

## ANALISIS CLUSTER DAN DIAGNOSA PENYAKIT MENGUNAKAN JARINGAN SYARAF TIRUAN

**Zeth Arthur Leleury, Henry Willyam Michel Patty**  
*Jurusan Matematika FMIPA Universitas Pattimura Ambon*  
*Jln. Ir. M. Putuhena, Kampus Poka*  
*zaleleury@yahoo.co.id & henry\_4t00@yahoo.com*

### ABSTRAK

Dengan tujuan pengklasteran multi-penyakit, Self-Organizing Map (SOM) dikembangkan. Dalam tulisan ini, ciri-ciri penyakit tomat diekstrak dan pemetaan hubungan antara penyakit dan ciri-cirinya dibuat. Menurut analisis, Jarak Euclid diambil sebagai pemisah utama dan pencarian yang berdekatan (*adjacent-searching*). Model yang diusulkan berbasis SOM memiliki dua layer. Susunan ciri-ciri penyakit dapat diurutkan secara akurat dan cepat kemudian dikelompokkan menggunakan model ini. Selanjutnya digunakan jaringan Learning Vector Quantization (LVQ) untuk uji diagnosa penyakit. Hasil simulasi menunjukkan bahwa kedua model yang diajukan berkinerja dengan baik.

**Kata kunci:** *adjacent-searching*, learning vector quantization, pengklasteran, penyakit tomat, self-organizing map

---

### PENDAHULUAN

Self Organizing Maps (SOM) merupakan perluasan dari jaringan kompetitif yang sering disebut sebagai jaringan Kohonen. Jaringan ini menggunakan metode pembelajaran *unsupervised*. Jaringan ini pertama kali diperkenalkan oleh Prof. Teuvo kohonen pada tahun 1982. Pada jaringan ini, suatu lapisan yang berisi neuron-neuron akan menyusun dirinya sendiri berdasarkan input nilai tertentu dalam suatu kelompok yang dikenal dengan istilah cluster. Selama proses penyusunan diri, cluster yang memiliki vektor bobot paling cocok dengan pola input (memiliki jarak paling dekat) akan terpilih sebagai pemenang. Neuron yang menjadi pemenang beserta neuron-neuron tetangganya akan memperbaiki bobot-bobotnya (Fausset, 1989)

Diagnosa adalah suatu aplikasi penting dari kecerdasan buatan (*artificial intelligence*). Hal ini umum bahwa multi-penyakit terjadi secara bersamaan dalam penyakit pertanian dan penyakit manusia. Suatu penyakit bisa disebabkan karena berbagai alasan, dan satu alasan juga dapat menyebabkan banyak penyakit. Pengklasteran penyakit pertanian sebagai contoh saat ini didasarkan pada jaringan saraf (*neural network*).

Hasil penelitian (Gil, Johnsson, Chamizo, Soriano, & Ruiz, 2009) menunjukkan bahwa jaringan *self-organizing map* (SOM) adalah jaringan saraf kompetitif dan kooperatif yang dapat menampung vektor input dari struktur topologi dan peta data jaringan berdimensi tinggi ke yang lebih rendah. *Iatrical scholar* menggunakan teknik SOM dan teori himpunan untuk menemukan kecenderungan kondisi pasien penderita penyakit kardiovaskular secara individual (Chou et al, 2007). Selain itu, Jaringan SOM juga telah sangat cocok untuk analisis cluster (Zhang, et al, 2010).

Jaringan saraf pengelompokkan, termasuk Learning Vector Quantization (LVQ) harus diterapkan untuk diagnosa penyakit. Jaringan ini berhasil digunakan dalam klasifikasi dari dataset penyakit tiroid untuk memperoleh diagnosa (Temurtas, 2007).

Dalam tulisan ini, model diagnosa jaringan SOM dua-level diusulkan, dan sistem pemetaan sifat penyakit dikembangkan untuk pengklasteran. Perbandingan dan klasifikasi yang diambil adalah penyakit, kumpulan kelompok ciri-ciri gejala sampel penyakit, dan kumpulan kelompok ciri-ciri gejala penyakit yang nyata. Untuk uji diagnosa penyakit digunakan jaringan Learning Vector Quantization (LVQ). Diagnosa penyakit pertanian

didasarkan pada kondisi pertumbuhan tanaman abnormal yang berarti suatu penyakit dikenali oleh gejala penyakit. Karena tomat rentan terhadap beberapa penyakit (Lv, 2000), terutama dalam lingkungan tumbuh abnormal maka digunakan sebagai contoh dalam penelitian ini. Selanjutnya untuk simulasi digunakan *software* Matlab.

