

# **SELF ORGANIZING MAP UNTUK ANALISIS KLASTER PADA SPESIMEN DAUN FAMILI DIKOTILEDON**

Prapto Tri Supriyo<sup>1</sup>, Panji Wasmana<sup>2</sup>, Arum Madarum<sup>3</sup>

## **ABSTRAK**

*Self Organizing Map* (SOM) Kohonen yang diperkenalkan oleh T. Kohonen merupakan jaringan syaraf tiruan dengan pembelajaran *unsupervised* yang mampu memetakan data berdimensi tinggi ke dimensi rendah. Pada penelitian ini diimplementasikan algoritma SOM untuk pengklasteran famili dikotiledon berdasarkan ciri-ciri morfologi spesimen daun hasil penelitian di Jambi yang telah tersedia dalam basis data. Kinerja algoritma pengklasteran diukur dari nilai *cluster recall* dan *cluster precision*. Secara visual terlihat bahwa SOM mampu mengelompokkan spesimen daun dengan baik, namun hasil ini kurang sesuai jika dibandingkan dengan hasil klaster menurut tingkat taksonomi.

Kata kunci: *Self organizing map, clustering, cluster recall, cluster precision*, taksonomi.

## **PENDAHULUAN**

### **Latar Belakang**

Indonesia dikenal sebagai negara *mega diversity* hayati dunia yang memiliki keanekaragaman hayati sangat tinggi. Diperkirakan sekitar 17% dari seluruh spesies dunia dapat ditemukan di kawasan ini (Rasnovi, 2001).

Berbagai penelitian telah dilakukan berkenaan dengan keanekaragaman hayati tersebut. Diantaranya adalah penelitian terhadap tumbuhan. Sebelum melakukan penelitian terhadap tumbuhan, diperlukan identifikasi jenis-jenis organisme spesimen dalam area penelitian. Penentuan jenis atau disebut identifikasi jenis memerlukan sejumlah informasi taksonomi berupa ciri-ciri yang terdapat pada organisme yang digunakan sebagai dasar untuk membedakan antara satu jenis dengan jenis lainnya.

Menurut Stace (1990), identifikasi jenis tumbuhan dapat dilakukan berdasarkan: karakteristik morfologi dan anatomi struktur vegetatif dan generatif, informasi kimiawi, informasi kromosom, informasi sistem, dan informasi geografi dan ekologi.

Dalam pelaksanaannya proses identifikasi jenis tumbuhan ada kecenderungan yang kuat untuk tidak menggunakan informasi-informasi yang membutuhkan usaha yang besar, hal ini disebabkan alasan kemudahan dan kecepatan. Berdasarkan hal tersebut, informasi taksonomi

yang paling banyak dan umum digunakan adalah informasi yang berasal dari struktur morfologi dan anatomi organ vegetatif dan generatif. Organ-organ generatif seperti bunga dan buah pada golongan dikotiledon hanya dapat dijumpai pada waktu dan musim tertentu saja. Di lain pihak, organ vegetatif seperti daun, batang maupun cabang hampir selalu dijumpai sepanjang tahun. Dengan demikian dalam penelitian ini organ vegetatif dipilih sebagai informasi taksonomi. Untuk mendukung proses identifikasi ini diperlukan suatu metode pengelompokan jenis tumbuhan berdasarkan karakteristik utama yang dimilikinya.

*Self-organizing* adalah jaringan syaraf tiruan yang mampu mempelajari dan mengorganisir informasi tanpa memberikan *output* yang benar terhadap pola *input*, maka *self-organizing* melakukan pembelajaran secara *unsupervised* (Limin, 1994). Model *self-organizing* efektif dalam memecahkan permasalahan yang sangat kompleks.

### **Tujuan**

Tujuan dari penelitian ini adalah menerapkan sistem jaringan *self-organizing map* (SOM) Kohonen dalam sebuah aplikasi untuk mengelompokkan spesimen daun berdasarkan ciri morfologinya. Hasil dari pengelompokan daun tersebut untuk mempermudah dalam klasifikasi daun ke dalam tingkat taksonomi.

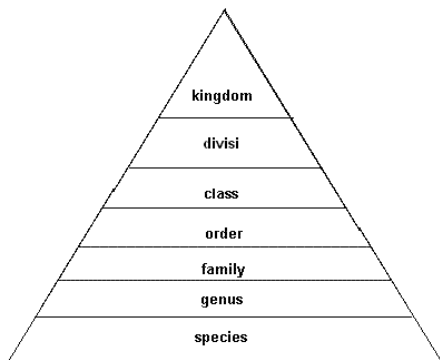
### Ruang Lingkup

Dalam penelitian ini proses klasifikasi hanya didasarkan pada organ vegetatif daun untuk famili dikotiledon. Hal ini disebabkan organ vegetatif dapat dijumpai sepanjang tahun sehingga mudah dalam pengumpulan data. Model jaringan syaraf tiruan yang digunakan dalam penelitian adalah model *self-organizing* yaitu jaringan SOM Kohonen.

