TUGAS MATA KULIAH JARINGAN SARAF TIRUAN

LEARNING VECTOR QUANTIZATION

Disusun oleh : KELOMPOK SATU

1. Christian Y. S. (1211501075)
2. Demmy Dwi Rhamadan (1211500176)
3. Fauzi Yudi S. (12/334626/PA/14859)
4. Joanna Dyas E. Pepe (12/331086/PA/14443)
5. Meisyarah Dwiastuti (12/331469/PA/14720)
6. Nur Rista K. (12/334855/PA/15044)

JURUSAN ILMU KOMPUTER DAN ELEKTONIKA

FAKULTAS MIPA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS GADJAH MADA

JOGJAKARTA

2014

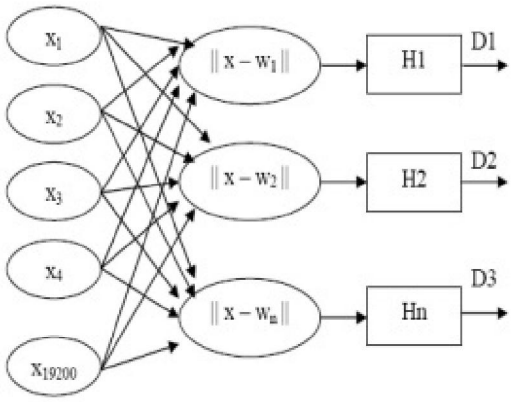
Apa itu Learning Vector Quantization ?

Learning Vector Quantization (LVQ) merupakan salah satu algoritma *supervised neural network* yang menggunakan strategi *competitive learning* yang dikenal sebagai jaringan *winner-take-all.* Algoritma ini berhubungan dengan *supervised neural network* lain seperti algoritma Perceptron dan Back-propagation. Algoritma ini juga berhubungan dengan *competitive learning* lain seperti algoritma Self-Organizing Map (SOM) yang memiliki kemiripan, tetapi merupakan *unsupervised learning* dengan tambahan hubungan antar neuron.

Karena LVQ merupakan *supervised learning,* algoritma ini sesuai untuk masalah klasifikasi. Sistem LVQ direpresentasikan oleh banyak n prototype yang didefinisikan dalam *feature space* data. Dalam algoritma *winner-take-all,* untuk setiap titik data akan ditetntukan prototype yang paling dekat dengan input berdasarkan *distance measure* yang diketahui. Posisi dari prototype (yang menang) tersebut kemuadian beradaptasi. Misalnya, prototype menjadi mendekat jika mengklasifikasi titik data dengan benar dan menjauh jika mengklasifikasi titik data dengan salah.

Bagaimana Arsitektur Learning Vector Quantization?

Arsitektur LVQ dapat dilihat seperti berikut:



Arsitektur Learning Vector Quantization

a) x1 sampai dengan x19200 = nilai input
b) || x – w1 || sampai dengan || x – wn || = jarak bobot
c) H1 sampai dengan Hn = lapisan output
d) D1 sampai dengan Dn = nilai output

e) n = jumlah data karakter (jumlah kelas)

dimana:

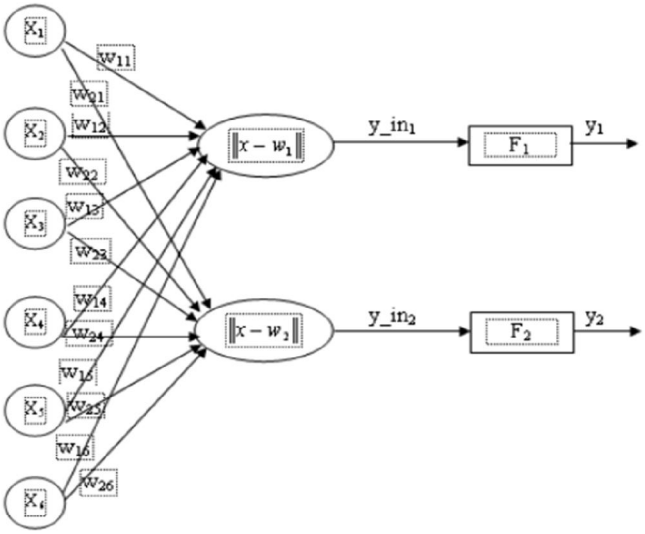
- x1 sampai dengan x19200 merupakan elemen matriks dalam setiap pola karakter yang akan dijadikan sebagai nilai input.
- -|| x – w1 || sampai dengan || x – wn || merupakan perhitungan jarak bobot terkecil dengan w1 sampai dengan wn adalah nilai data inisialisasi.
- H1 sampai dengan Hn adalah lapisan output.
- D1 sampai dengan Dn adalah bobot akhir yang nantinya akan dipakai dalam proses pengujian dengan data karakter baru yang dimasukkan.

