelajaran. Jika alfa terlalu besar, maka algoritma akan menjadi tidak stabil sebaliknya jika alfa terlalu kecil, maka prosesnya akan terlalu lama. Nilai Alfa adalah $0 < \alpha < 1$

2. DecAlfa (Penurunan *Learning Rate*)

Yaitu penurunan tingkat pembelajaran.

3. MinAlfa (Minimum *Learning Rate*)

Yaitu minimal tingkat pembelajaran yang masih diperbolehkan.

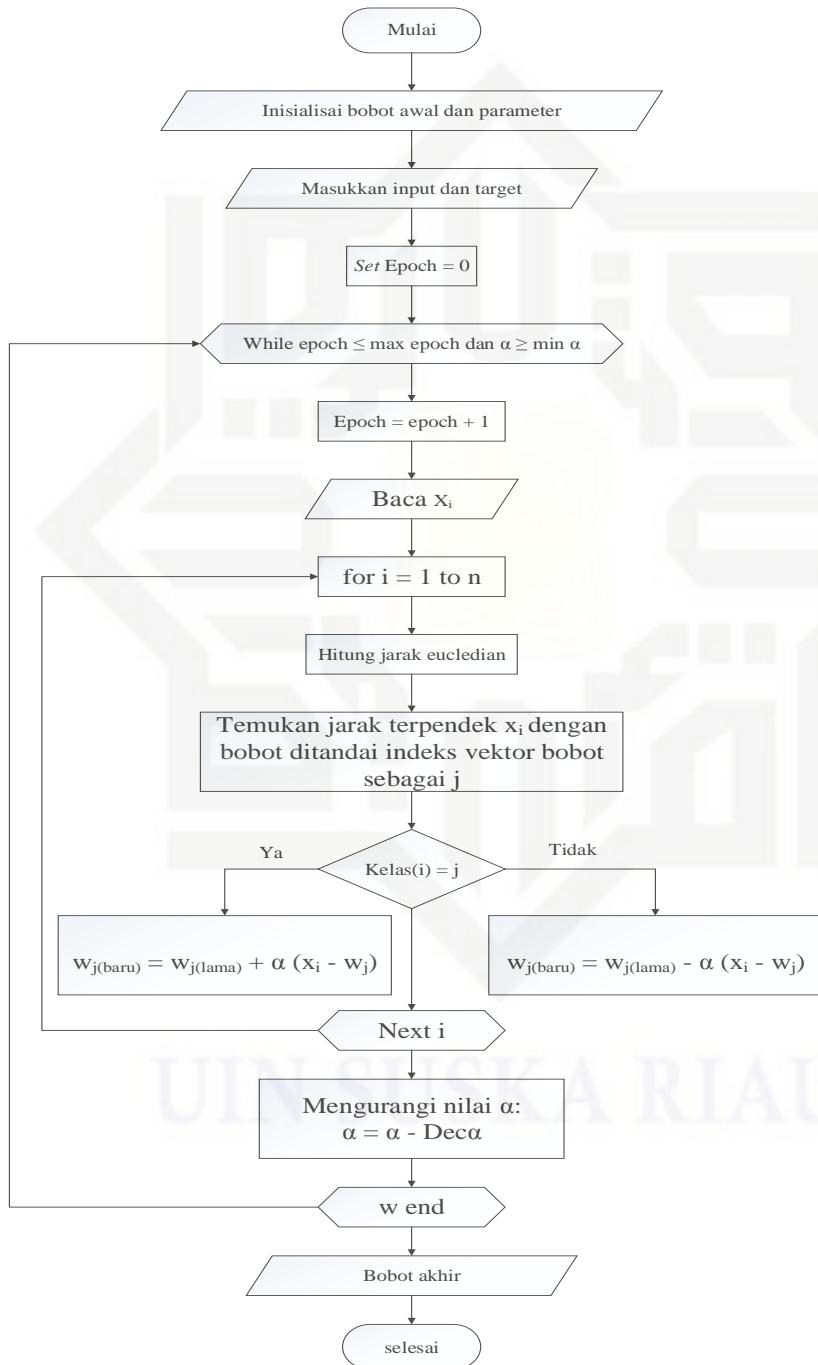
4. MaxEpoch (Maksimum Epoch)

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 mengikuti 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.