## METODE

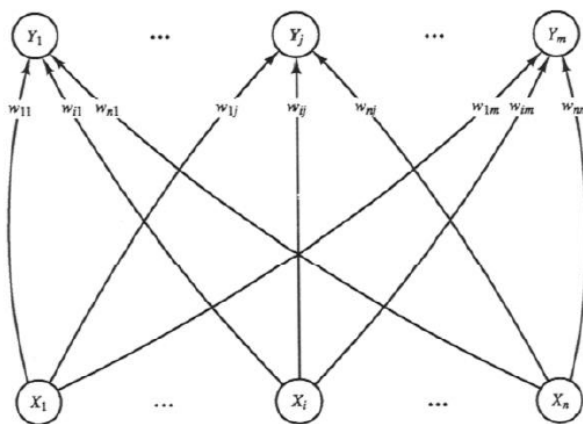
### Algoritma Self Organizing Map (SOM)

SOM dianggap sebagai salah satu jaringan saraf yang disukai untuk analisis pengelompokan. Titik kunci dari algoritma SOM adalah dengan perhitungan geometri. Agar bisa digunakan, *neural network* harus dilatih dulu. Pada prinsipnya ada dua cara untuk melatih network, yaitu dengan *supervised learning* dan *unsupervised learning* (Hagan *et al*, 1996). *Supervised learning* atau pembelajaran dengan diawasi memerlukan seperangkat data *input* untuk melatih yang terdiri dari pasangan data *input* dan data target. Network dilatih agar dapat memproses data *input* dan mengeluarkan *output* seperti yang ditargetkan. Jadi network akan mengatur diri sendiri agar dapat bertingkah seperti yang dilatihkan.

Pada *unsupervised learning*, di sini network hanya diberi seperangkat data *input* saja tanpa adanya target yang diharapkan. Network akan melatih dirinya sendiri untuk bisa mengenali pola data *input* dan kemudian mengklasifikasikannya. Berapa jumlah kelompok data *output* tidak ditentukan dan jumlah kelompok data ini bisa terus bertambah tanpa batas. Dalam *unsupervised* ini, network hanya diberi rambu-rambu tentang tingkat keseragaman data (*vigilance parameter*,  $\rho$ ) *input* saja. Atau dengan kata lain, jumlah kelompok data yang terbentuk ditentukan oleh tingkat keseragaman data. Jika tingkat keseragaman rendah maka jumlah kelompok data yang dihasilkan akan sedikit dan demikian sebaliknya.

Jaringan Kohonen SOM (*Self-Organizing Map*) merupakan salah satu model jaringan syaraf yang menggunakan metode pembelajaran *unsupervised*. Jaringan Kohonen SOM terdiri dari dua lapisan (*layer*), yaitu lapisan *input* dan lapisan *output*. Setiap neuron dalam lapisan *input* terhubung dengan setiap neuron pada lapisan *output*. Setiap neuron dalam lapisan *output* merepresentasikan kelas dari *input* yang diberikan.

Gambar struktur JST Kohonen terlihat seperti pada **Gambar 1**.



**Gambar 1.** Struktur JST Kohonen SOM

Ada beberapa langkah dalam penerapan algoritma. Ini adalah kompetisi dan pembelajaran untuk mendapatkan pemenang dalam proses. Langkah-langkah tersebut adalah:

Langkah 0 : Inisialisasi pembobotan dengan nilai random. Menyet parameter learning rate( ), dan radius tetangga ( )

Langkah 1 : Apabila kondisi selesai belum terpenuhi, lakukan langkah 2-8

Langkah 2 : Untuk tiap vektor input  $x$  ( , = 1,2, ..., ) lakukan langkah 3-5

Langkah 3 : Untuk tiap ( = 1,2, ..., ), hitung jarak Euclidean

$$() = \sum ( - )$$

Langkah 4 : Mencari indeks dengan jarak ( ) terdekat (minimum)

Langkah 5 : Melakukan perbaikan nilai dengan nilai tertentu. Yaitu:

$$( ) = ( ) + [ - ( ) ]$$

Langkah 6 : Melakukan update learning rate.