## TINJAUAN PUSTAKA

### Taksonomi

Taksonomi adalah ilmu pengetahuan yang mencakup identifikasi, tata nama, dan klasifikasi objek, yang biasanya terbatas pada objek biologi (Rasnovi, 2001). Organisme tumbuhan diklasifikasikan ke dalam tujuh grup mayor (*taksa*), sebagai berikut: *Kingdom*, *Divisi*, *Class*, *Ordo*, *Famili*, *Genus*, dan *Spesies*. Di antara tujuh grup tersebut ada sub grup, grup-grup tersebut yaitu *superorder*, *subfamily*, dan *subspecies*. Susunan dari *taksa* pada tumbuhan dapat dilihat dalam Gambar 1.



Gambar 1 Susunan taksa pada tumbuhan.

### Karakter Taksonomi

Setiap tugas taksonomi diawali dengan pemilihan *operational taxonomic unit* (OTU) yang akan diklasifikasi, dan kemudian dipilih karakter-karakter dari OTU tersebut. Karakter yang dimaksud adalah ciri yang dapat berubah di antara unit taksonomi, dan kemungkinan nilai yang dapat diberikan disebut *state* (Dunn, 1982). Misalnya “mempunyai rambut” adalah sebuah karakter dengan *state* “ya” atau “tidak”.

### Tipe dan Pengkodean Karakter

Menurut Dunn (1982) tipe karakter dan pengkodean dapat dibagi menjadi dua yaitu karakter kualitatif dan karakter kuantitatif.

#### a. Karakter Kualitatif

Karakter kualitatif dapat berupa karakter sederhana yang terdiri dari 2 *state* yaitu ada atau tidak ada, *state* seperti 4, 5, 6, dan 13. Untuk konvensi, satu dari *state* dikodekan dengan 1 dan lainnya 0 (tanpa mempedulikan yang mana). Terdapat juga karakter yang mempunyai *multistate*, untuk konvensi setiap *state* dikodekan dengan 1, 2, 3. Tetapi angka tersebut tidak mempunyai nilai kuantitatif, sehingga tidak mungkin dapat dilakukan operasi aritmatik. Sebagai alternatif dari pengkodean 1, 2, dan 3 adalah dengan pengkodean dalam bentuk biner. Sebagai contoh pada Tabel 1, dapat dilihat pengkodean untuk karakter asal duri.

Tabel 1 Karakter asal duri

Asal Duri	Karakter biner		
Daun atau <i>stipula</i>	1	0	0
Epidermis	0	1	0
Cabang	0	0	1

#### b. Karakter Kuantitatif

Karakter kuantitatif adalah karakter yang membedakan antar OTU satu dan yang lainnya dengan cara penghitungan ke dalam skala interval. Variabel-variabel tersebut dengan mudah dikonversi ke dalam karakter biner. Sebagai contoh pada Tabel 2 karakter “dimensi *lamina*”.

Tabel 2 Karakter Panjang Lamina

Panjang <i>Lamina</i>	Karakter biner	
Kurang dari 3 cm	0	0
3-40 cm	0	1
Lebih dari 40 cm	1	1

Jumlah kolom yang digunakan adalah jumlah dari skala yang digunakan dikurangi satu. Kenaikan skala tidak harus sama.

### Analisis Klaster

Jika diketahui sebuah himpunan parameter mendeskripsikan suatu objek yang dimiliki oleh beberapa kelas, tujuannya adalah untuk menentukan kelas dari suatu objek dengan hanya melihat parameter-parameternya saja. Hal tersebut dilakukan dengan tujuan kelas atau grup dari objek yang baru dapat ditentukan. Metode untuk melakukan pengelompokan secara sistematis termasuk

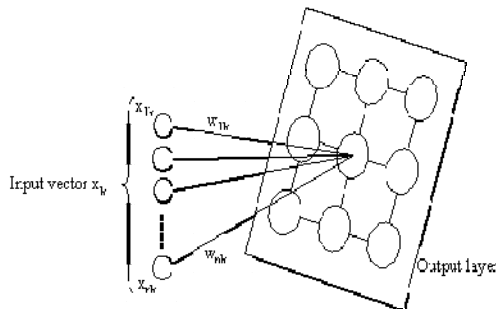
dalam bidang analisis kluster. Metode-metode tersebut menghasilkan aturan-aturan keanggotaan untuk menentukan keanggotaan suatu objek (Matthews, 2002).

Analisis kluster berdasarkan pada ukuran perbandingan antar objek, misalnya jarak Euclid. Jika ukuran jarak antar objek dihitung dengan menggunakan beberapa unit, maka diperlukan konversi ke dalam skala umum.

**Jaringan SOM Kohonen**

*Self-Organizing Map* (SOM) diperkenalkan oleh T. Kohonen pada tahun 1982. SOM merupakan salah satu jaringan syaraf tiruan yang dikonfigurasi untuk pengklasifikasian data. Jaringan SOM Kohonen menggunakan metode pembelajaran *unsupervised* untuk memetakan data yang berdimensi tinggi ke dimensi yang lebih rendah.

