

PREDIKSI TERJANGKITNYA PENYAKIT JANTUNG DENGAN METODE LEARNING VECTOR QUANTIZATION

Nurul Hidayati¹, Budi Warsito²

¹Alumni Program Studi Statistika FMIPA UNDIP

²Staf Pengajar Program Studi Statistika FMIPA UNDIP

Abstract

Learning Vector Quantization (LVQ) is a method that train the competitives layer with supervised. The competitives layer will learn automatically to classify the input vector given. If some input vectors has the short distance then the input vector will be grouped into the same class. The LVQ method can be used to classify the data into some classes or categories. At this paper, the LVQ method will be applied to classify if someone is suffer potencie of heart desease or not. The data that be trained are 268 data of heart desease patient from UCI (*University of California at Irvine*) with 10 variables that are factors influence that infected of heart desease. From some trials showed that the *learning rate* (α) = 0.25, decrease of *learning rate* ($Dec\alpha$) = 0.1, and the minimum *learning rate* ($Min\alpha$) = 0.001 are values that give a good prediction with level of accuracy is about 66.79 %.

Keywords: Learning Vector Quantization, Classify, Heart Desease

1. Pendahuluan

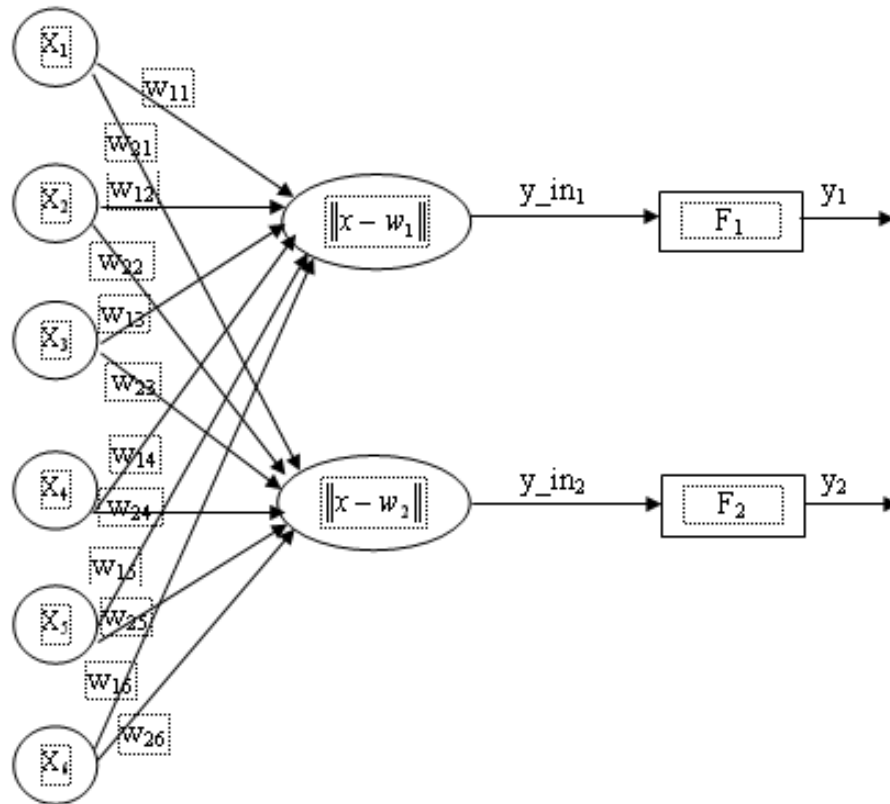
Learning Vector Quantization (LVQ) merupakan jaringan lapisan tunggal (*single-layer net*) di mana lapisan masukan terkoneksi secara langsung dengan setiap neuron pada keluaran. Koneksi antar neuron tersebut dihubungkan dengan bobot/*weight*. Neuron-neuron keluaran pada LVQ menyatakan suatu kelas atau kategori tertentu^[2]. Bobot merupakan nilai matematis dari koneksi yang mentransfer data dari satu lapisan ke lapisan lainnya, yang berfungsi untuk mengatur jaringan sehingga dapat menghasilkan output yang diinginkan. Bobot pada LVQ sangat penting, karena dengan bobot ini input dapat melakukan pembelajaran dalam mengenali suatu pola. Vektor bobot berfungsi untuk menghubungkan setiap neuron pada lapisan input dengan masing-masing neuron pada lapisan output. Vektor bobot biasanya dituliskan dengan $\mathbf{w}_{tj} = (w_{t1}, w_{t2}, w_{t3}, \dots, w_{tm})$ dimana t menunjukkan kelas yang nilainya antara 1 sampai K, dengan K adalah banyaknya kelas pada lapisan output, sedangkan m adalah banyaknya variabel yang digunakan. Pada tulisan dibahas algoritma LVQ dan penerapannya untuk memprediksi terjangkitnya penyakit jantung pada seseorang.

