BAB II

LANDASAN TEORI

2.1 Flowering Plants (Tumbuhan yang Berbunga)

Tumbuhan yang berbunga adalah bagian dari Angiospermae atau tumbuhan berbiji tertutup (Yusa, 2011). Angiospermae memiliki kemampuan beradaptasi di berbagai lingkungan. Bunga dan buah merupakan ciri dari Angiospermae karena organ tersebut tidak terdapat pada kelompok tumbuhan yang lain (Yusa, 2011). Bunga merupakan alat reproduksi utama pada Angiospermae dan merupakan tempat terjadinya reproduksi seksual (Yusa, 2011). Bunga tersusun atas kelopak, mahkota, alat kelamin jantan, alat kelamin betina, bakal buah, dan bakal biji (Yusa, 2011).

2.2 Pengolahan Citra

Pengolahan citra secara umum adalah pemrosesan gambar berdimensi-dua melalui komputer (Jain, 1989). Menurut Efford (2000), pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara.

Pengolahan citra merupakan bagian penting yang mendasari beberapa aplikasi nyata, seperti pengenalan pola, penginderaan jarak jauh melalui satelit atau pesawat udara, dan *machine vision* (Kadir, 2013). Berikut merupakan beberapa contoh prinsip dasar dalam pengolahan citra yaitu:

- Peningkatan kecerahan dan kontras,
- Penghilangan derau,
- Pencarian bentuk objek.

Citra digital dibentuk oleh kumpulan titik yang dinamakan piksel (*pixel* atau *picture element*) (Kadir, 2013).Setiap piksel digambarkan sebagai satu kotak kecil. Setiap piksel mempunyai koordinat posisi. Sebuah piksel mempunyai koordinat berupa (x,y). Dalam hal ini x menyatakan posisi kolom, y menyatakan posisi baris.

Ada tiga jenis citra yang umum digunakan dalam pemrosesan citra. Ketiga jenis tersebut yaitu citra berwarna, citra berskala keabuan, dan citra biner (Kadir, 2013). Citra berwarna, atau biasa dinamakan citra RGB, merupakan jenis citra yang menyajikan warna dalam bentuk komponen R (merah), G (hijau), dan B (biru). Setiap komponen warna menggunakan delapan *bit* yang nilainya berkisar antara 0 sampai dengan 255. Dengan demikian, kemungkinan warna yang dapat disajikan mencapai 255 x 255 x 255 atau 16.581.375 warna (Kadir, 2013).

Citra berskala keabuan adalah citra yang menangani gradasi warna hitam dan putih yang menghasilkan efek warna abu-abu (Kadir, 2013). Pada jenis gambar ini, warna dinyatakan dengan intensitas. Intensitas berkisar antara 0 sampai dengan 255. Nilai 0 menyatakan hitam dan 255 menyatakan putih.

Citra biner adalah citra dengan setiap piksel hanya dinyatakan dengan sebuah nilai dari dua kemungkinan yaitu nilai 0 dan 1 (Kadir, 2013). Nilai 0 menyatakan warna hitam dan nilai 1 menyatakan warna putih. Citra jenis ini banyak dipakai dalam pemrosesan citra, misalnya untuk kepentingan memperoleh tepi bentuk suatu objek.

Pada pengolahan citra, operasi dapat dikategorikan menjadi empat macam, yaitu:

1. Operasi Piksel

   Keluaran pada piksel hanya bergantung pada *input* pada piksel tersebut, yang independen terhadap semua piksel pada citra (Shih, 2009). Sebagai contoh, *thresholding*, sebuah proses yang membuat piksel masukan di atas suatu tingkat *thresholding* tertentu menjadi berwarna putih, dan yang lainnya menjadi berwarna hitam, merupakan contoh sederhana dari sebuah operasi piksel (Shih, 2009).