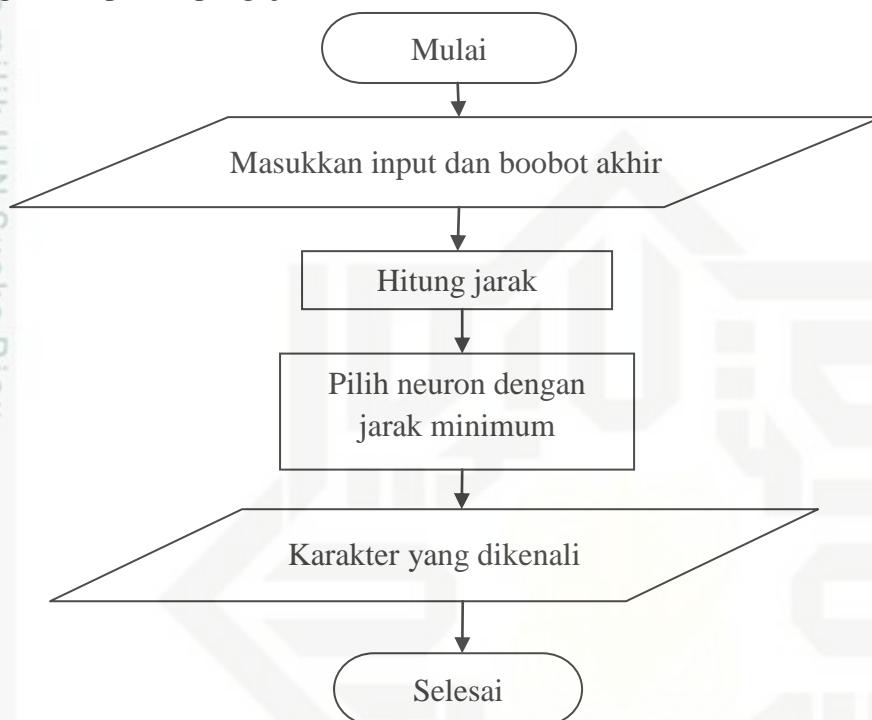
Yaitu jumlah epoch atau iterasi maksimum yang boleh dilakukan selama pelatihan. Iterasi akan dihentikan jika nilai epoch melebihi nilai epoch maksimum.

Proses pembelajaran dari metode LVQ adalah sebagai berikut :



Gambar 2.10 Algoritma Pembelajaran Metode LVQ(Hidayati & Warsito, n.d.)

Setelah dilakukan pelatihan , akan diperoleh bobot-bobot akhir (w). bobot ini selanjutnya akan digunakan untuk simulasi pengujian. Berikut ini merupakan algoritma proses pengujian dari metode LVQ.C



Gambar 2.11algoritma proses pengujian pada LVQ(Hidayati & Warsito, n.d.)

2.5.2 Perkembangan Metode LVQ

Metode Learning vector Quantization ini mengalami beberapa perkembangan mulai dari LVQ, LVQ 2, LVQ 2.1 dan LVQ 3. Ciri-ciri dari metode LVQ 1 adalah hanya vector referensi terdekat dengan vektor masukan yang diperbaharui. Arah perpindahan vector beragantung pada vektor referensi memiliki kelas yang sama dengan vektor masukan. Algoritma LVQ yang telah ditingkatkan, vektor pemenang dan vektor runner up akan sama-sama belajar bila kondisi tertentu terpenuhi. Idenya adalah bila jarak antara vektor masukan dengan vektor pemenang dan vektor runner up kira-kira mempunyai jarak yang sama. (Fausset, 1994)

2.5.3 Learning Vector Quantization 1 (LVQ 1)

Langkah-langkah pelatihan pada metode LVQ 1 (Kusumadewi dan Hartati, 2010) terdiri atas :

Setelah dilakukan pelatihan terhadap metode ini, akan diperoleh bobot akhir (W). Bobot ini nantinya akan digunakan untuk melakukan simulasi atau pengujian.

2.5.4 Learning Vector Quantization 2 (LVQ 2)

LVQ 2 merupakan sebuah algoritma hasil pengembangan dari algoritma LVQ awal (LVQ 1) (Budianita, 2013), berikut ini adalah kondisi dimana kedua vektor akan diperbaharui jika :

1. Unit pemenang dari runner up (vektor terdekat kedua) mempresentasikan kelas yang berbeda.
 2. Vektor masukan mempunyai kelas yang sama dengan runner up.
 3. Jarak antara vektor masukan ke pemenang dan jarak antara masukan ke runner up kira-kira sama.