2. Arsitektur Jaringan

Arsitektur LVQ terdiri dari lapisan input (*input layer*), lapisan kompetitif (terjadi kompetisi pada input untuk masuk ke dalam suatu kelas berdasarkan kedekatan jaraknya) dan lapisan output (*output layer*). Lapisan input dihubungkan dengan lapisan kompetitif oleh bobot. Dalam lapisan kompetitif, proses pembelajaran dilakukan secara terawasi^[3]. 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.

Misal ada enam variabel dari vektor input, yaitu $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$ dengan neuron keluaran y_1 dan y_2 , serta dua vektor bobot yaitu $\mathbf{w}_{1j} = (w_{11}, w_{12}, w_{13}, w_{14}, w_{15}, w_{16})$ dan $\mathbf{w}_{2j} = (w_{21}, w_{22}, w_{23}, w_{24}, w_{25}, w_{26})$, maka arsitektur jaringan LVQ dapat disajikan pada Gambar 1.



Gambar 1. Arsitektur Jaringan Learning Vector Quantization (LVQ) dengan 6 Unit Input dan Dua Vektor Bobot

Berdasarkan Gambar 1, tampak bahwa dalam LVQ terdapat dua vektor bobot yang menghubungkan setiap neuron masukan dengan neuron keluaran sehingga dapat dikatakan bahwa setiap neuron keluaran pada LVQ berhubungan dengan sebuah vektor bobot. Untuk melakukan proses pengenalan dan pembelajaran, LVQ menggunakan operasi-operasi vektor. Pola-pola akan disajikan dalam bentuk vektor. Pemrosesan yang terjadi pada setiap neuron adalah mencari jarak antara suatu vektor input ke bobot yang bersangkutan (\mathbf{w}_1 dan \mathbf{w}_2). Dalam hal ini \mathbf{w}_1 adalah vektor bobot yang menghubungkan setiap neuron pada lapisan input ke neuron pertama pada lapisan output, sedangkan \mathbf{w}_2 adalah vektor bobot yang menghubungkan setiap neuron pada lapisan input ke neuron kedua pada lapisan output. Fungsi aktivasi (F) yang digunakan pada arsitektur jaringan LVQ adalah fungsi linier. Tujuannya adalah agar diperoleh keluaran yang sama dengan masukan, sesuai dengan rumus fungsi linier

yaitu $y = x$. Fungsi aktivasi F_1 akan memetakan y_{in_1} ke $y_1 = 1$ apabila $|x - w_1| < |x - w_2|$, dan $y_1 = 0$ jika sebaliknya. Demikian pula fungsi aktivasi F_2 , akan memetakan y_{in_2} ke $y_2 = 1$ apabila $|x - w_2| < |x - w_1|$, dan $y_2 = 0$ jika sebaliknya^[2].