Pelatihan jaringan syaraf tiruan dikatakan berhasil jika pelatihan konvergen, dan gagal jika pelatihan divergen. Suatu pelatihan dikatakan konvergen jika kesalahan pada setiap iterasi pelatihan selalu mengecil, sampai pada titik dimana nilai bobot pada setiap neuron telah mencapai nilai yang paling baik untuk data pelatihan yang diberikan. Sebaliknya, pelatihan dikatakan divergen jika kesalahan pada pelatihan tidak cenderung mengecil menuju sebuah titik tertentu.

Bagaiman Bentuk Fungsi Aktivasi Pada Learning Vector Quantization ?

Lapisan kompetitif (terjadi kompetisi pada input untuk masuk 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. 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. Misalkan ada enam variable dari vektor input, yaitu X = {x1,x2,x3,x4,x5,x6} dengan neuron keluaran Y1 dan Y2, serta dua vektor bobot yaitu W1j = {W11,W12,W13,W14,W15,W16} dan W2j = {W21,W22,W23,W24,W25,W26}. Maka akan mempunyai gambaran seperti dibawah ini



Jaringan LVQ dengan 6 unit input dan 2 vektor bobot.

Berdasarkan gambar diatas, tampak bahwa dalam LVQ terdapat dua vektor bobot yang menghubungkan setiap neutron masukkan dengan neuron keluarga 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 (W1 dan W2). Dalam hal ini W1 adalah vektor bobot yang menghubungkan setiap neuron pada lapisan input ke neuron pertama pada lapisan output, **fungsi aktivasi (F)** yang digunakan pada arsitektur jaringan LVQ adalah *fungsi linear*. Tujuannya adalah agar diperoleh keluaran yang sama dengan masukkan, sesuai dengan rumus *fungsi linear* yaitu y = x. **Fungsi aktivasi F1** akan memetakan y_in1 ke Y1 = 1 apabila | x – W1 | < | x – W2 |, dan y1 = 0 jika sebaliknya. Demikian pula **fungsi aktivasi F2** akan memetakan y_in2 ke Y2 = 1 apabila | x – W2 | < | x – W1 |, dan y2 = 0 jika sebaliknya.

Bagaimana Algoritma Pembelajaran pada Learning Vector Quantization ?

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, masukkan dapat dikenali kelasnya berdasarkan neuron keluaran yang diperoleh. Metode LVQ mengenali pola masukkan berdasarkan pada pendekatan jarak antara dua vektor yaitu vektor dari unit/neuron masukkan dengan vektor bobot. Pengenalan metode LVQ ini terdapat dua proses yaitu :

- Proses pembelajaran
- Proses 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)

