

## **BAB II**

### **KAJIAN TEORI**

Bab II berisi tentang kajian teori yang diantaranya mengenai Kanker Paru, Pengolahan Citra Digital, Ekstraksi Fitur Citra, *Artificial Neural Network*, SOM Kohonen *clustering* dan Ketepatan Hasil Klasifikasi.

#### **A. Kanker Paru**

Berikut adalah hal-hal mengenai Kanker Paru:

##### **1. Pengertian Kanker Paru**

Kanker paru adalah pertumbuhan sel yang tidak normal, tidak terbatas, dan merusak sel-sel jaringan yang normal pada salah satu atau kedua paru. Tumor ganas paru berasal dari saluran napas atau epitel bronkus. Sel kanker dimulai dari mutasi yang terjadi pada DNA sel. Mutasi DNA disebabkan oleh proses penuaan dini atau adanya faktor lingkungan seperti asap rokok, polusi udara, hingga terpapar zat karsinogenik (arsen, asbestos, radon, kadmium) (Kemenkes RI, 2016).

*World Health Organization* (WHO) berpendapat bahwa kanker paru merupakan jenis kanker terbanyak pada laki-laki di Indonesia, dan terbanyak kelima untuk semua jenis kanker pada perempuan. Kanker paru juga merupakan penyebab kematian akibat kanker terbanyak pada laki-laki dan kedua pada perempuan (*Global Burden Cancer*, 2012).

## 2. Jenis Kanker Paru

Terdapat dua jenis kanker paru, yaitu (*American Cancer Society, 2017*):

### a. *Small Cell Lung Cancer* (SCLC)

SCLC adalah jenis kanker paru yang tumbuh lebih cepat daripada jenis kanker *Non Small Cell Lung Cancer* (NSCLC), akan tetapi pertumbuhan SCLC lebih dapat terkendali dengan kemoterapi. Sekitar 20% kasus kanker paru adalah SCLC, atau sekitar 30.000 pasien setiap tahunnya terdiagnosis penyakit tersebut.

### b. *Non Small Cell Lung Cancer* (NSCLC)

Sekitar 75%-80% kasus kanker paru adalah NSCLC. Terdapat 3 tipe NSCLC, yaitu:

#### 1) Adenokarsinoma

Adenokarsinoma adalah jenis dari NSCLC yang paling umum dari kanker paru dan lebih banyak muncul pada wanita. Kanker tipe ini berkembang dari sel-sel yang memproduksi lendir pada permukaan saluran udara.

#### 2) Karsinoma skuamosa

Jenis ini paling umum dari kanker paru serta paling banyak terjadi pada pria dan orang tua. Karsinoma skuamosa berkembang dalam sel yang mengisi saluran udara, dan kanker ini tumbuh relatif lambat.

#### 3) Karsinoma sel besar

Pertama kali muncul biasanya di saluran pernapasan yang lebih kecil dan dapat menyebar dengan cepat. Tipe ini sering disebut juga karsinoma tidak berdiferensiasi karena bentuk sel kanker ini bundar besar.

### **3. Faktor Risiko Kanker Paru**

Seperti umumnya penyakit kanker yang lain, penyebab yang pasti dari kanker paru belum diketahui. Namun, faktor risiko seseorang mengalami kanker paru sangat berhubungan erat dengan rokok atau paparan zat yang bersifat karsinogenik. Berikut ini faktor risiko penyebab terjadinya kanker paru (*American Cancer Society*, 2017):

#### **a. Merokok**

Merokok merupakan faktor yang berperan paling penting. Sekitar 80% kematian kanker paru disebabkan dari kebiasaan merokok. Risiko seorang perokok aktif lebih tinggi dibanding perokok pasif. Kejadian kanker paru pada perokok dipengaruhi oleh lamanya kebiasaan merokok, jumlah rokok yang dihisap setiap hari dan lamanya berhenti merokok.

#### **b. Polusi Udara**

Polusi udara menyebabkan risiko kanker paru terutama di daerah perkotaan yang padat kendaraan. Namun, risiko kanker paru yang ditimbulkan lebih rendah dibandingkan dengan kebiasaan merokok. Pada beberapa penelitian, diperkirakan sekitar 5% kematian kanker paru diseluruh dunia disebabkan dari polusi udara.

#### **c. Paparan zat karsinogenik**

Beberapa zat karsinogenik seperti radon, asbestos, arsen, uranium, kadmium, nikel, vinil klorida, kromium dapat menyebabkan kanker paru. Zat-zat karsinogenik tersebut dapat ditemui di lingkungan kerja seperti kawasan pabrik industri dan pertambangan.

d. Genetika

Faktor risiko penyebab kanker paru juga dapat disebabkan oleh genetika. Apabila keluarga memiliki riwayat kanker paru dapat meningkatkan risiko terkena kanker paru, namun belum begitu jelas seberapa besar tingkat risiko yang dialami.

#### **4. Gejala Kanker Paru**

Gambaran penyakit kanker paru terdiri dari keluhan subjektif dan gejala objektif. Keluhan utama dapat berupa batuk-batuk atau tanpa dahak, batuk darah, sesak napas, suara serak, sakit dada, sulit menelan, dan terdapat benjolan di pangkal leher. Gejala atau keluhan akibat metastasis di luar paru, seperti kelainan yang timbul karena kompresi hebat di otak, pembesaran hepar, dan berat badan berkurang juga merupakan ciri dari adanya kanker paru. (Perhimpunan Dokter Paru Indonesia, 2003: 3)

#### **5. Deteksi Dini Kanker Paru**

Menurut *Japanese Society of Radiology Technology*, nodul atau benjolan yang terdeteksi pada paru dikategorikan menjadi dua yaitu *non cancerous nodule* (tumor jinak) dan *cancerous nodule* (tumor ganas). Tumor jinak yang terdapat pada jaringan paru tidak akan menyerang selain organ paru karena tumor jinak hanya menyerang satu tempat dan tidak menyebar ke organ tubuh lainnya. Gejala awal kanker paru seperti batuk, demam, suara serak dan lain-lain dapat dengan mudah disalah artikan sebagai penyakit flu atau gejala pada orang merokok. sehingga banyak kanker paru pada stadium awal sulit dideteksi (*Japanese Society of Radiology Technology*, 1997).

Berikut ini beberapa cara deteksi dini kanker paru (*Modern Cancer Hospital Guangzhou, 2015*):

a. *X-Ray*

*X-Ray* adalah metode penting dalam pendeteksian kanker paru, sebagian besar kanker paru terdeteksi saat melakukan *X-Ray* dibagian dada. Melalui pemeriksaan ini, dapat diketahui ada tidaknya tumor, letak dan ukuran tumor, serta melihat perubahan inflamasi paru yang disebabkan hambatan tumor pada bronkus yang menimbulkan emfisema lokal atau lesi sekitar.

b. Sitologi dahak

Pemeriksaan sitologi dahak dapat menentukan jenis patologi sel kanker dan kanker paru. Metode ini memiliki tingkat keakuratan lebih dari 80%. Secara klinis, pada pasien yang memiliki risiko lebih besar dapat melakukan sitologi dahak secara berulang.

c. Bronkoskopi

Pengklasifikasian melalui bronkoskopi terhadap pasien kanker paru cenderung lebih tinggi. Melalui cara ini dapat dilihat secara langsung tumor dibagian rongga bronkus, serta mengambil jaringan kecil untuk dilakukan biopsi.

d. Biopsi jaringan

Biopsi dilakukan pada pasien kanker paru yang telah memasuki stadium lanjut, misalnya tumor sudah menyebar ke tulang selangka, bagian leher, ketiak dan kelenjar getah bening atau muncul nodul pada bagian bawah kulit.