Metode LVQ akan melakukan pengenalan terlebih dahulu terhadap pola masukan yang harus disajikan dalam bentuk vektor agar dapat dicari kelasnya. Karena setiap neuron keluaran menyatakan kelas atau kategori tertentu, maka pola masukan dapat dikenali kelasnya berdasarkan neuron keluaran yang diperoleh. Metode LVQ mengenali pola masukan berdasarkan pada kedekatan jarak antara dua vektor yaitu vektor dari unit/neuron masukan dengan vektor bobot. Pengenalan dalam metode LVQ ini terdapat dua proses, yaitu proses pembelajaran dan proses pengujian.

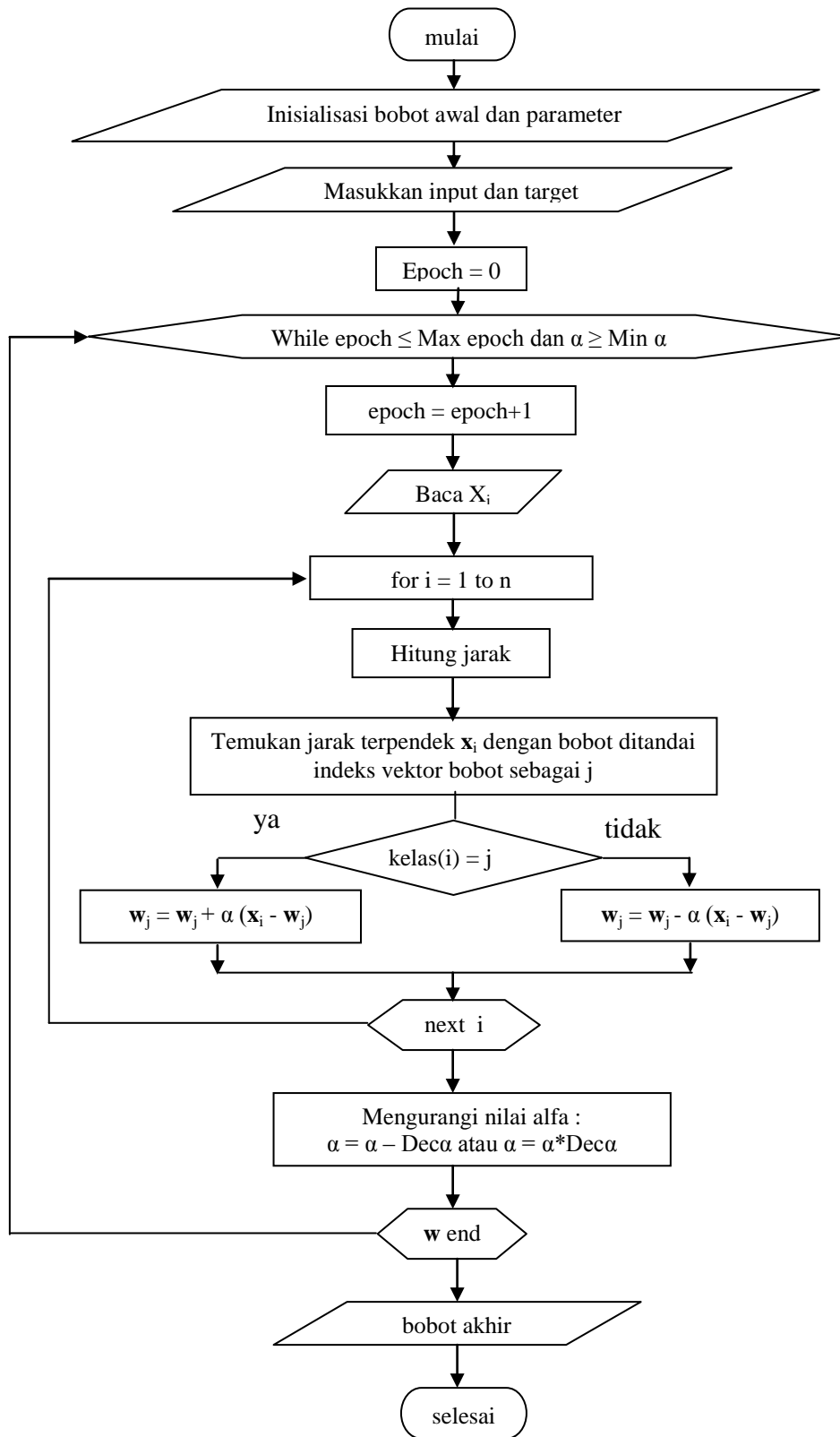
3. Proses Pembelajaran dan Pengujian

Pada proses awal pengenalan, vektor input akan mengalami proses pembelajaran yang dilakukan melalui beberapa epoch sampai batas epoch maksimal tercapai. LVQ melakukan pembelajaran pada lapisan kompetitif yang terawasi. Suatu lapisan kompetitif akan secara otomatis belajar untuk mengklasifikasikan vektor-vektor input. Kelas-kelas yang didapatkan sebagai hasil dari lapisan kompetitif ini hanya tergantung pada jarak antara vektor