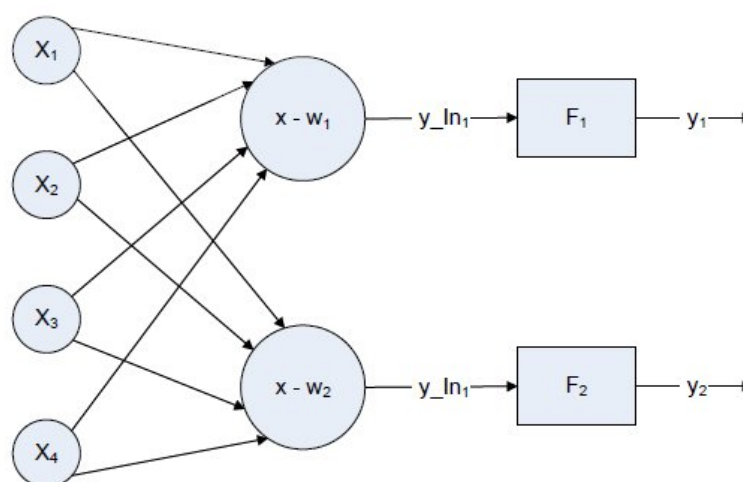
$$( ) = 0.5 ( )$$

Langkah 7 : Mereduksi radius dari fungsi tetangga pada waktu tertentu (epoch).

Langkah 8 : Menentukan kondisi STOP.

### Algoritma Learning Vector Quantization (LVQ)

Learning vector quantization (LVQ) merupakan suatu metode klasifikasi dimana jumlah kelompok yang diharapkan sudah ditentukan. Berbeda dengan jaringan SOM, LVQ merupakan salah satu metode dalam Jaringan Syaraf Tiruan untuk melakukan pembelajaran terhadap layer yang *supervised*.



Gambar 2. Struktur JST LVQ

**Gambar 2** menunjukkan jaringan LVQ dengan unit pada lapisan input, dan 2 unit (neuron) pada lapisan output. Pemrosesan yang terjadi pada setiap neuron adalah mencari jarak antara suatu vektor input ke bobot yang bersangkutan ( dan ). adalah vektor bobot yang menghubungkan setiap neuron pada lapisan input ke neuron pertama pada lapisan output, sedangkan adalah vektor bobot yang menghubungkan setiap neuron pada lapisan input ke neuron yang kedua pada lapisan output. Fungsi aktivasi  $F_1$  akan memetakan  $y_{in1}$  ke  $y_1 = 1$  apabila  $| - | < | - |$  dan  $y_1 = 0$  jika sebaliknya. Demikian pula dengan yang terjadi pada fungsi aktivasi  $F_2$ , akan memetakan  $y_{in1}$  ke  $y_1 = 1$  apabila  $| - | < | - |$ , dan  $y_1 = 0$  jika sebaliknya. Adapun algoritma untuk LVQ adalah sebagai berikut.

Langkah 0 : Inisialisasi

Langkah 1 : Apabila kondisi selesai belum terpenuhi, lakukan langkah 2-6

Langkah 2 : Untuk tiap vektor training  $x ( , = 1,2, \dots, )$ , lakukan langkah 3-4

Langkah 3 : Dapatkan nilai sehingga jarak Euclidean  $-$  bernilai minimum

Langkah 5 : Update nilai bobot dengan nilai tertentu.yaitu:

$$( ) = ( ) + [ - ( ) ] ; \text{jika } =$$

$$( ) = ( ) - [ - ( ) ] ; \text{jika } \neq$$

Langkah 6 : Melakukan update learning rate

$$( ) = 0.5 ( )$$

Langkah 8 : Uji kondisi STOP.

dengan,

: Kategori dari training vektor yang benar

: Kategori (hasil training)

Setelah dilakukan pelatihan, akan diperoleh bobot akhir ( ). Bobot-bobot ini nantinya akan digunakan untuk melakukan simulasi atau pengujian data yang lain.