Kondisi ini diperlihatkan didalam notasi berikut :

X vektor masukan saat ini

Yc vektor referensi terdekat dengan X

Hak Cipta Dilindungi Undang-Undang
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

Yr Vektor referensi terdekat berikutnya dengan X (*runner up*)

Dc jarak dari X ke Yc

Dr jarak dari X ke Yr

Vektor referensi dapat diperbarui jika masuk kedalam daerah yang disebut *window* (ϵ). *Window* yang digunakan untuk memperbarui vektor referensi didefinisikan sebagai berikut :

Vektor X akan masuk ke dalam window bila

$$\frac{dc}{dr} > 1 - \epsilon \quad \frac{dr}{dc} < +\epsilon \quad \dots \dots \dots \quad (2.15)$$

Dengan nilai ϵ tergantung pada jumlah data pelatihan. Berdasarkan kohonen(1990) dalam Fausett (1994) nilai $\epsilon = 0.3$ adalah nilai yang disarankan. Vektor Yc dan Yr akan diperbarui jika kondisi 1, 2 dan 3 terpenuhi. Vektor Yc dan Yr diperbarui menggunakan persamaan persamaan 2.13 untuk target bobot yang sama dengan kelas target hasil dan 2.14 untuk target bobot yang tidak sama dengan kelas target hasil.

2.5.5 Learning Vector Quantization (LVQ 2.1)

Algoritma LVQ 2.1 ini memodifikasi algoritma LVQ 1 dengan cara mempertimbangkan dua vektor referensi terdekat, yaitu Yc1 dan Yc2. Kondisi untuk memperbarui kedua vektor tersebut adalah apabila salah satu dari vektor tersebut masuk kedalam kelas yang sama dengan vektor x, sementara vektor yang lainnya tidak masuk kedalam kelas yang sama dengan vektor masukan x. Sama halnya dengan LVQ 2, vektor x harus masuk kedalam *window* agar dapat terjadi pembaharuan.

Window didefinisikan sebagai berikut :

$$\text{Min} \left[\frac{d_{c1}}{d_{c2}}, \frac{d_{c2}}{d_{c1}} \right] > 1 - \epsilon$$

Dan

$$\text{Max} \left[\frac{d_{c1}}{d_{c2}}, \frac{d_{c2}}{d_{c1}} \right] < 1 + \epsilon \quad \dots \dots \dots \quad (2.16)$$

Berikut ini merupakan algoritma untuk melakukan pelatihan menggunakan LVQ 2.1.



1. Tetapkan nilai masukan X dan nilai bobot awal (W), parameter pelatihan learning rate (alfa), minimum learning rate (MinAlfa), window (ϵ), dan epoch.
2. Lakukan perhitungan jarak antara masukan X dan semua bobot awal (W).
3. Dapatkan nilai pemenang dan nilai runner up dari perhitungan jarak.
4. Periksa apakah target nilai bobot pemenang sama dengan kelas target masukan.
5. Jika kelas bobot pemenang sama dengan target masukan X. Perbaharui bobot pemenang menggunakan persamaan rumus (2.13).
6. Jika kelas bobot pemenang tidak sama dengan kelas target masukan X. Periksa kelas bobot runner up, jika kelas runner up sama dengan kelas masukan X.
7. Lakukan perbandingan nilai pemenang dan runner up kedalam nilai window menggunakan persamaan rumus (2.16)
8. Jika hasilnya true lakukan pembaharuan bobot untuk bobot runner up menggunakan persamaan rumus (2.13) dan perbaharui kelas pemenang menggunakan rumus (2.14)
9. Jika kelas runner up juga tidak sama dengan kelas masukan X.
10. Lakukan pembaharuan bobot pemenang menggunakan persamaan rumus (2.14).
11. Perbaharui nilai alfa.

Berikut ini merupakan contoh perhitungan menggunakan jaringan syaraf tiruan LVQ 2 untuk mencari vector bobot masing-masing kelas dengan asumsi terdapat 2 kelas yang berbeda (kelas 1 dan kelas 2) dengan data pelatihan ditunjukkan dengan tabel 2.1

Tabel 2.1 Contoh data pelatihan berupa vector dengan ukuran 1x4

(x1,x2,x3,x4)	T
1 1 0 0	1
0 0 0 1	2
0 0 1 1	2
1 0 0 0	1
0 1 1 0	2

Pelatihan



Ambil 2 dua vector masukan pertama sebagai inisialisasi bobot awal masing-masing kelas.