input dengan vektor bobot dari masing-masing kelas dan vektor input akan masuk ke dalam kelas yang memiliki jarak terdekat. Algoritma pembelajaran pada LVQ bertujuan mencari nilai bobot yang sesuai untuk mengelompokkan vektor-vektor input ke dalam kelas yang sesuai dengan yang telah diinisialisasi pada saat pembentukan jaringan LVQ.

Parameter-parameter yang digunakan pada metode LVQ ini adalah sebagai berikut:

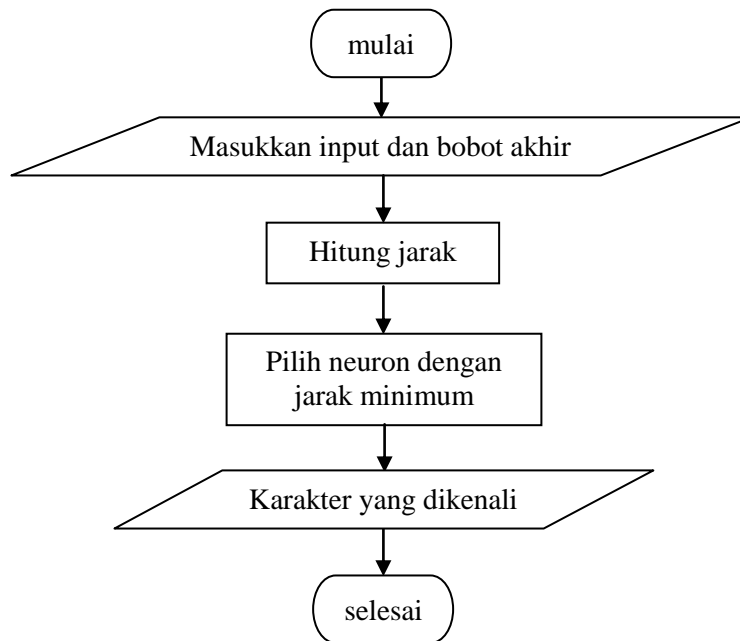
1. Alfa (*Learning rate*)
Alfa didefinisikan sebagai tingkat pembelajaran. Jika alfa terlalu besar, maka algoritma akan menjadi tidak stabil sebaliknya jika alfa terlalu kecil, maka prosesnya akan terlalu lama. Nilai alfa adalah $0 < \alpha < 1$.
2. DecAlfa (Penurunan *Learning rate*)
Yaitu penurunan tingkat pembelajaran.
3. MinAlfa (Minimum *Learning rate*)
Yaitu minimal nilai tingkat pembelajaran yang masih diperbolehkan.
4. MaxEpoch (Maksimum epoch)
Yaitu jumlah epoch atau iterasi maksimum yang boleh dilakukan selama pelatihan. Iterasi akan dihentikan jika nilai epoch melebihi epoch maksimum.