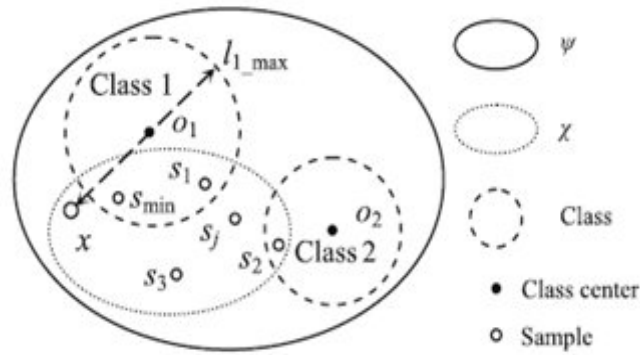
**Analisis adjacent-searching dalam SOM**

Algoritma SOM mengambil pusat kelas (output neuron dengan penyesuaian terbaik) sebagai kondisi pencarian dengan menghitung jarak Euclid dari vektor input ke pusat kelas.

Seperti ditunjukkan dalam Gambar 3, dengan asumsi  $\psi$  didefinisikan sebagai agregasi pemetaan ciri penyakit dan  $\chi$  sebagai agregasi sampel.

$\psi$  ini terdiri dari kelas 1 dan 2 kelas (masing-masing, menentukan dua kelas sebagai  $C_1$  dan  $C_2$ , pusat kelas masing-masing adalah  $o_1$  dan  $o_2$ , serta radiusnya adalah  $r_1$  dan  $r_2$ ).

$(x, y) \leq (x, y)$  diketahui jika sampel input adalah  $x$ . Kemudian, beberapa elemen akan berada di  $C_1$  dan  $C_2$ , sampel yang terdekat dengan  $o_1$  adalah  $s_{min}$ , jarak terkecil di  $C_2$  adalah  $s_2$ . Dalam rangka untuk membuat hasil pencarian berada di  $C_1$ ,  $s_{min}$  yang tidak termasuk dalam  $C_2$  harus dibuktikan.



**Gambar 3.** adjacent-searching dalam agregasi pemetaan ciri penyakit

Jelas bahwa kuadrat dari jarak Euclid terkecil  $l_{min}$  dari  $x$  ke kelas adalah  $[(x, y) - (o_1, o_1)]^2$ , dan kuadrat terbesar  $l_{max}$  adalah  $[(x, y) + (o_1, o_1)]^2$ . Kemudian

$$\begin{aligned} (x, y) &\geq (o_1, o_1) - l_{min}^2 \\ (x, y) &\geq (o_1, o_1) - l_{max}^2 \end{aligned} \tag{1}$$

Sehingga

$$(x, y) \geq [(x, y) - l_{min}^2] \tag{2}$$

Karena diketahui  $l_{1\_max} < l_{2\_min}$ , maka:

$$(x, y) \geq [(x, y) - l_{1\_max}^2] > (x, y) + l_{2\_min}^2 \tag{3}$$

Karena

$$(x, y) \in \psi, (x, y) < l_{1\_max} \tag{4}$$

sehingga diperoleh  $(x, y) < (o_1, o_1)$ . Dengan demikian,  $s_{min}$  termasuk dalam  $C_1$

sehingga  $(x, y) < (o_1, o_1)$  dan ditemukan

$$\psi \cap \chi = \emptyset \tag{5}$$

Dengan cara lain, asumsikan  $x \in \chi$  maka

$$(x, y) \geq [(x, y) - l_{min}^2] \tag{6}$$

Berdasarkan definisi dari pencarian kelas yang berdekatan maka dapat dibuktikan:

$$(x, y) + l_{2\_min}^2 \leq [(x, y) - l_{1\_max}^2] \text{ atau } (x, y) \leq \tag{7}$$

Diketahui

$$l_{1\_max} = [(x, y) - l_{min}^2] < (x, y) + l_{2\_min}^2 = \tag{8}$$

tetapi

$$l_{2\_min} < l_{1\_max} = [(x, y) - l_{min}^2] < l_{2\_min} = (x, y) + \tag{9}$$

sehingga

$$(x, y) + l_{2\_min}^2 \leq [(x, y) - l_{min}^2] \tag{10}$$

salah. Karena  $\cap = \emptyset$  maka  $( , ) > .$  Dari (10), proposisi tidak dapat dibuktikan dengan benar, maka