## B. Pengolahan Citra Digital

Citra digital adalah representasi dari citra dua dimensi sebagai sebuah kumpulan nilai digital yang disebut elemen gambar atau *pixel*. *Pixel* adalah istilah yang sering digunakan untuk menunjukkan unsur-unsur pada citra digital tersebut (Gonzales & Woods, 2008: 1).

Matriks yang dibentuk oleh citra digital dua dimensi berukuran  $e \times g$ , dengan  $e$  adalah jumlah baris dan  $g$  adalah jumlah kolom ditunjukkan sebagai berikut,

$$f(a,b) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,g-1) \\ f(1,0) & f(1,1) & \dots & f(1,g-1) \\ \vdots & \vdots & & \vdots \\ f(e-1,0) & f(e-1,1) & \dots & f(e-1,g-1) \end{bmatrix} \quad (2.1)$$

Indeks baris ( $a$ ) dan indeks kolom ( $b$ ) menyatakan suatu koordinat titik pada citra, sedangkan  $f(a,b)$  merupakan intensitas atau derajat keabuan pada titik ( $a,b$ ). Elemen pada matriks merupakan elemen citra digital yang disebut dengan *pixel*.

Suatu *pixel* memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0-255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra integer (Putra, 2010: 20). Citra digital terbagi menjadi beberapa jenis. Jenis citra digital yang sering digunakan adalah citra biner, citra *grayscale*, dan citra warna. Adapun penjelasan mengenai jenis citra adalah sebagai berikut (Usman, 2005: 14):

## 1. Citra biner

Citra biner adalah citra digital yang memiliki dua kemungkinan nilai *pixel* yaitu hitam yang dinyatakan dengan 0 dan putih yang dinyatakan dengan 1. Nilai 0 (hitam) adalah *background points*, biasanya bukan merupakan bagian dari citra sesungguhnya. Sedangkan nilai 1 (putih) adalah *region points*, yaitu bagian dari citra sebenarnya (bukan latar belakang). Citra biner juga disebut B&W (*black and white*) atau monokrom. Proses pemineranan dilakukan dengan membulatkan ke atas atau ke bawah untuk setiap nilai keabuan dari *pixel* yang berada di atas atau di bawah nilai batas.



**Gambar 2.1** Citra Biner

## 2. Citra grayscale

Citra *grayscale* atau abu – abu merupakan citra yang mampu menghasilkan gradasi warna abu-abu dari warna hitam hingga warna putih. Tingkat keabuan disini merupakan warna abu dengan berbagai tingkatan, berupa bilangan antara 0 s.d 255. Nilai ini digunakan untuk menunjukkan nilai intensitas. Nilai 0 untuk warna hitam, nilai 255 untuk warna putih dan nilai antara 0 sd 255 untuk warna antara hitam dan putih (keabuan).



**Gambar 2.2** Citra *Grayscale*

### 3. Citra warna

Citra warna atau yang biasa disebut dengan citra RGB adalah citra digital yang setiap *pixel*nya mewakili warna kombinasi dari tiga warna dasar yaitu R = *Red*, G = *Green*, dan B = *Blue*. Citra warna disebut juga *true color* karena mempunyai jumlah warna yang cukup besar yaitu mencapai 16 juta warna.



**Gambar 2.3** Citra Warna

#### C. Ekstraksi Fitur Citra

Salah satu teknik ekstraksi citra adalah GLCM. GLCM banyak digunakan dalam klasifikasi citra dan fitur-fitur yang diperoleh dari GLCM dapat membantu memahami rincian gambar secara keseluruhan dalam hal tekstur (Gadkari, 2004).

Ekstraksi citra yang dilakukan dengan metode GLCM dapat menghasilkan 13 fitur ekstraksi yaitu *contrast*, *correlation*, *energy*, *homogeneity*, *sum entropy*, *sum of square variance*, *invers difference moment*, *sum average*, *sum variance*, *entropy*, *difference entropy*, *maximum probability*, dan *dissimilarity entropy*.

##### 1. Contrast

*Contrast* merupakan perbedaan intensitas antara nilai tertinggi (terang) dan nilai terendah (gelap) dari *pixel* yang saling berdekatan. Suatu *contrast* merupakan ukuran variasi antar derajat keabuan dari suatu daerah citra (Gadkari, 2014: 13).

Rumus *contrast* adalah sebagai berikut (Kalas, 2010: 19):

$$\text{Contrast} = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} h(e, g)(e - g)^2 \quad (2.2)$$



dengan

$h(e, g)$  = menunjukkan *pixel* pada baris ke- $e$  dan kolom ke- $g$  dengan  $e = 1, 2, \dots, Ng$  ;  $g = 1, 2, \dots, Ng$ ,

$N_g$  = banyak derajat keabuan (*grayscale*) yang diperoleh dari citra.

Nilai *contrast* membesar jika dipengaruhi oleh variasi intensitas dalam citra tinggi.

Nilai *contrast* semakin kecil apabila variasi rendah.

## 2. Correlation

*Correlation* suatu citra menunjukkan ukuran ketergantungan linear derajat keabuan (*grayscale*) citra, sehingga dapat menunjukkan adanya struktur linear dalam citra yang dirumuskan sebagai berikut (Mohanaiah, *et al*, 2013: 2):

$$Correlation = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} \frac{\{(e, g)h(e, g)\} - \mu_a \mu_b}{\sigma_a \sigma_b} \quad (2.3)$$

dengan

$$\mu_a = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} \{(g)h(e, g)\}$$

(nilai rata-rata elemen baris pada *histogram* citra),

$$\mu_b = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} \{(e)h(e, g)\}$$

(nilai rata-rata elemen kolom pada *histogram* citra),

$$\sigma_a = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} \{(g - \mu_a)^2 h(e, g)\}$$

(simpangan baku elemen baris pada *histogram* citra),

$$\sigma_b = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} \{(e - \mu_b)^2 h(e, g)\}$$

(simpangan baku elemen kolom pada *histogram* citra).

### 3. *Energy*

*Energy* adalah fitur yang bekerja dengan mengukur konsentrasi pasangan intensitas pada matriks konkurensi. Rumus *energy* adalah sebagai berikut (Sharma & Mukharjee, 2013: 331):

$$Energy = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} \{h(e, g)\}^2 \quad (2.4)$$

Nilai energi semakin besar apabila *pixel* yang memenuhi syarat matriks intensitas berkumpul pada beberapa koordinat dan mengecil apabila letaknya menyebar.