Jaringan SOM Kohonen terdiri dari dua lapisan (*layer*), yaitu lapisan *input* dan lapisan *output* (Gambar 2). Setiap *neuron* dalam lapisan *input* terhubung dengan setiap *neuron* pada lapisan *output*. Setiap *neuron* dalam lapisan *output* merepresentasikan kelas dari *input* yang diberikan.



Gambar 2 Struktur jaringan Kohonen.

Setiap *neuron output* mempunyai bobot untuk masing-masing *neuron input*. Proses pembelajaran dilakukan dengan melakukan penyesuaian terhadap setiap bobot pada *neuron output*. Setiap *input* yang diberikan dihitung jarak euclidnya dengan setiap *neuron output* (1), kemudian dilakukan pencarian *neuron output* yang mempunyai jarak minimum. *Neuron* yang mempunyai jarak yang paling kecil disebut *neuron* pemenang.

$$d = \sqrt{\sum_i^n (W_i - X_i)^2} \dots\dots\dots (1)$$

dengan:  
d = jarak euclid  
Wi = bobot *neuron* ke-i

*Xi* = *input* vektor ke-i

Setelah mendapatkan *neuron* pemenang, kemudian memperbaharui nilai bobot *neuron* pemenang dan *neuron* tetangganya (2).

$$W_{ij}(t+1) = W_{ij}(t) + \alpha(t) * h(t) * [X_i(t) - W_{ij}(t)] \dots\dots(2)$$

dengan *Wij* adalah bobot untuk *neuron output* ke-j dan *neuron input* ke-i,  $\alpha(t)$  adalah laju pembelajaran, dan  $h(t)$  adalah fungsi tetangga.

Secara garis besar algoritma SOM Kohonen sebagai berikut:

1. Inisialisasi bobot (*Wij*) dengan nilai acak, laju pembelajaran dan fungsi tetangga.
2. Pilih *input* *Xi* secara acak dari himpunan *input*, kemudian normalisasikan.
3. Hitung tingkat kemiripannya dengan menggunakan jarak Euclid (1) untuk semua *neuron*.
4. Pilih *neuron* pemenangnya, yaitu *neuron* yang mempunyai jarak Euclid paling kecil.
5. Perbaharui bobot *neuron* pemenang (2) dan boto *neuron* tetangganya.
6. Perbaharui laju pembelajaran dan kurangi fungsi tetangga secara linear atau eksponensial.
7. Lakukan langkah 2 sampai 5 hingga tercapai nilai *epoch* (nilai maksimum iterasi).

**Pembelajaran Jaringan**

a. Normalisasi Data

Normalisasi adalah proses transformasi himpunan data secara linear sehingga tiap komponen data memiliki sifat statistik yang sama (Matthews, 2002). Normalisasi terhadap data observasi yang telah dikumpulkan merupakan langkah penting sebelum dilaksanakan proses *clustering* oleh SOM kohonen.

Proses normalisasi yang paling dasar adalah melakukan transformasi data sehingga mempunyai rataian dan standar deviasi nol. Hal tersebut dapat dituliskan sebagai berikut jika  $x = (x_i)$  menyatakan semua observasi dari satu komponen maka rataian (3) dan standar deviasinya (4) adalah:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i \dots\dots\dots(3)$$

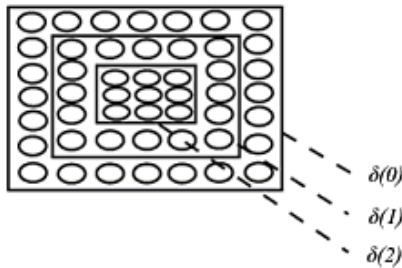
$$\sigma^2 = \frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2 \dots\dots\dots (4)$$

dan transformasi data sebagai berikut:

$$x'_i = \frac{x_i - \bar{x}}{\sigma} \dots\dots\dots (5)$$

dengan  $X_i$  adalah element ke-i dan  $X_i'$  adalah element ke-i yang baru.

b. Fungsi Tetangga



Gambar 3 Bentuk *neuron output* dan tetangga.

Fungsi tetangga adalah derajat pengubahan terhadap bobot *neuron* pemenang dan tetangganya relatif terhadap jarak (6). Lebar tetangga akan berkurang seiring dengan langkah pembelajaran. Bentuk *neuron output* yang digunakan adalah *grid* seperti terlihat pada Gambar 3.

Fungsi tetangga yang digunakan adalah fungsi Gauss, berikut:

$$h(t) = \exp\left(-\frac{\|r_i - r_c\|^2}{2\delta(t)^2}\right) \dots\dots\dots (6)$$

dengan:

$\|r_i - r_c\|^2$  = jarak *neuron* ke-i dengan *neuron* pemenang dalam grid.