Misalkan terdapat n buah data dengan m buah variabel input. Data-data tersebut terbagi dalam K kelas. Motivasi dari algoritma LVQ adalah menemukan unit output yang paling dekat dengan vektor input^[1]. Algoritma proses pembelajaran pada metode LVQ disajikan pada Gambar 2.



Gambar 2. Algoritma Proses Pembelajaran pada Metode Learning Vector Quantization (LVQ)

Setelah dilakukan pelatihan, akan diperoleh bobot-bobot akhir (W). Bobot-bobot ini selanjutnya digunakan untuk melakukan simulasi atau pengujian. Misalkan akan diuji p buah data. Algoritma proses pengujian pada metode LVQ disajikan pada Gambar 3.



Gambar 3. Algoritma Proses Pengujian pada Metode Learning Vector Quantization (LVQ)

Penerapan LVQ untuk Prediksi Terjangkitnya Penyakit Jantung Berdasarkan Faktor-faktor yang Mempengaruhinya

Pada bagian ini dibahas prediksi penyakit jantung menggunakan LVQ. Untuk dapat mengetahui bahwa seseorang menderita penyakit jantung dibutuhkan pengetahuan khusus untuk mendeteksi dan melakukan diagnosis awal dari pemeriksaan fisik yang merupakan syarat bila terdapat keluhan atau gejala yang berhubungan dengan jantung. Menurut berbagai sumber^[4], terdapat 10 faktor yang mempengaruhi seseorang berpotensi terjangkit penyakit jantung yaitu :

1. Usia
2. Jenis kelamin
Agar dapat dikenali oleh jaringan, data pada variabel jenis kelamin harus diubah ke dalam bentuk numerik, yaitu :
 - Diberi nilai "0" jika jenis kelamin adalah perempuan
 - Diberi nilai "1" jika jenis kelamin adalah laki-laki
3. Angina (nyeri dada)
Agar dapat dikenali oleh jaringan, data pada variabel angina harus diubah ke dalam bentuk numerik, yaitu :
 - Diberi nilai "1" jika jenis angina adalah *angina pectoris*
 - Diberi nilai "2" jika jenis angina adalah *unstable angina*
 - Diberi nilai "3" jika jenis angina adalah *variant pectoris*

- Diberi nilai "4" jika jenis angina adalah *myocardiac infarction*
- 4. Tekanan darah saat beristirahat
- 5. Kolesterol
- 6. Kadar gula darah > 120 mg/dl
Agar dapat dikenali oleh jaringan, data pada variabel Kadar gula darah > 120 mg/dl harus diubah ke dalam bentuk numerik, yaitu :
 - Diberi nilai "0" jika kadar gula darah tidak > 120 mg/dl
 - Diberi nilai "1" jika kadar gula darah > 120 mg/dl
- 7. Denyut nadi maksimal
- 8. Merokok
Agar dapat dikenali oleh jaringan, data pada variabel merokok harus diubah ke dalam bentuk numerik, yaitu :
 - Diberi nilai "0" jika tidak merokok
 - Diberi nilai "1" jika merokok kurang dari 3 kali sehari
 - Diberi nilai "2" jika merokok lebih dari 4 kali sehari
- 9. Keturunan
Agar dapat dikenali oleh jaringan, data pada variabel keturunan harus diubah ke dalam bentuk numerik, yaitu :
 - Diberi nilai "0" jika tidak mempunyai sejarah keluarga yang sakit jantung
 - Diberi nilai "1" jika mempunyai sejarah keluarga yang sakit jantung
- 10. Olah raga
Pada variabel olah raga ini, yang dimaksud olah raga adalah olah raga jalan kaki dengan jarak yang ditempuh kurang lebih 4 km. Sedangkan yang dimaksud 0, 1, 2, 3, 4 dan 5 dalam data pasien penyakit jantung adalah frekuensi olah raga tiap minggunya