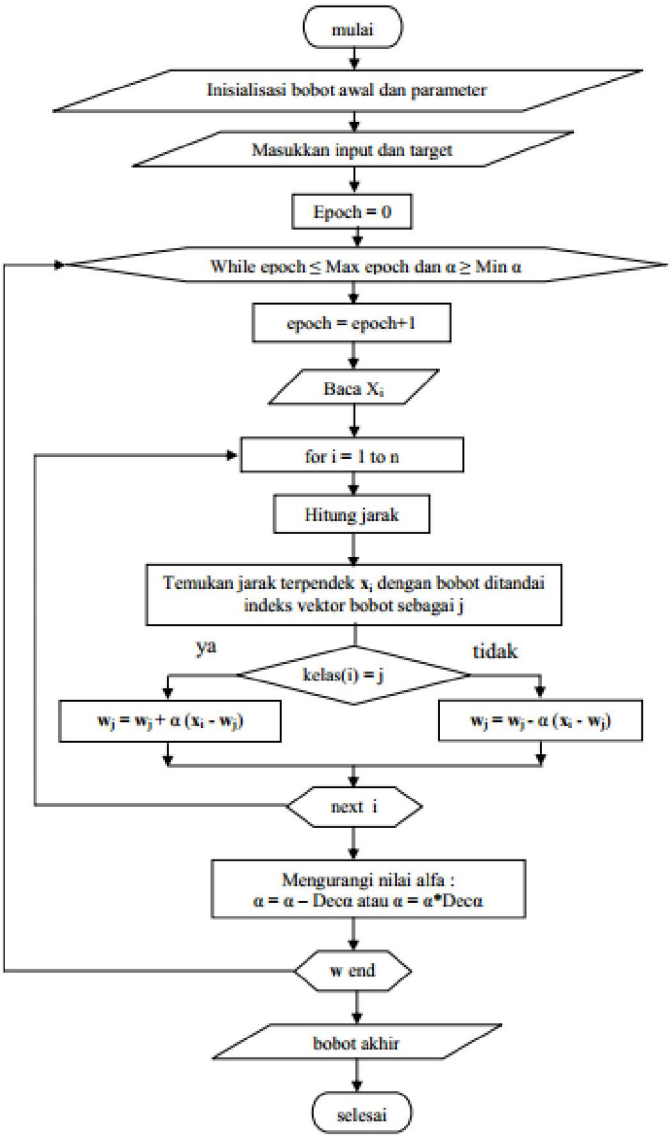
   Alfa didefinisikan sebagai tingkat pembelajaran. Jika alfa terlalu besar, maka algoritma akan menjadi tidak stabil sebaliknya jika alfa terlalu kecil, maka prosesnya akan terlalu lama. Nilai alfa adalah $0 < \alpha < 1$.
2. DecAlfa (Penurunan Learning rate)

   Yaitu penurunan tingkat pembelajaran.
3. MinAlfa (Minimum Learning rate)

   Yaitu minimal nilai tingkat pembelajaran yang masih diperbolehkan.
4. MaxEpoch (Maksimum epoch)

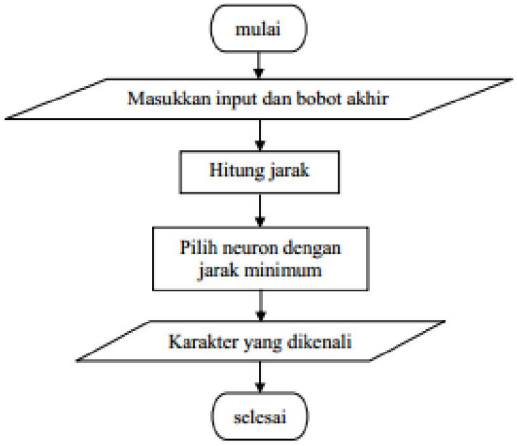
   Yaitu jumlah epoch atau iterasi maksimum yang boleh dilakukan selama pelatihan. Iterasi akan dihentikan jika nilai epoch melebihi epoch maksimum. Misalkan terdapat *n* buah data dengan m buah variabel input. Data-data tersebut terbagi dalam K kelas. Motivasi dari algoritma LVQ adalah menemukan unit output yang paling dekat dengan vektor input.

Algoritma proses pembelajaran pada metode LVQ disajikan pada Gambar berikut



Algoritma Proses Pembelajaran pada Metode Learning Vector Quantization

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 berikut.



Algoritma Proses Pembelajaran pada Metode Learning Vector Quantization

CONTOH SOAL DAN STUDI KASUS LEARNING VECTOR QUANTIZATION

A. Soal Learning Vector Quantization Contoh Sederhana

Lima vektor akan ditetapkan pada 2 kelas yaitu vektor :

$X_1 = (1,1,0,0)$ kelas 1

$X_2 = (0,0,0,1)$ kelas 2

$X_3 = (0,0,1,1)$ kelas 2

$X_4 = (1,0,0,0)$ kelas 1

$X_5 = (0,1,1,0)$ kelas 1

1. Pertama-tama inisialisasi bobot awal, kita akan mengambil bobot awal untuk kelas 1 memakai vektor 1 dan bobot awal kelas 2 memakai vektor 2.
   Maka akan didapat :
   $W_1 = (1,1,0,0)$
   $W_2 = (0,0,0,1)$
   Inisialisasi rating pembelajaran awal misalkan 0,2.