$\delta(t)$  = lebar tetangga, berkurang seiring dengan t langkah pembelajaran.

$r_i$  = *neuron* ke-i

$r_c$  = *neuron* pemenang

c. Laju Pembelajaran ( $\alpha$ )

Laju pembelajaran adalah fungsi pengurangan terhadap waktu. Dua metode yang biasa dipakai adalah fungsi linear terhadap waktu (7) dan fungsi yang meng-invers secara proporsional terhadap waktu (8). Fungsi secara linear dapat ditulis:

$$\alpha_t = \alpha_0 - \Delta\alpha \dots\dots\dots(7)$$

sedangkan secara eksponensial sebagai berikut:

$$\alpha_t = \alpha_0 e^{-d*t} \dots\dots\dots(8)$$

dengan:

d = tingkat pengurangan

t = iterasi

**Cluster Recall dan Cluster Precision**

Untuk mengevaluasi hasil dari kluster digunakan perhitungan *cluster recall* (CR) dan *cluster precision* (CP). *Cluster precision* didefinisikan sebagai proporsi antar jumlah data yang terkluster dengan benar dibanding jumlah anggota dalam kluster hasil *clustering*. *Cluster recall* didefinisikan sebagai proporsi jumlah data yang terkluster dengan benar dibandingkan dengan jumlah data dalam kelas yang sebenarnya. *Clustering* yang ideal adalah yang mempunyai nilai CP dan CR mendekati satu. CR (9) dan CP (10) dengan  $n_{ij}$  adalah jumlah anggota kelas i dalam kluster j, sedangkan  $n_i$  adalah jumlah anggota kelas i, dan  $n_j$  jumlah anggota kluster j. Nilai CP dalam penelitian ini digunakan sebagai alat identifikasi.

$$r(i, j) = \frac{n_{ij}}{n_i} \dots\dots\dots(9)$$

$$p(i, j) = \frac{n_{ij}}{n_j} \dots\dots\dots(10)$$

**METODOLOGI**

Kegiatan penelitian ini akan dilakukan dalam empat tahapan utama yaitu:

**Analisis**

Pada tahap analisis dilakukan identifikasi permasalahan sistem yang akan dibuat dan disesuaikan dengan kebutuhan pengguna serta sistem yang sudah ada. Dalam mengidentifikasi sistem, perlu dilakukan penentuan modul-modul yang harus didukung oleh sistem. Dari modul-modul yang telah ditentukan, akan dibuat skenario untuk mempermudah saat menganalisis pergerakan data dan saat mendesain perangkat lunak serta langkah-langkah pengujian.

### Desain Sistem

Dalam tahap ini dilakukan penentuan proses dan data yang diperlukan oleh sistem. Desain sistem menggambarkan keseluruhan kerja sistem mulai dari proses *input* data, pengolahan data, sampai dengan menghasilkan *output*. Desain sistem akan diwujudkan dalam sebuah *flowchart* diagram.

### Implementasi Sistem

Pada tahap ini hasil desain sistem ditranslasikan ke dalam perangkat lunak. Pembangunan sistem akan dipisahkan berdasarkan modul-modul yang telah dianalisis dalam tahap sebelumnya.

### Pengujian

Pada tahap ini dilakukan pengujian terhadap aplikasi yang dibangun. Pengujian menggunakan metode penghitungan *cluster recall* (CR) dan *cluster precision* (CP) secara kuantitatif.

## HASIL DAN PEMBAHASAN

### Analisis

#### a. Karakteristik Data

Data yang diproses bersumber pada basis data Divora, yaitu sistem basis data yang dikelola oleh *World Agroforestry Centre* (ICRAF) yang berlokasi di Bogor. Basis data ini menyimpan spesimen-spesimen daun yang dikumpulkan di hutan karet yang ada di wilayah Jambi dan Lampung.

Data yang berada dalam basis data mempunyai struktur yang dibagi ke dalam beberapa *level*. *Level* bisa merepresentasikan *state* atau subkarakter, setiap karakter dapat mempunyai satu atau lebih subkarakter, dan *level* terdalam adalah 10. Setiap spesimen mempunyai satu atau lebih *morphotype* dan setiap *morphotype* mempunyai karakter dalam bentuk *level* yaitu *field*: ID\_1, ID\_2, dan seterusnya sampai ID\_10. *Level* tersebut kemudian diekstrak ke dalam satu *level* karakter. Proses ekstraksi ini hanya mengambil karakter pada *level* satu yaitu karakter utama.

Data yang digunakan dalam penelitian terdiri dari 127 data spesimen. Setelah dilakukan ekstraksi, diperoleh 21 karakter. Dalam Tabel 3 disajikan tabel karakter untuk salah satu spesimen yang ada dalam basis

data, dengan spesiesnya adalah *Actinodaphne procera*.

Tabel 3 Karakter dan state spesimen dengan spesies *Actinodaphne procera*.