### 4. *Homogeneity*

*Homogeneity* memberikan nilai yang merupakan ukuran kedekatan dari distribusi elemen di GLCM ke diagonal GLCM dan dirumuskan sebagai berikut (Sharma & Mukharjee, 2013: 331):

$$Homogeneity = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} \frac{h(e, g)}{1+(e-g)^2} \quad (2.5)$$

### 5. *Sum entropy*

*Sum entropy* adalah fitur yang menunjukkan seberapa banyak derajat keabuan (*grayscale*) yang acak. Rumus *sum entropy* adalah sebagai berikut (Haralick, *et al*, 1973: 619):

$$Sum\ entropy = - \sum_{s=2}^{2Ng} h_{a+b(s)} \log \{h_{a+b(s)}\} \quad (2.6)$$

dengan

$$h_{a+b(s)} = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} h(e, g); e + g = s; s = 2, 3, \dots, 2Ng.$$

## 6. *Sum of square (variance)*

*Variance* adalah ukuran statistik yang mengukur tingkat keragaman suatu *pixel* pada citra. Rumus *sum of square variance* adalah sebagai berikut (Anami & Burkpalli, 2009: 11):

$$\text{Variance} = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} h(e, g)(e - \mu)^2 \quad (2.7)$$

dengan

$$\mu = \text{rata-rata} \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} h(e, g).$$

## 7. *Inverse Difference Moment*

IDM adalah ukuran dari homogenitas lokal. Nilai *Inverse Difference Moment* tinggi ketika derajat keabuan (*grayscale*) lokal seragam dan invers dari GLCM tinggi (Mohanaiah, *et al*, 2013: 2). Rumus IDM adalah sebagai berikut (Sharma & Mukharjee, 2013: 331):

$$\text{Inverse Difference Moment} = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} \frac{h(e, g)}{1+(e-g)^2} \quad (2.8)$$

## 8. *Sum average*

*Sum average* adalah fitur yang menunjukkan seberapa banyak nilai rata-rata *pixel* yang ada dalam citra. Rumus *sum average* adalah sebagai berikut (Haralick, *et al*, 1973: 619):

$$\text{Sum average} = - \sum_{s=2}^{2Ng} \left\{ (s) \left( h_{a+b(s)} \right) \right\} \quad (2.9)$$

## 9. *Sum variance*

*Sum variance* menunjukkan seberapa banyak level keabu-abuan yang bervariasi dari nilai rata-rata (Sharma & Mukharjee, 2013: 331). Rumus *sum variance* adalah sebagai berikut (Haralick, *et al*, 1973: 619):

$$\text{Sum variance} = \sum_{s=2}^{2N_g} (e - SE)^2 h_{a+b(s)} \quad (2.10)$$

dengan

$SE$  = *sum entropy*.

## 10. *Entropy*

*Entropy* adalah fitur untuk mengukur ketidakteraturan dari distribusi intensitas. *Entropy* menunjukkan jumlah informasi dari sebuah citra yang dibutuhkan untuk mengompres (memampatkan) citra (Mohanaiah, *et al*, 2013: 2). Rumus *entropy* dari suatu citra adalah sebagai berikut (Haralick, *et al*, 1973: 619):

$$\text{Entropy} = - \sum_{e=1}^{N_g} \sum_{g=1}^{N_g} h(e, g) \log_2 \{h(e, g)\} \quad (2.11)$$

## 11. *Difference entropy*

*Difference entropy* adalah fitur yang menunjukkan ketidakteraturan dalam suatu citra. Rumus *difference entropy* adalah sebagai berikut (Haralick, *et al*, 1973: 619):

$$\text{Difference entropy} = - \sum_{q=0}^{N_g-1} (h_{a-b(q)}) \{ \log (h_{a-b(q)}) \} \quad (2.12)$$

dengan

$$h_{a-b(q)} = \sum_{e=1}^{N_g} \sum_{g=1}^{N_g} h(e, g); |e - g| = q; q = 0, 1, \dots, (N_g - 1).$$

## 12. *Maximum probability*

*Maximum probability* menunjukkan derajat keabuan (*grayscale*) yang memenuhi relasi pada persamaan *entropy* dan dirumuskan sebagai berikut (Anami & Burkpalli, 2009: 11):

$$\text{Maximum probability} = \max_{e,g} \{h(e, g)\} \quad (2.13)$$

## 13. *Dissimilarity*

*Dissimilarity* menunjukkan perbedaan tiap *pixel* dan dirumuskan sebagai berikut (Anami & Burkpalli, 2009: 11):

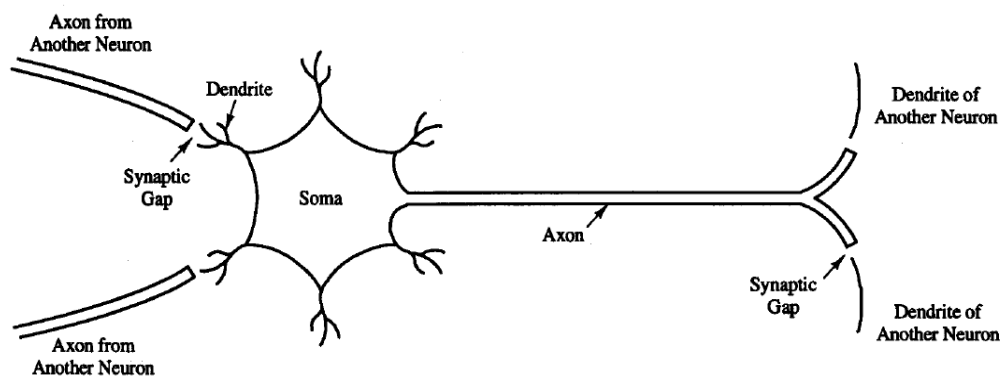
$$\text{Dissimilarity} = \sum_{e=1}^{Ng} \sum_{g=1}^{Ng} h(e, g) |e - g| \quad (2.14)$$

## D. **Artificial Neural Network (ANN)**

*Artificial neural network* sederhana diperkenalkan oleh McCulloch dan Pitts di tahun 1943 yang menyimpulkan bahwa kombinasi beberapa neuron sederhana menjadi sebuah sistem neural akan meningkatkan kemampuan komputasinya. Bobot dalam jaringan yang diusulkan oleh McCulloch dan Pitts diatur untuk melakukan fungsi logika sederhana. Fungsi aktivasi yang digunakan adalah fungsi *threshold*. Tahun 1958, Rosenblatt memperkenalkan dan mulai mengembangkan model jaringan yang disebut *perceptron*. Metode pelatihan diperkenalkan untuk mengoptimalkan hasil iterasinya. Widrow dan Hoff di tahun 1960 mengembangkan *perceptron* dengan memperkenalkan aturan pelatihan jaringan yang dikenal sebagai aturan *delta* (kuadrat rata-rata terkecil). Peneliti terdahulu hanya menggunakan jaringan dengan lapisan tunggal. Rumelhart di tahun 1986 mengembangkan *perceptron* menjadi *backpropagation*, yang memungkinkan jaringan diproses melalui beberapa lapisan. Selain itu, beberapa model jaringan saraf tiruan lain juga

dikembangkan oleh Kohonen di tahun 1972, Hopfield di tahun 1982 (Siang, 2009:4).