Bobot kelas 1

$$W_{(1,1)} = 1$$

$$W_{(1,2)} = 1$$

$$W_{(1,3)} = 0$$

$$W_{(1,4)} = 0$$

Learning rate = 0,1

Window = 0.3

dengan nilai α akan dikurangi sebesar 0.1 α pada akhir epoch saat ini.

Bobot kelas ke 2

$$W_{(1,1)} = 0$$

$$W_{(1,2)} = 0$$

$$W_{(1,3)} = 0$$

$$W_{(1,4)} = 1$$

Epoch ke-1

1. Vector masukan 0 0 1 1, dengan target kelas 2.
2. hitung jarak menggunakan persamaan rumus 2.12

$$D1 = \sqrt{(0-1)^2 + (0-1)^2 + (1-0)^2 + (1-0)^2} = 2$$

$$D2 = \sqrt{(0-0)^2 + (0-0)^2 + (1-0)^2 + (1-1)^2} = 1$$

Didapatkan bobot pemenang pada bobot ke 2, sehingga kelas bobot pemenang sama dengan kelas bobot masukan. Lakukan langkah ke 5 pada algoritma yaitu pembaharuan bobot menggunakan persamaan rumus (2.13)

$$W_{2(1,1)} = 0 + 0.1(0 - 0) = 0$$

$$W_{2(1,2)} = 0 + 0.1(0 - 0) = 0$$

$$W_{2(1,3)} = 0 + 0.1(1 - 0) = 0.1$$

$$W_{2(1,4)} = 1 + 0.1(1 - 1) = 1$$

3. Vektor masukan 1 0 0 0
4. Lakukan perhitungan jarak menggunakan rumus 2.12

$$D1 = \sqrt{(1-1)^2 + (0-1)^2 + (0-0)^2 + (0-0)^2} = 1$$

$$D2 = \sqrt{(1-0)^2 + (0-0)^2 + (0-0.1)^2 + (0-1)^2} = 1.41$$

Didapatkan bobot pemenang pada bobot ke 1, sehingga kelas bobot pemenang sama dengan kelas bobot masukan. Lakukan langkah ke 5 pada algoritma yaitu pembaharuan bobot menggunakan persamaan rumus (2.13)



$$\begin{aligned}W1_{(1,1)} &= 1 + 0.1(1 - 1) = 1 \\W1_{(1,2)} &= 1 + 0.1(0 - 1) = 0.9 \\W1_{(1,3)} &= 0 + 0.1(0 - 0) = 0 \\W1_{(1,4)} &= 0 + 0.1(0 - 0) = 0\end{aligned}$$

5. Vektor masukan 0 1 1 0
6. Lakukan perhitungan jarak menggunakan rumus 2.12

$$D1 = \sqrt{(0 - 1)^2 + (1 - 0.9)^2 + (1 - 0)^2 + (0 - 0)^2} = 1.41$$

$$D2 = \sqrt{(0 - 0)^2 + (1 - 0)^2 + (1 - 0.1)^2 + (0 - 1)^2} = 1.79$$

7. Vektor bobot pemenang terdapat pada bobot 1. Sehingga target bobot pemenang tidak sama dengan target masukan x.
8. Periksa vektor runner up, jika target runner up sama dengan target masukan lakukan perbandingan nilai window menggunakan rumus 2.16

$$\left(\frac{1.41}{1.79}, \frac{1.79}{1.41}\right) > 1 - 0.3 \text{ AND } \left(\frac{1.41}{1.79}, \frac{1.79}{1.41}\right) < 1 + 0.3$$

Hasil dari perbandingan window diatas bernilai True dan True, sehingga lakukan langkah 8 pada algoritma yaitu pembaharuan kedua bobot menggunakan rumus 2.13 untuk bobot runner up dan rumus 2.14 untuk bobot pemenang.

Bobot 2

$$\begin{aligned}W2_{(1,1)} &= 0 + 0.1(0 - 0) = 0 \\W2_{(1,2)} &= 0 + 0.1(1 - 0) = 0.1 \\W2_{(1,3)} &= 0.1 + 0.1(1 - 0.1) = 0.19 \\W2_{(1,4)} &= 1 + 0.1(0 - 1) = 0.9\end{aligned}$$

Bobot 1

$$\begin{aligned}W1_{(1,1)} &= 1 - 0.1(0 - 1) = 1.1 \\W1_{(1,2)} &= 0.9 - 0.1(0 - 0.9) = 0.99 \\W1_{(1,3)} &= 0 - 0.1(0 - 0) = 0 \\W1_{(1,4)} &= 0 - 0.1(0 - 0) = 0\end{aligned}$$

9. Kurangi nilai α

$$\alpha(\text{baru}) = \alpha(\text{lama}) - 0.1 * \alpha(\text{lama}) = 0.1 - (0.1 * 0.1) = 0.09$$

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 mengikuti 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.