$$\cap = \emptyset \tag{11}$$

Jadi kesimpulan yang didapat: untuk suatu sampel input , jika  $\in U$  dan untuk yang memenuhi  $\in U$  , (21) diperoleh:

$$( , ) < ( , ) \tag{12}$$

maka untuk *arbitrary map distribution*, syarat perlu dan syarat cukup dari (5) (sama seperti (11)) adalah:

$$\begin{aligned} [ ( , ) - ] > ( , ) + \\ ( , ) > \end{aligned} \tag{13}$$

### HASIL SIMULASI

Suatu lingkungan tumbuh sangat merugikan membuat ciri-ciri serupa dari penyakit bersamaan. Karakteristik yang paling umum ada pada penyakit tomat. Ada 22 macam single-penyakit berbeda dan penyakit ini dapat berisi 24 ciri penyakit yang mungkin (Lv, 2000).

#### Pemetaan penyakit daun tomat dan ciri-cirinya

Dalam keadaan ini, ciri-ciri yang sama dapat mengkonfirmasi berbagai penyakit. Sebagai contoh: *Abnormal green spot* merupakan ciri penyakit Leaf Mold, Cercospora Leaf Mold, Early Blight, dan lainnya sehingga ciri tersebut dapat bertindak sebagai kriteria klasifikasi dan memperkecil jangkauan pencarian. Adapun atribut ciri-ciri penyakit daun yang akan dipakai dalam pengelompokkan adalah seperti yang disajikan dalam **Tabel 1** berikut:

**Tabel 1.** Atribut 24 ciri penyakit daun tomat

Code	Ciri-Ciri	Code	Ciri-Ciri
1	<i>Abnormal green spot</i>	13	<i>Papilose or punctuated spot</i>
2	<i>White spot</i>	14	<i>Abnormal red mildew layer</i>
3	<i>Yellow or tawny spot</i>	15	<i>White mildew layer</i>
4	<i>Brown spot</i>	16	<i>Brown mildew</i>
5	<i>Black spot</i>	17	<i>Black mildew layer</i>
6	<i>Grey spot</i>	18	<i>Grey mildew layer</i>
7	<i>Irregular shape spot</i>	19	<i>Irregular shape mildew layer</i>
8	<i>Rounded or oval spot</i>	20	<i>Felty and pilose mildew layer</i>
9	<i>Strip or line type spot</i>	21	<i>Villose or filose mildew layer</i>
10	<i>Circle or wheeling veins spot</i>	22	<i>Putrid</i>
11	<i>Cupped spot</i>	23	<i>Withered</i>
12	<i>Rust water or water spot</i>	24	<i>Color changed</i>

Data penyakit daun tomat yang akan dikelompokkan berdasarkan ciri-ciri penyakit adalah sebanyak 22 penyakit. Selanjutnya terhadap data atribut ciri-ciri penyakit tersebut akan dilakukan pengkodean sebagai berikut: data diberi kode 1 jika ada ciri terjadi, dan kode 0 jika ciri tersebut tidak ada. Dari hasil pengkodean selanjutnya disajikan seperti pada Tabel 2 berikut.

**Tabel 2.** Data atribut yang telah decoding

No	Nama Penyakit	Atribut Ciri-Ciri Penyakit																								
		1	2	3	4	5	6	7	8	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	Leaf Mold	1	0	1	0	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1
2	Cercospora Leaf Mold	1	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	
3	Early Blight	1	0	1	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	Corynespora Leaf Spot	0	0	1	1	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	1	1	0	0	0	
5	Fruit Rot	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	Septoria Leaf Spot	1	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	
7	Verticillium Wilt	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	
8	Fusarium Wilt	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	1	0	1
9	Bacterial Leaf Spot	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
10	Powdery Mildew	1	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1
11	Helminthospor Fruit Rot	0	0	0	1	0	1	0	1	1	1	1	0	0	0	0	0	1	0	0	1	0	0	0	0	
12	Late Blight	1	0	0	1	1	0	1	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	1	0
13	Gray Mold	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	0	1	0	0
14	Bacterial Scab	1	0	0	1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	Stem Rot	0	0	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0
16	Stem Blight	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
17	Brown Rot	1	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0
18	Gray Leaf Spot	1	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	Mosaic Virus I	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
20	Mosaic Virus II	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
21	Gray Spot	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	Unknown disease	0	1	0	0	1	1	0	1	0	1	0	0	0	1	0	0	0	0	0	1	1	0	1	1	0