2. Tentukan jarak terdekat pada vektor 3, vektor 4 dan vektor 5. Apakah mereka masuk ke kelas 1 atau kelas 2.
   (0,0,1,1) → Jarak (0,0,1,1) ke W_1 (1,1,0,0) dihitung dengan cara
   $D = \sum(x_3 - W_1)^2$
   $D_1 = [\ (0,0,1,1) - (1,1,0,0)\ ]$
   $= (-1,-1,1,1) = (-1^2)+(-1^2)+(1^2)+(1^2) = 4$
   Sedangkan untuk jarak (0,0,1,1) ke W_2 (0,0,0,1) diperoleh
   $D_2 = [\ (0,0,1,1) - (0,0,0,1)\ ]$
   $= (0,0,1,0) = (0^2)+(0^2)+(1^2)+(0^2) = 1$
   Maka dipilih jarak terpendek yaitu ke W_2, sehingga vektor ke-2 dimasukkan ke kelas 2
   Dengan cara yang sama vektor 4 dan vektor 5 maka akan diperoleh
   (1,0,0,0) = lebih dekat ke W_1
   (0,1,1,0) = lebih dekat ke W_1

3. Perbarui W_2 menggunakan vektor 3 karena vektor 3 lebih dekat ke W_2.
   $W_2 = W_2 + 0.2(X_3 - W_2)$

$= (0,0,0,1) + 0.2[(0,0,1,1) – (0,0,0,1)]$

$= (0, 0, 0.2, 1)$

4. Perbarui W_1 menggunakan vektor 4 karena lebih dekat ke W_1.

$W_1 = W_1 + 0.2(X_4 – W_1)$

$= (1,1,0,0) + 0.2[(1,0,0,0) – (1,1,0,0)]$

$= (1, 0.8, 0, 0)$

5. Perbarui W_1 menggunakan vektor 5 karena lebih dekat ke W_1.

$W_1 = W_1 + 0.2(X_4 – W_1)$

$= (1,0.8,0,0) + 0.2[(0,1,1,0) – (1,0.8,0,0)]$

$= (1,0.8,0,0) + 0.2(-1,0.2,1,0)$

$= (0.8, 1.04, 0.2, 0)$

Dengan langkah ini akan diperoleh bobot baru untuk kelas 1 dan kelas 2 yaitu :

$W_1 = (0.8, 1.04, 0.2, 0)$

$W_2 = (0, 0, 0.2, 1)$

Dari bobot tiap kelas ini, dilakukan pengecekan ulang untuk X_1 sampai X_5 apakah masuk di kelas 1 atau kelas 2. Perhitungan dengan menggunakan jarak terpendek dan diperoleh :

$X_1 = (1,1,0,0)$ akan masuk ke kelas : 1

$X_2 = (0,0,0,1)$ masuk ke kelas : 2

$X_3 = (0,0,1,1)$ masuk ke kelas : 2

$X_4 = (1,0,0,0)$ masuk ke kelas : 1

$X_5 = (0,1,1,0)$ masuk ke kelas : 1

Akan dilakukan perulangan langkah ke 2 hingga langkah ke 5 dengan menggunakan bobot baru yang didapat dan menggunakan rating pembelajaran yang lebih kecil.

STUDI KASUS : 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, 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 yang disediakan oleh UCI (University Of Calfornia at Irvine) . 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 dituliskan dengan $w_{1j} = (w_{11}, w_{12}, w_{13}, \ldots, w_{1m})$ yang merupakan vektor bobot kelas pertama dengan m variabel dan $w_{2j} = (w_{21}, w_{22}, w_{23}, \ldots, w_{2m})$ 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), α = 0.25
2. DecAlfa (Penurunan Learning rate), Decα = 0.1
3. MinAlfa (Minimum Learning rate), Minα = 0.001
4. MaxEpoch (Maksimum epoch), MaxEpoch = 100

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;
bl=num2str(b);
disp(strcat('tingkat keberhasilan = ',bl))
%disp('tingkat keberhasilan = ',bl)
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
```

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

Listing program Checking Model LVQ

```
lvqjantung(data);
cek = xlsread('datalvqceck');
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
```

Hasil perhitungan memberikan kesimpulan bahwa pasien dengan ciri-ciri tersebut positif penderita penyakit jantung.

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

Referensi

http://download.portalgaruda.org/article.php?article=116150&val=5271&title=

http://ejournal.undip.ac.id/index.php/media_statistika/article/download/2522/2313

http://www.cleveralgorithms.com/nature-inspired/neural/lvq.html

http://en.wikipedia.org/wiki/Learning_vector_quantization