10. Tes kondisi berhenti

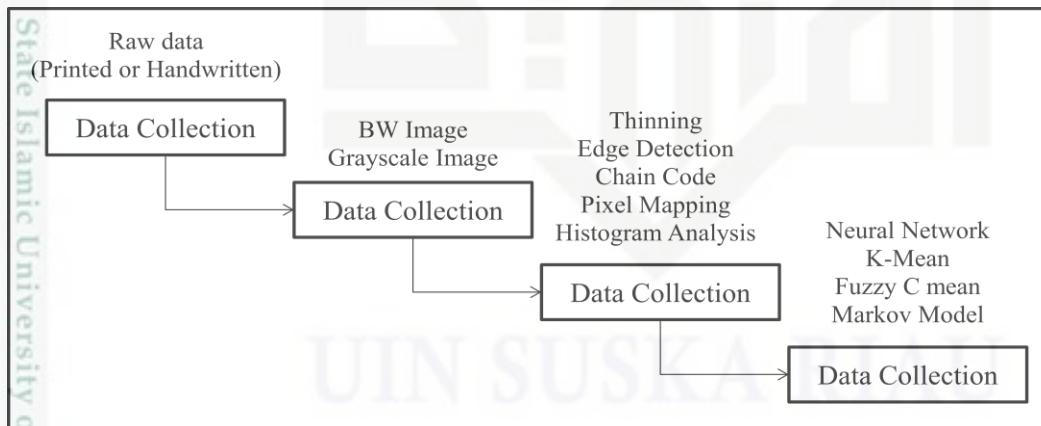
Yaitu jika jumlah epoh sudah melebihi jumlah maksimum epoh yang diperbolehkan, atau nilai α lebih kecil dari nilai maksimum dari nilai eror.

Berikut ini merupakan algoritma untuk pengujian pada LVQ 2.1

1. Hitung jarak antara vektor masukan X dengan bobot yang telah dilatih dengan data latih. Cara perhitungan sama dengan perhitungan pada pelatihan
2. Tentukan jarak pemenang.
3. Jika target bobot pemenang sama dengan target vektor masukan, maka hasil akurasi bernilai benar, jika target bobot pemenang tidak sama dengan target vektor masukan maka hasil akurasi salah.

2.6 Pengenalan Pola Karakter

Hampir semua tipe pengenalan pola karakter menggunakan fase yang sama dalam mengenal suatu karakter. Umumnya, prosedur pengenalan karakter ini digunakan oleh beberapa peneliti (Handayani dkk, 2014).



Gambar 2.12 Fase Pengenalan Karakter (Handayani dkk, 2014)

Pertama, dibutuhkan data mentah yang banyak atau koleksi data (*collection data*). Selanjutnya fase *pre-processing* adalah pengolahan citra (*image processing*). Sebagai contoh, konversi citra ke tipe *grayscale* setelah itu dikonversi ke tipe biner. Desain dari fase *pre-processing* harus dilakukan dengan baik. Fase *processing* termasuk didalamnya *thinning*, *edge detection*, *chain mode*,



pixel mapping, dan *histogram analysis* merupakan bagian dari *processing*. Bagian ini pada dasarnya proses perubahan data mentah menjadi komponen yang dapat siap diproses pada proses pelatihan (*training process*). Akhirnya, pelatihan (*training*) dan pengenalan (*recognition*) data akan diproses pada bagian *classification*. Sistem akan melatih data masuk, jadi untuk selanjutnya dapat lebih mudah mengenali suatu inputan data (Handayani dkk, 2014).

2.7 Huruf Hijaiyah

Huruf hijaiyah merupakan huruf-huruf yang digunakan dalam pembentukan kata-kata dalam alquran. Jumlah keseluruhan huruf hijaiyah berjumlah 29 huruf. Ada beberapa pola huruf hijaiyah ini yang mempunyai bentuk sama, namun terdapat perbedaan pada letak titik dan jumlah titik pada huruf tersebut. Sehingga jika tidak teliti dalam mengenali huruf hijaiyah bisa saja akan terjadi kesalahan dalam membacanya.