**Pengelompokan sampel penyakit tomat**

Dari output hasil pengelompokan menggunakan jaringan self-organizing map (SOM) terlihat bahwa 22 penyakit tanaman tersebut telah terbagi menjadi 12 kelas, yaitu:

**Tabel 3.** Data Pengklasteran 24 Penyakit Daun Tomat

Kelas	Nama Penyakit
1	Leaf Mold, Cercospora Leaf Mold
2	Powdery Mildew
3	Verticillium Wilt, Mosaic Virus I
4	Stem Blight, Brown Rot
5	Bacterial Scab, Mosaic Virus II
6	Late Blight
7	Fusarium Wilt, Stem Rot
8	Corynespora Leaf Spot, Gray Mold
9	Unknown disease
10	Helminthospor Fruit Rot, Gray Leaf Spot
11	Septoria Leaf Spot
12	Early Blight, Fruit Rot, Bacterial Leaf Spot, Gray Spot

Dengan pengaturan epoch menjadi 5000 maka hasil cluster seperti yang ditunjukkan pada Tabel di atas diidentifikasi baik dan dapat diterima karena ternyata setiap penyakit pada kelas yang sama memiliki kedekatan sesuai dengan ciri penyakit .

### Uji Diagnosa

Data ciri-ciri dan penyakit dalam tulisan ini diperoleh dari Fujian Academy of Agricultural Sciences. Beberapa hasil pengujian ditunjukkan pada **Tabel 4** berikut:

**Tabel 4.** Uji diagnosa untuk 5 (lima) sampel penyakit

Kombinasi ciri penyakit (code)	Hasil Diagnosa Menggunakan LVQ Neural Network					
	Epoch 500	Epoch 1000	Epoch 2000	Epoch 3000	Epoch 4000	Epoch 5000
101111010101000001000100	Early Blight	Early Blight	Early Blight	Early Blight	Early Blight	Early Blight
110001110010001000010100	Gray Leaf Spot	Powdery Mildew	Powdery Mildew	Gray Leaf Spot	Gray Leaf Spot	Gray Leaf Spot
110000100000001000010100	Powdery Mildew	Powdery Mildew	Powdery Mildew	Powdery Mildew	Powdery Mildew	Powdery Mildew
100000100101000000000011	Brown Rot	Brown Rot	Brown Rot	Brown Rot	Brown Rot	Brown Rot
001110000000000000000000	Bacterial Leaf Spot	Bacterial Leaf Spot	Bacterial Leaf Spot	Bacterial Leaf Spot	Bacterial Leaf Spot	Bacterial Leaf Spot

Hasil simulasi menunjukkan bahwa model algoritma LVQ untuk diagnosa dapat dengan tepat membedakan antara single-penyakit dan multi-penyakit terutama ketika ciri-ciri inputan diuji pada beberapa epoch berbeda. Pemetaan hasil diagnostik untuk agregasi penyakit ini sesuai dengan data (Lv, 2000).

### DAFTAR PUSTAKA

- Chou, H. C., Cheng, C. H., & Chang, J. R. (2007). Extracting drug utilization knowledge using self-organizing map and rough set theory. *Expert Systems with Applications*, 33(2), 499–508.
- Gil, D., Johnsson, M., Chamizo, J. M. G., Soriano, P. A., & Ruiz, F. D. (2009). Application of artificial neural networks in the diagnosis of urological dysfunctions. *Expert Systems with Applications*, 36(3), 5754–5760.
- Hagan, M. T., Demuth, H. B., and Beale, M., 1996, *Neural Network Design*, An International Thompson Pub. Co., USA.

- Fausett, L.(1989). Fundamentals Of Neural Networks, Architecture, Algorithms, and applications
- Ly, P. (2000). The primitive color picture of the Chinese vegetables' pests & disease (2<sup>nd</sup> ed.), Beijing: HuaXia Publishing Company (in Chinese).
- Temurtas, F. (2007). A comparative study on thyroid disease diagnosis using neural networks. *Expert Systems with Applications*, 36(1), 944–949.
- Zhang, K., Chai, Y., & Yang, S.X. (2010). Self-organizing feature map for cluster analysis in multi-disease diagnosis. *Expert Systems with Applications* 37, 6359–6367