Suatu *artificial neural network* ditandai dengan (1) arsitektur yaitu pola dari hubungan antar neuron, (2) algoritma pembelajaran yaitu metode untuk menentukan bobot pada hubungan neuron dan (3) fungsi aktivasi (Fausett, 1994: 3). Sebuah neuron memiliki tiga bagian utama yaitu sel tubuh atau soma (dimana nukleus/inti sel terletak), dendrit, dan akson (gambar 2.4). Dendrit adalah sambungan tunggal berbentuk silinder yang memanjang dari sel tubuh dan membawa impuls (sinyal) dari neuron (Lin & Lee, 1995: 206). Sinyal tersebut berupa impuls elektrik yang dikirim melalui celah sinaptik melalui proses kimiawi. Sinyal yang masuk dimodifikasi (diperkuat/ diperlemah). Selanjutnya, soma atau sel tubuh menjumlahkan semua sinyal yang masuk. Apabila sinyal yang diterima cukup kuat, maka sinyal akan diteruskan ke sel lain melalui akson (Fausett, 1994: 5).



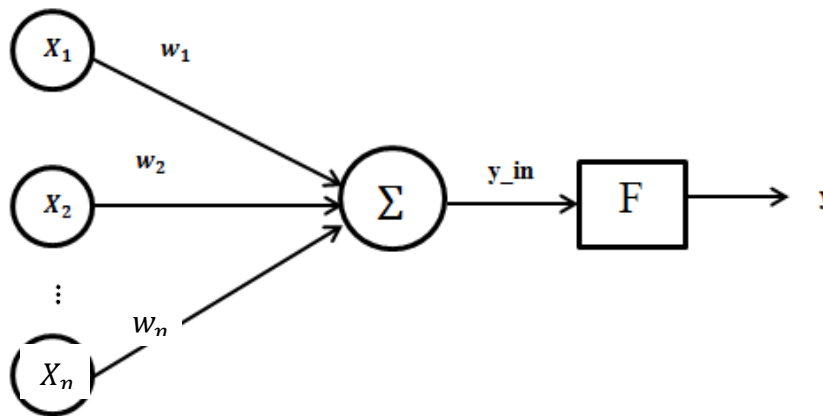
**Gambar 2.4** Jaringan Saraf Biologi

*Artificial neural network* mempunyai karakteristik sebagai berikut (Fausett, 1994:3):

- 1) Pola hubungan antar neuron (disebut arsitektur jaringan),
- 2) Metode untuk menentukan bobot pada penghubung (pelatihan atau pembelajaran, algoritma),
- 3) Fungsi aktivasi.

Gambar 2.5 menunjukkan *artificial neural network* sederhana dengan fungsi aktivasi ( $F$ ). Sebuah neuron akan mengolah ( $x_i$ ) input ( $i = 1, 2, \dots, n$ ) yang masing-masing memiliki bobot ( $w_i$ ), kemudian fungsi aktivasi ( $F$ ) akan mengaktivasi ( $y_{in}$ ) menjadi output jaringan ( $y$ ), dengan rumus (Kusumadewi & Hartati, 2010: 72):

$$y_{in} = \sum_{i=1}^n x_i w_i, \quad (2.15)$$



**Gambar 2.5** Artificial Neural Network Sederhana

Jika *artificial neural network* pada lapisan output sebanyak ( $y_j$ ) neuron ( $j = 1, 2, \dots, m$ ), maka proses pengolahan data neuron ke- $j$  adalah sebagai berikut.

$$y_{in_j} = \sum_{i=1}^n \sum_{j=1}^m x_i w_{ij}, \quad (2.16)$$

dengan ( $w_{ij}$ ) adalah bobot yang menghubungkan input ke- $i$  menuju neuron ke- $j$ .

Dalam *artificial neural network*, informasi yang disebut dengan input akan dikirim ke neuron dengan bobot tertentu. Input ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot. Hasil penjumlahan ini kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila input tersebut melewati suatu ambang tertentu, maka neuron tersebut akan diaktifkan, tetapi jika tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya, dan demikian seterusnya.

Pada *artificial neural network*, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (lapisan) yang disebut dengan lapisan neuron (neuron layers). Neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan input dan output). Informasi yang diberikan pada *artificial neural network* akan dirambatkan dari lapisan ke lapisan, mulai dari lapisan input sampai ke lapisan output melalui lapisan yang lainnya, yang disebut lapisan tersembunyi (tersembunyi lapisan). Arah perambatan ini sesuai dengan algoritma pembelajaran yang digunakan (Kusumadewi & Hartati, 2010: 69-72). Menurut Fausett (1994:3) karakteristik dari *artificial neural network* ditentukan oleh beberapa hal, yaitu: arsitektur, fungsi aktivasi, dan *learning algorithm* (algoritma pembelajaran).



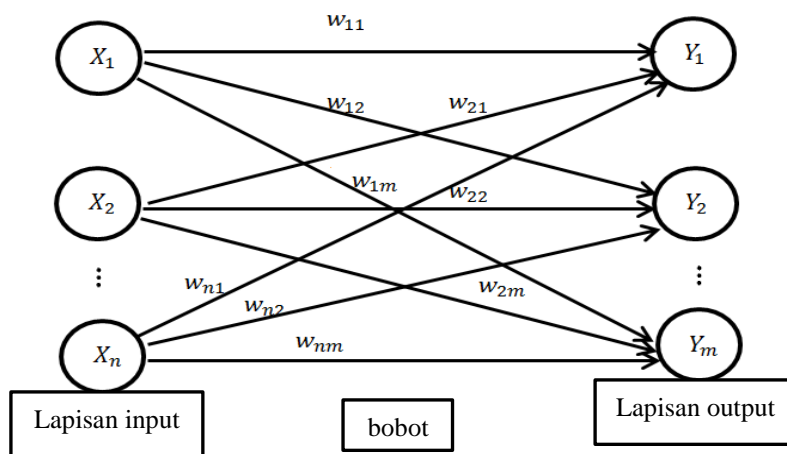
## 1. Arsitektur ANN

Hubungan antar neuron dalam *artificial neural network* mengikuti pola tertentu tergantung pada arsitektur jaringan sarafnya. Menurut Fausett (1994: 12-15) terdapat 3 arsitektur dalam *artificial neural network*, antara lain:

### a. Jaringan Lapisan Tunggal (*single lapisan network*)

Dalam jaringan ini, sekumpulan input neuron dihubungkan langsung dengan sekumpulan outputnya. Dalam beberapa model (misal *perceptron*), hanya ada sebuah neuron output.

Gambar 2.6 menunjukkan arsitektur jaringan dengan ( $n$ ) neuron input ( $x_1, x_2, \dots, x_n$ ) dan ( $m$ ) neuron output ( $y_1, y_2, \dots, y_m$ ). Dalam jaringan ini, semua neuron input dihubungkan dengan semua neuron output, meskipun dengan bobot yang berbeda-beda. Tidak ada neuron input yang dihubungkan dengan neuron input lainnya. Demikian pula dengan neuron output.