Kriteria	Nilai
Jenis daun berdasarkan ada tidaknya anak daun	Tidak ada
Susunan daun pada batang	Tersebar
Ada tidaknya stipula	Tidak ada
Ada tidaknya stipel	Ada
Helaian daun	Jorong
Ujung daun	Runcing
Pangkal daun	Runcing
Tepi daun	Tidak rata
Ada tidaknya kelenjar daun	Ada
Tangkai daun atau ibu tangkai daun	Ada
Tipe pertulangan daun	Menyirip
Pola urat daun tersier	Menjala
Arah urat daun primer	NA
Ada tidaknya vena intramarginal	Ada
Ada tidaknya duri	Tidak ada
Ada tidaknya gabus	NA
Ada tidaknya getah	Tidak ada
Ada tidaknya rambut	NA
Ada tidaknya bintik	Ada
Ada tidaknya domatia	Tida ada
Paralel tidaknya urat daun sekunder	Tidak

#### b. Kodefikasi

Karakter-karakter spesimen yang telah diekstrak tersebut dilakukan proses kodefikasi. Kodefikasi yaitu perubahan data dari bentuk kualitatif ke dalam bentuk biner, seperti terlihat pada Tabel 4 untuk karakter helaian daun dan Tabel 5 untuk karakter tepi daun. Sedangkan untuk karakter yang berbentuk ya atau tidak cukup dikodekan dengan 1 atau 0, seperti pada karakter ada tidaknya kelenjar daun (Tabel 6).

Tabel 4 Kode karakter biner helaian daun (*shape of the lamina*)

State	Kode				
	X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	X <sub>5</sub>
Bulat memanjang ( <i>oblong</i> )	1	0	0	0	0
Jorong ( <i>elliptic</i> )	0	1	0	0	0
Bulat telur ( <i>ovate</i> )	0	0	1	0	0
Bulat telur	0	0	0	1	0

terbalik ( <i>obovate</i> )					
Bulat ( <i>rounded</i> )	0	0	0	0	1

Tabel 5 Kode karakter biner tepi daun (*type of the margin*)

State	Kode	
	X <sub>1</sub>	X <sub>2</sub>
Rata; halus ( <i>margin smooth</i> )	1	0
Tidak rata ( <i>margin not smooth</i> )	0	1

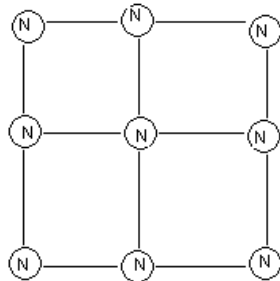
Tabel 6 Kode biner ada tidaknya kelenjar daun

State	Kode
	X <sub>1</sub>
Ya	1
Tidak	0

Sebagai contoh sebuah spesimen daun yang mempunyai karakter; helaian daun bulat memanjang (1,0,0,0), tepi daun rata (0,1), dan ada terdapat kelenjar daun (1), direpresentasikan ke dalam bentuk biner  $X = \{1,0,0,0,0,1,0,1\}$ . Jumlah digit adalah delapan yaitu: lima untuk karakter helaian daun, dua untuk karakter bentuk tepi daun, dan satu buah untuk karakter ada tidaknya kelenjar daun.

### c. Arsitektur Jaringan

Struktur *neuron output* yang digunakan adalah *grid*. Setiap *neuron output* mempunyai 4 tetangga kecuali *neuron output* yang berada di sisi, seperti terlihat pada Gambar 4.



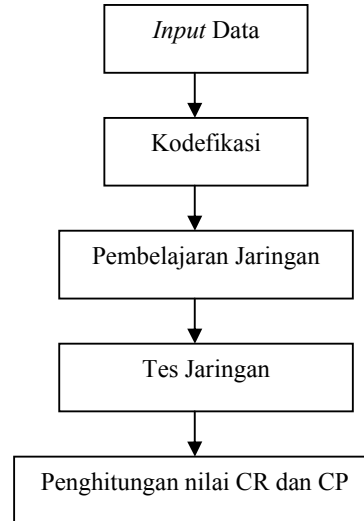
Gambar 4 Struktur *neuron output grid*  $3 \times 3$ .

### d. Pembelajaran Jaringan

Pembelajaran dilakukan dengan menggunakan parameter: jumlah iterasi, alpha, gamma, serta ukuran neuron output yang dicari melalui percobaan. Parameter yang dicari adalah parameter yang menghasilkan nilai CP dan CR maksimum.

## Desain Sistem

*Flowchart* sistem dapat dilihat pada Gambar 5. Langkah pertama adalah proses *input* data dari basis data Divora, kemudian kodefikasi data spesimen ke dalam bentuk biner. Setelah semua data di-*input* dilakukan proses pembelajaran jaringan, kemudian tes jaringan dengan menggunakan data *input*. Langkah terakhir adalah penghitungan nilai CR dan CP untuk masing-masing klaster.



Gambar 5 *Flowchart* sistem.