Pada metode LVQ, target/sasaran (Y) yang diinginkan juga harus dituliskan. Dalam hal ini targetnya berupa kategori terjangkit penyakit jantung dan tidak terjangkit penyakit jantung. Agar dapat dikenali oleh jaringan, kategori harus diubah ke dalam bentuk numerik, yaitu :

- Diberi nilai "1" jika orang tersebut tidak terjangkit penyakit jantung
- Diberi nilai "2" jika orang tersebut terjangkit penyakit jantung

Data yang digunakan dalam tulisan ini diambil dari situs internet : <http://www.ics.uci.edu/~mllearn/MLRepository.html> yang disediakan oleh UCI (*University Of California at Irvine*)^[5]. Jumlah data seluruhnya adalah 270. Data ke-1 dan 2 digunakan sebagai bobot awal yang mewakili dari masing-masing kelas atau dapat dilihat pada tabel 1. Sedangkan data yang akan dilatih dan selanjutnya akan diuji adalah data ke-3 sampai data ke-270 sebanyak 268 data.

Proses Pembelajaran pada Metode LVQ

Pada metode LVQ, bobot awal menggunakan pola-pola yang sudah ada. Kemudian bobot tersebut akan diubah (*di-update*) tergantung dari kelas vektor masukan sesuai dengan kelas yang dinyatakan sebagai neuron pemenang. Jika sesuai, maka vektor bobot *di-update* sehingga jaraknya semakin dekat dengan vektor masukan. Jika tidak, vektor bobot *di-update* sehingga jaraknya semakin jauh dengan vektor masukan tersebut. Bobot awal ini diambil dari data ke-1 dan 2, yang harus dibawa ke dalam bentuk vektor. Vektor bobot ini biasanya

ditulisikan dengan $w_{1j} = (w_{11}, w_{12}, w_{13}, \dots, w_{1m})$ yang merupakan vektor bobot kelas pertama dengan m variabel dan $w_{2j} = (w_{21}, w_{22}, w_{23}, \dots, w_{2m})$ yang merupakan vektor bobot kelas kedua dengan m variabel. Pada tulisan ini, data terbagi dalam dua kelas dengan 10 variabel. Bobot awal dalam tulisan ini ditunjukkan pada Tabel 1 berikut:

Tabel 1. Bobot Awal pada Proses Pembelajaran

Bobot ke-i	w _{i1}	w _{i2}	w _{i3}	w _{i4}	w _{i5}	w _{i6}	w _{i7}	w _{i8}	w _{i9}	w _{i10}
1	39	0	3	94	199	0	179	1	0	3
2	63	1	4	130	254	0	147	2	1	0

Karena harus berupa vektor, maka bentuk bobot awal pada tabel 1 adalah sebagai berikut :

$$w_{1j} = (39, 0, 3, 94, 199, 0, 179, 1, 0, 3)$$

$$w_{2j} = (63, 1, 4, 130, 254, 0, 147, 2, 1, 0)$$

Parameter – parameter yang digunakan pada tulisan ini adalah sebagai berikut:

1. Alfa (*Learning rate*), $\alpha = 0.25$
2. DecAlfa (Penurunan *Learning rate*), $Dec\alpha = 0.1$
3. MinAlfa (Minimum *Learning rate*), $Min\alpha = 0.001$
4. MaxEpoch (Maksimum epoch), $MaxEpoch = 100$

Fungsi secara lengkap disajikan pada Lampiran 1. Hasil output menunjukkan bahwa setelah dilakukan training sampai 100 epoch tingkat keberhasilan jaringan LVQ untuk dapat mengenali pola dengan benar sebesar 66.79%. Ditunjukkan pula data mana saja yang dapat dikenali dengan benar serta data yang tidak dapat dikenali.