Pola pada huruf hijaiyah ini terbagi menjadi 4 bagian, yaitu :

- 3 Pola huruf hijaiyah terlepas
- 4 Pola huruf hijaiyah terletak di awal
- 5 Pola huruf hijaiyah terletak di tengah
- 6 Pola huruf hijaiyah terletak di akhir

Berikut ini adalah table huruf hijaiyah serta bacaannya.

Table 2.2 Huruf Hijaiyah (Handayani, Iskandar, & Adrian, 2014)

No	Huruf lepas	Huruf di awal	Huruf di tengah	Huruf di akhir	Lafazh huruf
1	—	—	—	—	Alif
2	ج	ج	ج	ج	Ba
3	ج	ج	ج	ج	Ta
4	س	س	س	س	Tsa
5	ه	ه	ه	ه	Jim
6	ه	ه	ه	ه	Ha
7	خ	خ	خ	خ	Kha



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 mengikuti 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.

8	د	د	Dal
9	ذ	ذ	Dzal
10	ر	ر	Ra
11	ز	ز	Zal
12	س	س	Sin
13	ش	ش	Syin
14	ص	ص	Shad
15	ض	ض	Dhad
16	ط	ط	Tha
17	ظ	ظ	Zha
18	ع	ع	Ain
19	غ	غ	Ghain
20	ف	ف	Fa
21	ق	ق	Qaf
22	ك	ك	Kaf
23	ل	ل	Lam
24	م	م	Mim
25	ن	ن	Nun
26	و	و	Wau
27	ه	ه	Ha
28	هـ	هـ	Hamzah
29	يـ	يـ	Ya

2.8 Penelitian Terkait

Table 2.3 Penelitian Terkait

No	Peneliti dan Tahun	Judul	Metode	Keterangan
1	KeumalaAngraini, ST.	Penerapan Metode <i>Learning Vector Quantization</i> Untuk	<i>LVQ</i>	Hasil dari penelitian ini adalah sebagai



© Hak cipta milik 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 mengikuti 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	EkoSriwahyono dan Ernastuti (2012)	Pengenalan Karakter Huruf Hijaiyah		berikut untuk hijaiyah tunggal sebesar 76,43%, hijaiyah awal 65,45%, hijaiyah tengah 62,73%, dan huruf hijaiyah akhir 80%.
3	Nurul Hidayati dan Budi Warsito	Identifikasi Nomor Polisi Mobil Menggunakan Metode JST <i>Learning Vector Quantization</i>	<i>Learning Vector Quantization</i>	Hasil percobaan menunjukkan keberhasilan sebesar 78% dari pola karakter yang dikenal
4	RischanMafrur, dkk.	Prediksi Terjangkitnya Penyakit Jantung Dengan Metode Learning Vector Quantization	LVQ	hasil jaringan mampu mengenali pola dengan benar sebesar 66,79
5	Grace Levina, HendrawanArmento	Pengenalan Huruf Jawa Menggunakan Metode <i>Learning Vector Quantization</i> (LVQ)	LVQ	Percentase ketepatan dari penelitian ini sebesar 56,61 dari 30 data yang diujikan.
		Analisa Berbagai Jenis Huruf Komputer Menggunakan Algoritma <i>Chain Code</i> Dalam Bentuk <i>Run</i>	<i>Chain code</i>	Hasil dari pengujian system menunjukkan nilai akurasi

<p>© Hak cipta milik UIN Suska Riau</p> <p>Hak Cipta Dilindungi Undang-Undang</p> <p>1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:</p> <ol style="list-style-type: none"> Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau. 	<p><i>Length Encoding</i></p>	<p>Jenis huruf Adobe Caslon Pro Bold dan Cooper Black memiliki tingkat keakuratan 100% dalam perhitungan chain code dan RLE nya sedangkan Times New Roman memiliki tingkat keakuratan 53,8%</p>
<p>6</p> <p>State Islamic University of Sultan Syarif Kasim Riau</p>	<p>Elvia Budianita, Muhammad Firdaus</p> <p>Diagnosis Penyakit Kejiwaan Menggunakan Jaringan Syaraf Tiruan Learning Vector Quantization2 (LVQ2) (Studi Kasus : RSJ Tampak Pekanbaru)</p>	<p>LVQ2</p> <p>Pada penelitian yang telah dilakukan berdasarkan hasil pengujian parameter dengan nilai learning rate = 0,1 minimal learning rate = 0,01 dan nilai window = 0,4 pada tabel yang telah diujikan di peroleh nilai akurasi sebesar 90%.</p>