## Implementasi Sistem

Program dibangun dan dijalankan di dalam Microsoft Access, dengan adanya *Visual Basic for Application* (VBA) maka sangat memungkinkan untuk melakukan pemrograman yang kompleks dalam Microsoft Access.

Setiap spesimen dalam tabel *Specimen* mempunyai *speciesID*, yang merupakan *foreign key* dari table *Species*. *MorphotypeId* adalah key ke dalam tabel *Morphotype* yaitu tabel yang menyimpan karakter-karakter dan *state*-nya untuk spesimen-spesimen yang sudah diidentifikasi. Proses yang terjadi dalam sistem adalah sebagai berikut:

### a. Input data

Input data dalam sistem dilakukan dengan mengekstrak data dari basis data Divora. Parameter-parameter pembelajaran seperti jumlah iterasi, alpha, dan gamma dimasukan melalui *form* dari antar muka utama aplikasi.

### b. Kodefikasi

Dari karakter-karakter hasil ekstraksi dari basis data Divora dilakukan kodefikasi yang kemudian disimpan dalam tabel *MorphpType*.

### c. Pembelajaran jaringan

Pada tahapan ini sistem melakukan pembelajaran jaringan SOM Kohonen. Berikut ini adalah *pseudocode* fungsi pembelajaran yang diimplementasikan dalam sistem.

```
Function Training (Data, N,  
maxStep, numberOfNeuron,  
numberOfWeight, alpha,  
deltaAlpha, gamma, deltagamma)
```

```
For step=1 to maxStep  
  q = Random(1, N)  
  X = Data[q]  
  Normalization(X)  
  din=MAX_INTEGER  
  For j=1 to numberOfNeuron  
    d = Euclid(X,Neuron[j])  
    if (dmin>d) dmin = d  
  Next j  
  winner = Neuron[dmin]  
  
  For i=1 to numberOfNeuron  
    h = Gauss(i,j,gamma)  
    For j=1 to numberOfWeight  
      W(i,j) = W(i,j) +  
      alpha*h*(winner(j)-W(i,j))  
    Next j  
  Next i  
  
  alpha = alpha - deltaAlpha  
  gamma = gamma - deltaGammas  
Next step  
End Function
```

### d. Tes Jaringan

Tes jaringan adalah mencari *neuron output* yang memiliki jarak minimum terhadap spesimen X. Proses tes ini dapat digambarkan dengan *pseudocode* berikut:

```
Function Test(X)  
  dmin = MAX_INTEGER  
  For j=1 to numberOfNeuron  
    d = Euclid(X,Neuron[j])  
    if (dmin>d) dmin = d  
  Next j  
  return dmin  
End Function
```

### e. Penghitungan CP dan CR

Setelah semua data input diuji, dilakukan penghitungan CR untuk semua kelas hasil identifikasi dari ahli, dan CP untuk semua klaster hasil dari jaringan. Penghitungan CR dan CP ini masih dilakukan secara manual dengan menggunakan aplikasi *spreadsheet*,

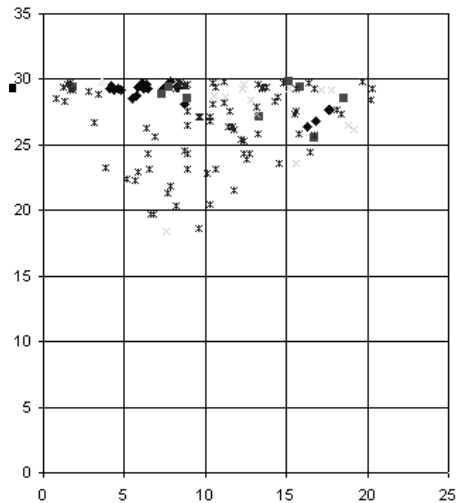
yaitu dengan menyalin tabel distribusi spesimen ke dalam *spreadsheet*.

## Hasil dan Pengujian

Pada tahapan ini dilakukan pencarian parameter jaringan SOM Kohonen dan menguji hasilnya dengan dua parameter yaitu: *cluster recall* (CR) dan *cluster precision* (CP).

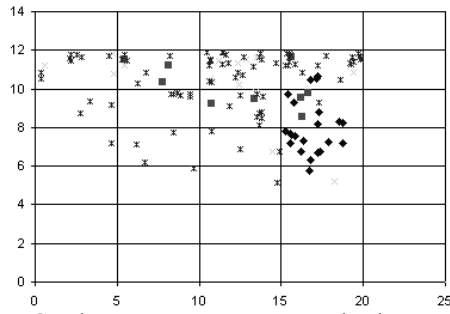
### a. Ukuran Neuron Output

Untuk menentukan ukuran *neuron output* yang sesuai, diambil ukuran *neuron output* sebesar mungkin, jika masih terdapat klaster kosong, maka ukuran *neuron output* diubah menyesuaikan dengan ukuran hasil klaster. Fungsi  $\alpha$  yang digunakan adalah fungsi linear yang berkurang dari 1 sampai 0,01 dan fungsi  $\delta$  digunakan fungsi linear yang berkurang dari 1 sampai 0,01 serta iterasi dilakukan sebanyak 2000 kali.