2. Operasi Lokal

   Keluaran pada piksel tidak hanya bergantung pada piksel di lokasi yang sama pada citra masukan, tetapi juga pada piksel masukan yang berada di sekitarnya (Shih, 2009). Pada operasi ini, citra keluaran harus dibuat terpisah dari citra masukannya, sehingga piksel sekitarnya pada perhitungan semua berasal dari citra masukan (Shih, 2009). Contohnya adalah penyaringan morfologi, pendeteksian sisi atau garis, penyaringan halus, dsb (Shih, 2009).

3. Operasi Geometri

   Keluaran pada piksel hanya bergantung pada tingkatan masukan pada piksel lainnya yang didefinisikan oleh transformasi geometri (Shih, 2009). Operasi geometri berbeda dari operasi global dimana masukan hanya berasal dari beberapa piksel yang dikalkulasi menggunakan transformasi geometri (Shih, 2009).

4. Operasi Global

   Keluaran pada piksel bergantung pada semua piksel dalam citra tersebut (Shih, 2009). Sebuah operasi global dirancang untuk merefleksikan informasi statistik yang dikalkulasikan dari semua piksel pada citra, namun bukan dari sebuah *subset* piksel lokal (Shih, 2009). Sebagai contoh, sebuah transformasi jarak pada citra yang diberikan untuk setiap piksel jarak minimal ke piksel latar belakang, merupakan operasi global (Shih, 2009).

2.3 Edge Detection

Edge Detection berfungsi untuk memperoleh tepi objek (Kadir, 2013). Deteksi tepi memanfaatkan perubahan nilai intensitas yang drastis pada batas dua area. Definisi tepi di sini adalah "himpunan piksel yang terhubung yang terletak pada batas dua area" (Gonzalez & Woods, 2002). Perlu diketahui, tepi sesungguhnya mengandung informasi sangat penting. Informasi yang diperoleh dapat berupa bentuk maupun ukuran objek.

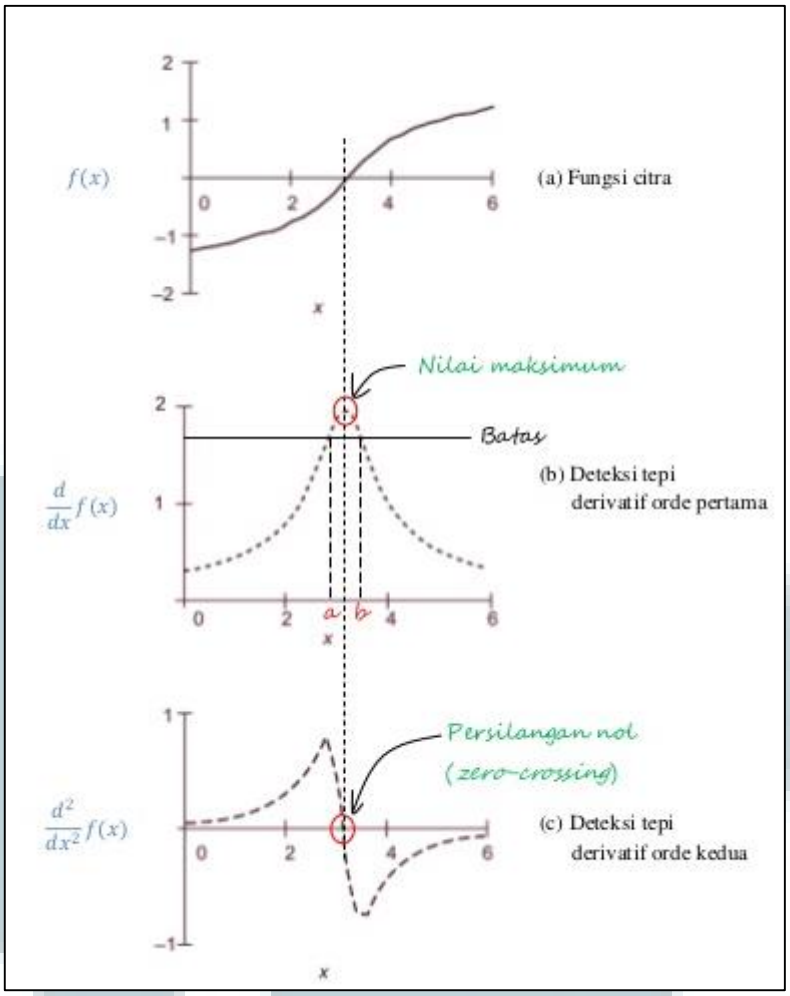
Umumnya, deteksi tepi menggunakan dua macam *detector*, yaitu *detector* baris (Hy) dan *detector* kolom (Hx). Beberapa contoh yang tergolong jenis ini adalah operator *Roberts*, *Prewitt*, *Sobel*, dan *Frei-Chen* (Kadir, 2013).