Sekarang misalkan dipunyai data simulasi pasien baru seperti pada Tabel 2. Akan dilakukan pengecekan apakah yang bersangkutan berpotensi menderita penyakit jantung atau tidak.

Tabel 2. Data simulasi pasien baru dengan gejala penyakit jantung

Nomor	Variabel	Nilai
1	Usia	68
2	Jenis Kelamin	0
3	Angina	4
4	Tekanan darah saat beristirahat	121
5	Kolesterol	157
6	Kecepatan gula darah > 120 mg/dl	0
7	Denyut nadi maksimal	134
8	Merokok	2
9	Keturunan	0
10	Olah raga	4

Untuk melakukan checking digunakan fungsi pada Lampiran 2. Hasil perhitungan memberikan kesimpulan bahwa pasien dengan ciri-ciri tersebut positif menderita penyakit jantung.

4. Kesimpulan

Jaringan LVQ yang terdiri dari layer input, layer kompetitif dan layer output dapat digunakan untuk pengenalan pola/klasifikasi berdasarkan kedekatan jarak antar kelompok. Hasil uji coba dengan data penyakit jantung memberikan hasil, jaringan mampu mengenali pola dengan benar sebesar 66,79%.

DAFTAR PUSTAKA

1. Fausset, L.V., *Fundamentals of Neural Network: Arsitektur, Algorithm, and Application*, Prentice Hall, New Jersey, 1994.
2. Kusumadewi, S., *Membangun Jaringan Saraf Tiruan Menggunakan MATLAB dan Excel Link*, Graha Ilmu, Yogyakarta, 2004.
3. Ranadhi, D., Indarto, W., dan Hidayat, T., Implementasi Learning Vector Quantization (LVQ) untuk Pengenal Pola Sidik Jari pada Sistem Informasi Narapidana LP Wirogunan, *Media Informatika*, 2006, Vol. 4, No. 1: 51-65.
4. Soeharto, I., *Serangan Jantung dan Stroke: Hubungannya dengan Lemak dan Kolesterol*, Edisi ke-2, PT. Gramedia Pustaka, Jakarta, 2004.
5. _____, URL: <http://www.ics.uci.edu/~mllearn/MLRepository.html>.

Lampiran 1

Listing program Training Model LVQ

```
function [Y,Yc,j,b]=lvqjantung(data);
data=xlsread('datalvq');
P=data(1:270,2:11)';
Tc=data(1:270,12)';
T = ind2vec(Tc);
net = newlvq(minmax(P),2,[.556 .444]);
net.trainParam.epochs = 100;
net.trainParam.show = 10;
net = train(net,P,T);
Y = sim(net,P);
Yc = (vec2ind(Y))';
j=length(Tc);
for i=1:j,
    k(i)=0;
    if Tc(i)~=Yc(i),
        k(i)=k(i)+0;
    else
        k(i)=k(i)+1;
    end;
end;
jum=sum(k);
disp('Menghitung tingkat keberhasilan simulasi jaringan LVQ')
disp(strcat('jumlah data yang tepat dikenali = ',int2str(jum)))
disp(strcat('jumlah data yang diujikan = ',int2str(j)))
b=(jum/j)*100;
b1=num2str(b);
disp(strcat('tingkat keberhasilan = ',b1))
%disp('tingkat keberhasilan = ',b1)
disp('P R O S E S   P E N G U J I A N')
disp('Data ke- Target Hasil')
disp([(1:length(Tc))' Tc' Yc])
pause
```

Lampiran 2

Listing program Checking Model LVQ

```
lvqjantung(data);  
cek = xlsread('data1vqcek');  
cek1=cek(:,2:11);  
pchk1=cek1';  
Y = sim(net,pchk1);  
Yc1 = vec2ind(Y);  
if Yc1~=1,  
    disp('maaf, jantung anda kurang sehat')  
else  
    disp('selamat, jantung anda sehat')  
end
```