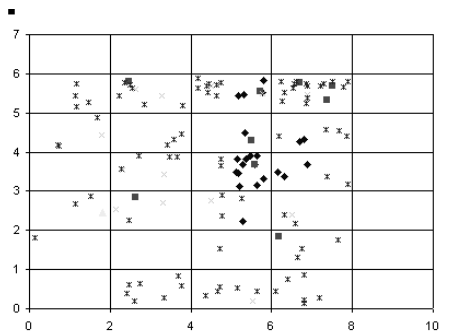
Gambar 6 Pemetaan *output* pada ukuran *neuron output* ukuran  $30 \times 30$ .

Dalam percobaan ini diambil ukuran *neuron output*  $30 \times 30$ , terlihat pada Gambar 6, data menyebar di daerah atas dan terdapat daerah yang kosong, berdasarkan hasil di atas maka kurangi ukuran *neuron output* menjadi  $20 \times 12$  dengan mengurangi daerah kosong, hasilnya seperti pada Gambar 7.



Gambar 7 Pemetaan *output* pada ukuran *neuron output* ukuran  $20 \times 12$

Setelah dilakukan percobaan sebanyak empat kali, diperoleh ukuran *neuron output*  $8 \times 6$  seperti terlihat pada Gambar 8.



Gambar 8 Pemetaan *output* pada ukuran *neuron output* ukuran  $8 \times 6$ .

#### b. Jumlah Iterasi Pembelajaran

Untuk menentukan jumlah iterasi yang digunakan dalam penelitian ini dicari dengan menggunakan jumlah iterasi secara variabel, kemudian dicari jumlah iterasi pembelajaran yang menghasilkan nilai CP dan CR maksimum. Ukuran *neuron output* yang digunakan adalah  $8 \times 6$  dan fungsi  $\alpha$  yang digunakan adalah fungsi linear yang berkurang dari 1 sampai 0,01 serta fungsi  $\delta$  digunakan fungsi linear yang berkurang dari 1 sampai 0,01.

Tabel 7 menyajikan hasil dari percobaan, jumlah iterasi tidak terlalu mempengaruhi nilai CP dan CR. Pembelajaran dengan iterasi sebanyak 2000 kali didapat nilai CP yang paling tinggi, sedangkan pada iterasi sebanyak 6000 kali didapat nilai CR paling tinggi. Selanjutnya pembelajaran dilakukan dengan iterasi sebanyak 2000 kali, karena mempunyai nilai CP yang paling tinggi.

Tabel 7 Pengaruh jumlah iterasi terhadap jumlah *neuron output* yang terisi (N), rata-rata CR (CR) dan rata-rata CP (CP).

Iterasi	N	CR	CP
500	26	0.454936	0.43139
1000	36	0.403308	0.5395
2000	39	0.402192	0.59323
3000	33	0.468281	0.55813
4000	33	0.449693	0.56164
5000	37	0.380637	0.57119
6000	29	0.489879	0.57463

#### c. Nilai $\alpha$ dan $\delta$

Dalam penentuan nilai  $\alpha$  dan  $\delta$  yang dilakukan secara acak, seperti halnya dalam penentuan ukuran *neuron output* dan jumlah iterasi, digunakan ukuran *neuron output*  $8 \times 6$  dan jumlah iterasi sebanyak 2000 kali. Hasil dari percobaan dapat dilihat dalam Tabel 8.

Seperti terlihat dalam Tabel 8, semakin kecil nilai  $\delta$  maka jumlah *neuron* yang terisi (jumlah kluster) semakin kecil dan nilai rata-rata CR semakin besar. Hal tersebut disebabkan oleh lebar tetangga *neuron* pemenang yang mengecil seiring dengan pengurangan nilai  $\delta$ , sehingga kluster mengumpul pada satu area. Pada tabel tersebut juga dapat dilihat bahwa dengan semakin kecil kluster yang dihasilkan maka nilai CR semakin besar, tetapi mempunyai nilai CP yang kecil.

Tabel 8 Pengaruh nilai  $\alpha$  dan  $\delta$  terhadap jumlah *neuron output* yang terisi (N), rata-rata CP (CP) dan rata-rata CR (CR)

$\alpha$	$\delta$	N	CR	CP
1	1	39	0.402192	0.59323
0.75	1	34	0.427596695	0.574350649
0.5	1	32	0.452634609	0.458585859
1	0.75	36	0.451048361	0.587844353
0.75	0.75	34	0.433623234	0.503041781
0.5	0.75	31	0.43491983	0.484512331
1	0.5	31	0.512508452	0.592857143
0.75	0.5	18	0.51127451	0.364575954
0.5	0.5	16	0.67037037	0.352123508

Berdasarkan hasil percobaan nilai  $\alpha$  mempengaruhi nilai CP. Nilai  $\alpha$  berbanding lurus dengan nilai CP, hal ini dapat terlihat dengan semakin besar nilai  $\alpha$  maka semakin besar nilai CP. Nilai  $\alpha$  mempengaruhi kecepatan proses pembelajaran, semakin besar nilai  $\alpha$  maka proses pembelajaran semakin



lambat tetapi menghasilkan *neuron* yang stabil, sehingga menghasilkan CP yang tinggi.