Deteksi tepi dapat dibagi menjadi dua golongan. Golongan pertama disebut deteksi tepi orde pertama, yang bekerja menggunakan turunan atau diferensial orde pertama. Termasuk dalam kelompok ini adalah operator *Roberts*, *Prewitt*, dan *Sobel*. Golongan kedua dinamakan deteksi tepi orde kedua, yang menggunakan turunan orde kedua. Contoh yang termasuk kelompok ini adalah *Laplacian of Gaussian*.

Tabel 2. 1 Turunan orde pertama dan kedua pada bentuk kontinu dan diskret (Kadir, 2013)

Turunan	Bentuk kontinu	Bentuk diskret
$\frac{df}{dx}$	$\lim_{\Delta x \to 0} \frac{f(y,x+\Delta x)f(y,x)}{\Delta x}$	f(y,x+1)f(y,x)
$\frac{df}{dy}$	$\lim_{\Delta x \to 0} \frac{f(y+\Delta y,x)f(y,x)}{\Delta y}$	f(y+1,x)f(y,x)
$\nabla f(y,x)$	$\left[\frac{df}{dy}, \frac{df}{dx}\right]$	[f(y, x+1)f(y,x), f(y+1,x)f(y,x)]
$\frac{d^2f}{dx^2}$	$\lim_{\Delta x \to 0} \frac{\left(\frac{df}{dx}\right)(y,x+\Delta x)\left(\frac{df}{dx}\right)f(y,x)}{\Delta x}$	f(y,x+1)- 2f(y,x)+f(y,x-1)
$\frac{d^2f}{dy^2}$	$\lim_{\Delta x \to 0} \frac{\left(\frac{df}{dy}\right)(y+\Delta y,x)\left(\frac{df}{dy}\right)f(y,x)}{\Delta y}$	f(y+1,x) -2f(y,x)+f(y-1,x)
$\nabla^2 f(y,x)$	$\frac{d^2f}{dx^2}+\frac{d^2f}{dy^2}$	f(y,x+1)+f(y,x-1) - 4f(y,x)+f(y+1,x)+f(y-1,x)

Tabel diatas memberikan definisi turunan orde pertama dan kedua baik pada bentuk kontinu maupun diskret. Bentuk diskret sangat berguna untuk deteksi tepi. Gambar 2.1 menunjukkan hubungan antara fungsi citra dan deteksi tepi orde pertama dan kedua. Perlu diketahui, terkait dengan turunan, tepi sesungguhnya terletak pada nilai absolut maksimum pada turunan pertama dan persilangan nol pada turunan kedua (Kadir, 2013).



Gambar 2.1 Deteksi tepi orde pertama dan orde kedua pada arah x (Kadir, 2013)

Contoh di Gambar 2.1(a) menunjukkan keadaan fungsi intensitas citra f(x, y) pada arah x dengan bentuk tepi tanjakan landai. Gambar 2.1(b) menunjukkan keadaan turunan pertama pada arah x. Puncak di Gambar 2.1(b) menyatakan letak tepi pada turunan pertama, sedangkan persilangan nol di Gambar 2.1(c) menyatakan letak tepi pada turunan kedua. Apabila nilai batas dikenakan pada turunan pertama, puncak tidak lagi menjadi tepi. Akibatnya, terdapat dua nilai yang memenuhi yaitu a dan b. Kedua nilai tersebut akan menjadi piksel-piksel tepi. Berbeda halnya pada turunan kedua, tepi akan selalu berupa satu piksel. Hal itu

terlihat pada perpotongan fungsi turunan kedua dengan sumbu x. Akibatnya ketebalan tepi akan selalu berupa satu piksel.