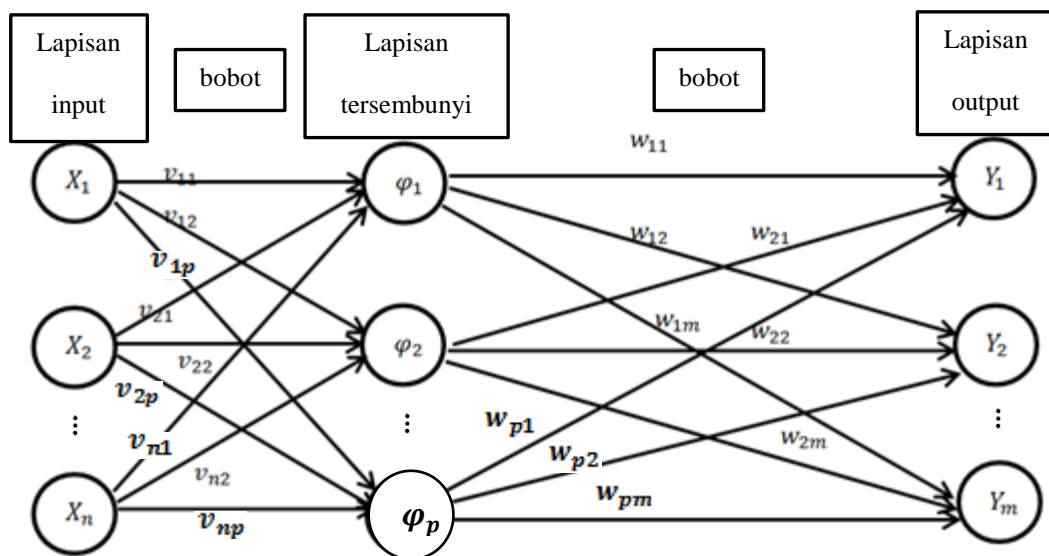


**Gambar 2.6** Jaringan Lapisan Tunggal

b. Jaringan Lapisan Jamak (*multilapisan network*)

Jaringan lapisan jamak merupakan jaringan dengan satu lapisan simpul atau lebih (disebut tersembunyi neuron atau neuron tersembunyi) antara neuron input dan neuron output. Terdapat lapisan bobot antara dua tingkat neuron yang berdekatan (input, tersembunyi, output).

Gambar 2.7 adalah jaringan dengan ( $n$ ) neuron input ( $x_1, x_2, \dots, x_n$ ), sebuah lapisan tersembunyi yang terdiri dari ( $p$ ) neuron ( $\varphi_1, \varphi_2, \dots, \varphi_p$ ) dan ( $m$ ) neuron output ( $y_1, y_2, \dots, y_m$ ). Jaringan ini dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan lapisan tunggal, meskipun kadangkala proses pelatihan lebih kompleks dan lama.



**Gambar 2.7** Jaringan Lapisan Jamak

## 2. Prosedur Pemodelan ANN

Berikut ini adalah prosedur pemodelan ANN sebagai generalisasi model matematika dari jaringan saraf biologi (Siang, 2009: 3):

- a. Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron),
- b. Sinyal dikirimkan diantara neuron-neuron melalui penghubung-penghubung,
- c. Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal,
- d. Untuk menentukan output, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan input yang diterima. Besarnya output ini selanjutnya dibandingkan dengan suatu batas ambang.

## 3. Fungsi Aktivasi

Pada *artificial neural network*, fungsi aktivasi digunakan untuk menentukan keluaran suatu neuron. Jika *net* merupakan kombinasi linear input dan bobot ( $net = \sum_{k=1}^p \varphi_{uk}x_k$ ), maka fungsi aktivasinya adalah  $f(net) = f(\sum_{k=1}^p \varphi_{uk}x_k)$  (Siang, 2009: 26). Pada beberapa kasus, fungsi aktivasi nonlinear dapat digunakan. Berikut ini adalah fungsi aktivasi non linear yang umum digunakan pada *artificial neural network* (Kusumadewi, 2004: 49-61):

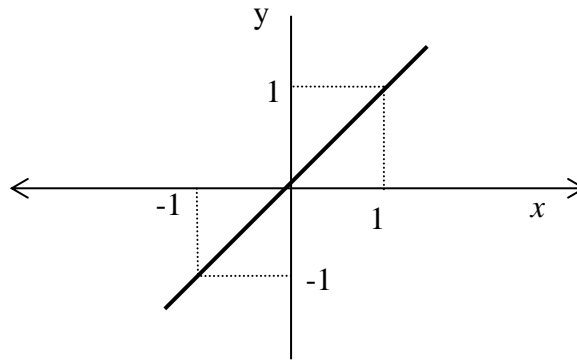
### a. Fungsi Linear

Fungsi linear sering dipakai apabila menginginkan output jaringan berupa sembarang bilangan riil. Pada fungsi identitas, nilai output yang dihasilkan sama dengan nilai input. Fungsi linear dirumuskan sebagai berikut:

$$y = f(x) = ax + b, x \in R \quad (2.17)$$

Fungsi identitas merupakan fungsi linear dengan nilai  $a = 1$  dan  $b = 0$ . Fungsi identitas dirumuskan sebagai berikut:

$$y = f(x) = x, x \in R \quad (2.18)$$

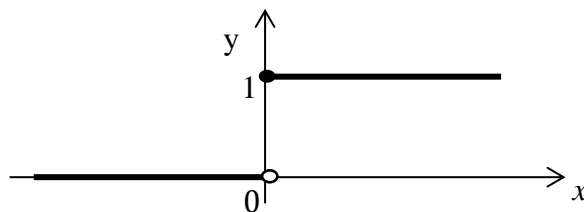


**Gambar 2.8** Fungsi Aktivasi Identitas

b. Fungsi Undak Biner (*Hard Limit*)

Jaringan dengan lapisan tunggal sering menggunakan fungsi undak (*step function*) untuk mengkonversikan input dari suatu variabel yang bernilai kontinu ke suatu output biner (0 atau 1). Fungsi ini sering digunakan pada jaringan dengan lapisan tunggal. Pada Matlab R2013a, perintah untuk menggunakan fungsi undak biner adalah *hardlim*. Gambar 2.9 adalah fungsi undak biner (*hard limit*) dengan rumus sebagai berikut:

$$y = f(x) = \begin{cases} 0, & \text{jika } x < 0 \\ 1, & \text{jika } x \geq 0 \end{cases} \quad (2.19)$$

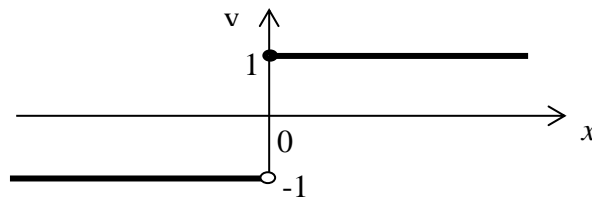


**Gambar 2.9** Fungsi Aktivasi Undak Biner (Hard Limit)

c. Fungsi Bipolar (*Symetric Hard Limit*)

Fungsi bipolar mirip dengan fungsi undak biner, perbedaannya terletak pada nilai output yang dihasilkan. Nilai output bipolar berupa 1 atau -1 (gambar 2.10). Pada Matlab R2013a, perintah untuk menggunakan fungsi bipolar adalah *hardlims*. Fungsi bipolar dirumuskan sebagai berikut:

$$y = f(x) = \begin{cases} -1, & \text{jika } x < 0 \\ 1, & \text{jika } x \geq 0 \end{cases} \quad (2.20)$$