Jaringan SOM Kohonen menghasilkan CP terbesar pada nilai  $\alpha$  dan  $\delta$  yang maksimum dengan nilai 1 untuk  $\alpha$  dan 1 untuk  $\delta$ . Percobaan selanjutnya menggunakan nilai 1 untuk  $\alpha$  dan 1 untuk  $\delta$ .

Setelah mendapatkan parameter jaringan, yaitu: ukuran *neuron output*, jumlah iterasi, serta nilai  $\alpha$  dan  $\delta$ , langkah berikutnya adalah membuat tabel distribusi spesimen terhadap kluster. Tabel distribusi tersebut dibuat menggunakan parameter-parameter SOM Kohonen seperti terlihat pada Tabel 9. Tingkat famili dipilih karena pada tingkat yang lebih rendah yaitu spesies ataupun *genus* jumlah tiap spesimen tidak merata.

Tabel 9 Parameter-parameter jaringan SOM Kohonen yang digunakan

Parameter	Nilai
Ukuran <i>neuron output</i>	$8 \times 6$
Jumlah iterasi	2000
Nilai $\alpha$	1
Nilai $\delta$	1

*Cluster recall* (CR) yaitu distribusi suatu kelas (famili) ke dalam kluster, menghasilkan rata-rata CR sangat kecil yaitu 0,4297019. Berdasarkan data tersebut hanya ada 2 famili yang mempunyai nilai CR besar yaitu *Elaeocarpaceae* (0,75) dan *Myrtaceae* (0,8). Famili lainnya yang memiliki CR kecil disebabkan karena tumpang tindihnya morfologi antar famili.

*Cluster Precision* (CP) dan menghasilkan rata-rata CP 0,5304834. Hanya beberapa kluster yang diisi tepat oleh sebuah famili seperti pada kluster 3, 8, 9, 16, 20, 32, 33, 36 dan 40 yang tepat diisi oleh satu famili.

## KESIMPULAN DAN SARAN

### Kesimpulan

Berdasarkan hasil percobaan, diperoleh beberapa kesimpulan sebagai berikut:

1. Ukuran *output* SOM tidak mempengaruhi kluster data.
2. Nilai  $\delta$  akan mempengaruhi struktur *neuron output*, makin besar nilai  $\delta$  maka makin banyak *neuron output* yang terisi.
3. Nilai CR dipengaruhi oleh jumlah kluster yang terbentuk, nilai CR berbanding

terbalik dengan jumlah kluster yang terbentuk.

4. Nilai  $\alpha$  akan mempengaruhi nilai CP, nilai  $\alpha$  berbanding lurus dengan nilai CP.
5. Tidak semua spesimen dalam satu famili terkluster tepat pada satu kluster.
6. Dari hasil perhitungan *cluster recall* dan *cluster precision* yang masing-masing menghasilkan 0.402192 dan 0.59323 SOM kohonen tidak menghasilkan kluster yang ideal.

### Saran

Sistem yang dibangun masih belum sempurna, oleh karena itu masih dibutuhkan pengembangan lebih lanjut. Untuk penelitian selanjutnya, dapat dikembangkan suatu sistem yang menggunakan metode ukuran jarak lain selain Euclid, khususnya untuk data kualitatif, atau dapat menggunakan metode selain SOM Kohonen.

## DAFTAR PUSTAKA

- Dalli, A.**, *Adaptation of the F-measure to Cluster Based Lexicon Quality Evaluation*. Department of Computer Science, University of Sheffield.
- Dunn, G., B.S. Everit.** 1982. *An Introduction to Mathematical Taxonomy*. Cambridge University Press.
- Foody, G.M.** 1999. *Applications of the self-organizing feature map neural network in community data analysis*. Ecological Modelling 1999;120:97–107.
- Limin Fu.** 1994. *Neural Network in Computer Intelligence*. McGraw-Hill.
- Matthews, P.C.** 2002. *The Application of Self Organizing Maps in Conceptual Design* [dissertation]. Cambridge University Engineering Department.
- Mu-Chun Su, Hsiao-Te Chang.** 2000. *Fast Self-Organizing Feature Map Algorithm*. IEEE Transactions on Neural Networks 2000:Vol. 11, No. 3.
- Rasnovi, S.** 2001. *Kajian Pemakaian Morfologi Daun untuk Identifikasi Jenis pada Beberapa Famili Dokotiledon Berhabitus Pohon di Sumatra* [tesis]. Bogor: Program Pasca Sarjana IPB.