**Gambar 2.10** Fungsi Aktivasi Bipolar (*Symetric Hard Limit*)

d. Fungsi Sigmoid Biner

Fungsi sigmoid biner sering digunakan karena nilai fungsinya terletak antara 0 dan 1 dan dapat diturunkan dengan mudah. Pada Matlab R2013a, perintah untuk menggunakan fungsi sigmoid biner adalah *logsig*. Fungsi sigmoid biner dirumuskan sebagai berikut:

$$y = f(x) = \frac{1}{1+e^{-x}}, \quad x \in R \quad (2.21)$$

e. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, perbedaannya terletak pada rentang nilai outputnya. Rentang nilai output fungsi sigmoid bipolar adalah -1 sampai 1. Pada Matlab R2013a, perintah untuk menggunakan fungsi sigmoid bipolar adalah *tagsig*. Fungsi sigmoid bipolar dirumuskan sebagai sebagai:

$$y = f(x) = \frac{1-e^{-x}}{1+e^{-x}}, \quad x \in R \quad (2.22)$$

#### **4. Algoritma Pembelajaran (*Learning Algorithm*)**

Algoritma pembelajaran adalah prosedur untuk menentukan bobot pada lapisan yang berhubungan dalam *artificial neural network* (Fausett, 1994: 429). Selama proses pembelajaran akan terjadi perbaikan bobot-bobot berdasarkan algoritma tertentu. Nilai bobot akan bertambah, jika informasi yang diberikan oleh neuron yang bersangkutan tersampaikan, sebaliknya jika informasi tidak tersampaikan oleh suatu neuron ke neuron lain, maka nilai bobot yang menghubungkan keduanya akan dikurangi. Pada saat pembelajaran dilakukan pada input yang berbeda maka nilai bobot akan diubah secara dinamis hingga mencapai suatu nilai yang cukup seimbang. Apabila nilai ini telah seimbang, maka mengindikasikan bahwa tiap-tiap input telah berhubungan dengan output yang diharapkan (Kusumadewi, 2004: 116-117).

Ada 2 metode pembelajaran NN, yaitu pembelajaran terawasi (*supervised learning*) dan pembelajaran tak terawasi (*unsupervised learning*) (Siang, 2009: 28-29):

##### **a. Pembelajaran Terawasi (*Supervised Learning*)**

Metode pembelajaran pada NN disebut terawasi jika, output yang diharapkan telah diketahui sebelumnya. Satu pola input akan diberikan ke suatu neuron pada lapisan input. Selanjutnya pola akan dirambatkan sepanjang NN hingga sampai ke neuron pada lapisan output. Lapisan output akan membangkitkan pola output yang akan dicocokkan dengan pola output targetnya. Error muncul apabila terdapat perbedaan antara pola output hasil pembelajaran dengan pola target. Diperlukan pembelajaran lagi apabila nilai error masih cukup besar.

b. Pembelajaran Tak Terawasi (*Unsupervised Learning*)

Pembelajaran tak terawasi tidak memerlukan target output dan tidak dapat ditentukan hasil yang diharapkan selama proses pembelajaran. Pada metode ini, tidak dapat ditentukan hasil outputnya. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu sesuai dengan nilai input yang diberikan. Tujuan pembelajaran ini adalah untuk mengelompokkan unit-unit yang hampir sama ke dalam suatu area tertentu.

**E. SOM Kohonen Clustering**

SOM Kohonen diperkenalkan oleh Teuvo Kohonen seorang ilmuwan Finlandia pada tahun 1982, sehingga SOM Kohonen dikenal juga dengan jaringan Kohonen. Jaringan SOM Kohonen merupakan salah satu jaringan saraf tiruan yang dikonfigurasi untuk pengelompokan data. Jaringan SOM Kohonen menggunakan metode pembelajaran *unsupervised* yang proses pelatihannya tidak memerlukan pengawasan (target output). Jaringan SOM Kohonen terdiri dari 2 lapisan (lapisan), yaitu lapisan input dan lapisan output. Setiap neuron dalam lapisan input terhubung dengan setiap neuron pada lapisan output. Setiap neuron dalam lapisan output merepresentasikan kelompok dari input yang diberikan (Siang, 2009: 141).

Penulisan istilah yang ada pada struktur jaringan SOM Kohonen adalah sebagai berikut :

$\alpha$  : laju pemahaman (*learning rate*),

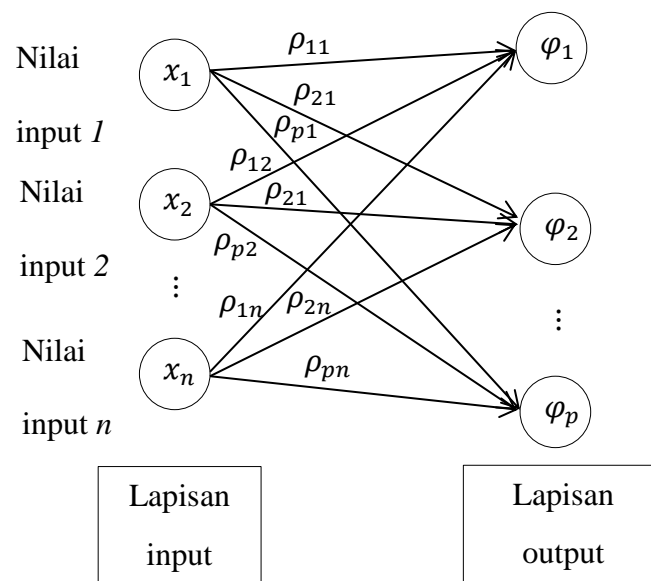
$R$  : parameter tetangga (*radius neighbourhood*),

$\rho$  : bobot lapisan kohonen,

$Cons$  : konstanta.

## 1. Arsitektur Jaringan SOM Kohonen

Arsitektur Jaringan SOM Kohonen tampak dalam gambar 2.11. Arsitektur ini mirip dengan model lain. Hanya saja Jaringan SOM Kohonen tidak menggunakan perhitungan net (hasil kali vektor masukan dengan bobot) maupun fungsi aktivasi (Siang, 2009: 142).



**Gambar 2.11** Arsitektur Jaringan SOM Kohonen

## 2. Algoritma Jaringan SOM Kohonen

Algoritma pengelompokan data jaringan SOM Kohonen adalah sebagai berikut (Siang, 2009: 143):

Langkah 0 Inisialisasi bobot  $\rho_{ki}$  dengan nilai random. Mengatur parameter laju pemahaman dan parameter tetangga,

Langkah 1 Jika kondisi penghentian bernilai salah atau kondisi selesai belum terpenuhi maka dilakukan langkah 2 sampai 7,

Langkah 2 Untuk setiap vektor masukan  $x$ , dilakukan langkah 3 sampai 5,

Langkah 3 Untuk setiap  $k$ , dihitung nilai,



$$D(k) = \sum_{i=1}^n (\rho_{ki} - x_i)^2 \quad (2.23)$$

Langkah 4 Mencari unit pemenang  $k$ , yaitu unit yang memiliki  $D(k)$  minimum,

Langkah 5 Melakukan perbaikan nilai  $\rho_{ki}$  dengan nilai tertentu yaitu :

$$\rho_{ki}(\text{baru}) = \rho_{ki}(\text{lama}) + \alpha[x_i - \rho_{ki}(\text{lama})] \quad (2.24)$$

Langkah 6 Memperbarui nilai laju pemahaman,

Langkah 7 Menentukan kondisi stop atau kondisi kapan iterasi diberhentikan.

Kondisi penghentian iterasi adalah selisih antara  $\rho_{ki}$  saat itu dengan  $\rho_{ki}$  iterasi sebelumnya. Apabila semua  $\rho_{ki}$  hanya berubah sedikit saja, berarti iterasi sudah mencapai konvergensi sehingga dapat dihentikan.

Menghitung pusat dan jarak dengan menggunakan metode SOM Kohonen *clustering* dalam penelitian ini dilakukan dengan menggunakan bantuan program Matlab R2013a, *script* secara lengkap terlampir pada lampiran (9) halaman (122). Berikut sebagian *script* terpenting dalam SOM Kohonen *clustering* (Siang, 2009: 283-291):

```
net=newc(PR,S,KLR,CLR);
net.IW{1,1};
net.b{1};
net.trainParam.epochs=500;
net= train(net,p);
b=sim(net,p);
ab=vec2ind(b);
```

dengan

net = Jaringan yang didefinisikan dalam newc  
[net=newc(PR,S,KLR,CLR),

|                                    |  |
|------------------------------------|--|
| <code>newc</code>                  | = Membentuk sebuah jaringan SOM Kohonen dengan spesifikasi tertentu $[PR, S, KLR, CLR]$ ,  |
| <code>PR</code>                    | = Matriks ukuran $R \times 2$ yang berisi nilai minimum dan maksimum masing-masing nilai vektor masukan,   |
| <code>S</code>                     | = Jumlah neuron target,  |
| <code>KLR</code>                   | = Laju Pemahaman SOM Kohonen ( <i>default</i> = 0,01),   |
| <code>CLR</code>                   | = Laju Pemahaman <i>Conscience</i> ( <i>default</i> = 0,001),  |
| <code>net.IW{1,1}</code>           | = Menampilkan bobot lapisan,   |
| <code>net.b{1}</code>              | = Menampilkan bobot bias lapisan,  |
| <code>net.trainParam.epochs</code> | = Menentukan epoch atau banyaknya iterasi ( <i>default</i> =500),  |
| <code>p</code>                     | = Input data,  |
| <code>train</code>                 | = Perintah <code>train</code> akan menghasilkan jaringan ( <code>net,p</code> ) baru [ <code>train(net,p)</code> ],  |
| <code>sim</code>                   | = menghitung keluaran jaringan yang berupa matriks $B$ berordo $p \times n$ ( $p$ = jumlah neuron target dan $n$ = jumlah vektor masukan). $B(k, i) = 1$ menyatakan bahwa vektor masukan ke- $i$ masuk ke dalam kelompok ke-1. |
| <code>vec2ind</code>               | = Mengetahui secara langsung pengelompokan vektor dari perintah <code>sim</code> ,   |

Bobot awal yang dibentuk adalah titik tengah `PR`.

### 3. Contoh Kasus *Cluster* Menggunakan Metode SOM Kohonen

Berikut merupakan contoh penerapan SOM Kohonen pada suatu kasus (Siang, 2009: 144-147):

Diketahui 4 buah vektor  $x_1 = (1,1,0,0)$ ,  $x_2 = (0,0,0,1)$ ,  $x_3 = (1,0,0,0)$  dan  $x_4 = (0,0,1,1)$ . Menggunakan jaringan SOM Kohonen untuk mengelompokkan 4 buah vektor tersebut ke dalam maksimum 2 kelompok. Diberikan laju pemahaman awal  $\alpha(0) = 0,6$  dan  $\alpha(t + 1) = 0,5\alpha(t)$ . Jari-jari vektor sekitar yang dimodifikasi = 0 (berarti hanya vektor pemenang yang dimodifikasi bobotnya pada setiap langkah).

Penyelesaian:

Inisialisasi bobot : Kolom matriks bobot menyatakan jumlah komponen dalam sebuah vektor (dalam kasus ini = 4) dan baris menyatakan jumlah maksimum kelompok yang akan dibentuk (dalam kasus = 2). Misalkan secara acak didapat bobot awal :

$$\rho = \begin{pmatrix} 0,2 & 0,6 & 0,5 & 0,9 \\ 0,8 & 0,4 & 0,7 & 0,3 \end{pmatrix}$$

Jari-jari sekitar yang dipakai = 0 dan laju pemahaman awal = 0,6.

Berdasarkan persamaan 2.23 pelatihan dilakukan untuk tiap vektor :

Untuk vektor  $x_1 = (1,1,0,0)$  :

$$D_1 = (0,2 - 1)^2 + (0,6 - 1)^2 + (0,5 - 0)^2 + (0,9 - 0)^2 = 1,86$$

$$D_2 = (0,8 - 1)^2 + (0,4 - 1)^2 + (0,7 - 0)^2 + (0,3 - 0)^2 = 0,98$$

$D_k$  minimum untuk  $k = 2$ , sehingga vektor bobot di baris 2 dimodifikasi menurut aturan persamaan 2.24 yaitu  $\rho_{2i}^{baru} = \rho_{2i}^{lama} + 0,6(x_i - \rho_{2i}^{lama})$ ,

$$\rho_{21} = 0,8 + 0,6(1 - 0,8) = 0,92; \rho_{22} = 0,4 + 0,6(1 - 0,4) = 0,76$$

$$\rho_{23} = 0,7 + 0,6(0 - 0,7) = 0,28; \rho_{24} = 0,3 + 0,6(0 - 0,3) = 0,12$$

Diperoleh vektor bobot baru :  $\rho = \begin{pmatrix} 0,2 & 0,6 & 0,5 & 0,9 \\ 0,92 & 0,76 & 0,28 & 0,12 \end{pmatrix}$

Untuk vektor  $x_2 = (0,0,0,1)$  :

$$D_1 = (0,2 - 0)^2 + (0,6 - 0)^2 + (0,5 - 0)^2 + (0,9 - 1)^2 = 0,66$$

$$D_2 = (0,92 - 0)^2 + (0,76 - 0)^2 + (0,28 - 0)^2 + (0,12 - 1)^2 = 2,28$$

$D_k$  minimum untuk  $k = 1$ , sehingga vektor bobot di baris 1 dimodifikasi.

Diperoleh:

$$\rho_{11} = 0,2 + 0,6(0 - 0,2) = 0,08; \rho_{12} = 0,6 + 0,6(0 - 0,6) = 0,24$$

$$\rho_{13} = 0,5 + 0,6(0 - 0,5) = 0,2; \rho_{14} = 0,9 + 0,6(1 - 0,9) = 0,96$$

Vektor bobot baru :  $\rho = \begin{pmatrix} 0,08 & 0,24 & 0,2 & 0,96 \\ 0,92 & 0,76 & 0,28 & 0,12 \end{pmatrix}$

Untuk vektor  $x_3 = (1,0,0,0)$  :

$$D_1 = (0,08 - 1)^2 + (0,24 - 0)^2 + (0,2 - 0)^2 + (0,96 - 0)^2 = 1,87$$

$$D_2 = (0,92 - 1)^2 + (0,76 - 0)^2 + (0,28 - 0)^2 + (0,12 - 0)^2 = 0,68$$

$D_k$  minimum untuk  $k = 2$ , sehingga vektor bobot di baris 2 dimodifikasi.

Diperoleh:

$$\rho_{21} = 0,92 + 0,6(1 - 0,92) = 0,968; \rho_{22} = 0,76 + 0,6(0 - 0,76) = 0,304$$

$$\rho_{23} = 0,28 + 0,6(0 - 0,28) = 0,112; \rho_{24} = 0,12 + 0,6(0 - 0,12) = 0,048$$

Vektor bobot baru :  $\rho = \begin{pmatrix} 0,08 & 0,24 & 0,2 & 0,96 \\ 0,968 & 0,304 & 0,112 & 0,048 \end{pmatrix}$

Untuk vektor  $x_4 = (0,0,1,1)$  :

$$D_1 = (0,08 - 0)^2 + (0,24 - 0)^2 + (0,2 - 1)^2 + (0,96 - 1)^2 = 0,7056$$

$$D_2 = (0,968 - 0)^2 + (0,304 - 0)^2 + (0,112 - 1)^2 + (0,048 - 1)^2 = 2,724$$

$D_k$  minimum untuk  $k = 1$ , sehingga bobot di baris 1 dimodifikasi. Diperoleh:

$$\rho_{11} = 0,08 + 0,6(0 - 0,08) = 0,032; \rho_{12} = 0,24 + 0,6(0 - 0,24) = 0,096$$

$$\rho_{13} = 0,2 + 0,6(1 - 0,2) = 0,68; \rho_{14} = 0,96 + 0,6(1 - 0,96) = 0,984$$

$$\text{Vektor bobot baru : } \rho = \begin{pmatrix} 0,032 & 0,096 & 0,68 & 0,984 \\ 0,968 & 0,304 & 0,112 & 0,048 \end{pmatrix}$$

Sebelum melakukan iterasi kedua dalam mengubah bobot, terlebih dahulu dilakukan modifikasi laju pemahaman,

$$\alpha^{\text{baru}} = 0,5(0,6) = 0,3$$

Iterasi kedua dilakukan secara analog dengan iterasi pertama. Baris vektor bobot diubah berdasarkan vektor yang saat itu digunakan.

Diperoleh hasil iterasi :

$$\text{Bobot awal : } \rho = \begin{pmatrix} 0,2 & 0,6 & 0,5 & 0,9 \\ 0,8 & 0,4 & 0,7 & 0,3 \end{pmatrix}$$

$$\text{Iterasi-1 : } \rho = \begin{pmatrix} 0,032 & 0,096 & 0,68 & 0,984 \\ 0,968 & 0,304 & 0,112 & 0,048 \end{pmatrix}$$

$$\text{Iterasi-2 : } \rho = \begin{pmatrix} 0,0053 & -0,17 & 0,7 & 1 \\ 0,99 & 0,3 & 0,02 & 0,086 \end{pmatrix}$$

dan seterusnya,

$$\text{Iterasi-100 : } \rho = \begin{pmatrix} 6,7e^{-17} & 2e^{-16} & 0,51 & 1 \\ 1 & 0,49 & 2,3e^{-16} & 1e^{-16} \end{pmatrix}$$

Tampak bahwa iterasi tersebut akan konvergen ke vektor bobot

$$\rho = \begin{pmatrix} 0 & 0 & 0,5 & 1 \\ 1 & 0,5 & 0 & 0 \end{pmatrix}$$

Pengelompokan vektor dilakukan dengan menghitung jarak vektor dengan bobot optimal.

Vektor  $x_1 = (1,1,0,0)$  memiliki

$$D_1 = (0 - 1)^2 + (0 - 1)^2 + (0,5 - 0)^2 + (1 - 0)^2 = 3,25$$

$$D_2 = (1 - 1)^2 + (0,5 - 1)^2 + (0 - 0)^2 + (0 - 0)^2 = 0,25$$

Karena  $D_k$  minimum untuk  $k = 2$ , sehingga  $x_1$  masuk dalam kelompok ke-2.

Secara analog, untuk  $x_2 = (0,0,0,1)$ ,  $D_1 = 0,25$  dan  $D_2 = 2,25$ , sehingga  $x_2$  masuk dalam kelompok ke-1.

Untuk  $x_3 = (1,0,0,0)$ ,  $D_1 = 2,25$  dan  $D_2 = 0,25$ , sehingga  $x_3$  masuk dalam kelompok ke-2.

Untuk  $x_4 = (0,0,1,1)$ ,  $D_1 = 0,25$  dan  $D_2 = 3,25$ , sehingga  $x_4$  masuk dalam kelompok ke-1.

#### F. Ketepatan Hasil Klasifikasi

Pengujian ketepatan hasil klasifikasi dilakukan untuk menguji sejauh mana model memberikan ketepatan hasil klasifikasi. Untuk menghitung ketepatan hasil klasifikasi digunakan tiga ukuran statistik. Tiga ukuran statistik tersebut adalah sensitivitas, spesifisitas, dan akurasi. Hasil kemungkinan yang dapat terjadi dalam hasil uji klasifikasi tersaji pada tabel 2.1 di bawah ini (Wong & Lim, 2011: 317):

**Tabel 2.1 Hasil Uji Klasifikasi**

| Hasil Uji Klasifikasi | Kondisi Penyakit sebagaimana yang ditetapkan oleh Standar Kebenaran |                |                              |
|-----------------------|---|----------------|------------------------------|
|                       | Positif   | Negatif        | Jumlah Baris                 |
| Positif               | <i>TP</i>   | <i>FP</i>      | <i>TP + FP</i>               |
| Negatif               | <i>FN</i>   | <i>TN</i>      | <i>FN + TN</i>               |
| Jumlah Kolom          | <i>TP + FN</i>  | <i>FP + TN</i> | <i>N = TP + FN + FP + TN</i> |

Berdasarkan tabel 2.1, terdapat empat kemungkinan hasil uji klasifikasi yaitu:

*True Positive* (TP) : klasifikasi asli citra menyatakan kanker dan hasil pembelajaran menyatakan kanker.

*True Negative* (TN) : klasifikasi asli citra menyatakan normal dan hasil pembelajaran menyatakan normal.

*False Positive* (FP) : klasifikasi asli citra menyatakan normal dan hasil pembelajaran menyatakan kanker.

*False Negative* (FN) : klasifikasi asli citra menyatakan kanker dan hasil pembelajaran menyatakan normal.

Sensitivitas adalah kemampuan uji untuk mengidentifikasi hasil positif yang tepat. Sensitivitas digunakan untuk mengidentifikasi pasien yang memiliki penyakit kanker dari keseluruhan pasien penyakit kanker. Rumus dari sensitivitas adalah sebagai berikut (Wong & Lim, 2011: 316):

$$\text{Sensitivitas} = \frac{TP}{TP+FN} \times 100\% \quad (2.25)$$

Spesifisitas adalah kemampuan uji untuk mengidentifikasi hasil negatif yang tepat. Spesifisitas digunakan untuk mengidentifikasi pasien yang normal dari keseluruhan pasien normal. Berikut adalah rumus dari spesifisitas (Wong & Lim, 2011: 316):

$$\text{Spesifisitas} = \frac{TN}{TN+FP} \times 100\% \quad (2.26)$$

Akurasi adalah kemampuan uji untuk mengidentifikasi hasil yang tepat, baik tepat positif atau tepat negatif dalam sebuah populasi. Berikut adalah rumus dari akurasi:

$$\text{Akurasi} = \frac{TN+TP}{N} \times 100\% \quad (2.27)$$