Deteksi tepi dengan turunan orde pertama dilakukan dengan menggunakan operator gradien (Kadir, 2013). Operator gradien didefinisikan sebagai vektor pada rumus dibawah ini.

$$v_f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{df}{dx} \\ \frac{df}{dy} \end{bmatrix}$$ .....Rumus 2.1

Besaran vektor dihitung menggunakan rumus seperti rumus dibawah ini.

$$\nabla f = mag(\nabla f) = \left[{G_x}^2 + {G_y}^2 \right]^{1/2} = \left[(\frac{df}{dx})^2 + (\frac{df}{dy})^2 \right]^{1/2}$$ ..Rumus 2.2

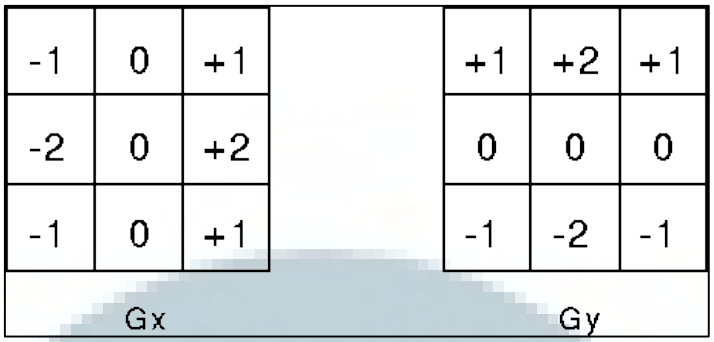
Namun, untuk alasan penyederhanaan komputasi, operasi akar ditiadakan sehingga besaran vektor tersebut dihampiri melalui rumus 2.3.

$$\nabla f \approx {G_x}^2 + {G_y}^2$$ .....Rumus 2.3

Perlu diketahui, besaran gradien sering disebut sebagai "gradien" saja. Adapun turunan orde kedua yang biasa digunakan dalam pengolahan citra dihitung dengan menggunakan *Laplacian* (Kadir, 2013). Perhitungannya seperti rumus pada Gambar 2.5.

$$\nabla^2 f(y,x) = \frac{d^2 f(y,x)}{dx^2} + \frac{d^2 f(y,x)}{dy^2}$$ .....Rumus 2.4

Operator Sobel lebih sensitif terhadap tepi diagonal daripada tepi vertikal dan *horizontal* (Crane, 1997). Operator Sobel dapat dilihat pada gambar berikut.



Gambar 2.2 Operator Sobel

Sumber : http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm

2.4 Neural Network (Jaringan Syaraf Tiruan)

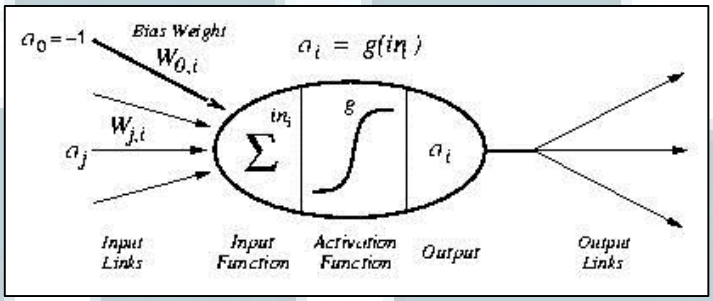
Jaringan syaraf tiruan merupakan sebuah model seperti jaringan syaraf pada manusia, yang dapat digunakan untuk pengenalan pola dan pembelajaran mesin (Lejap, 2008). Jaringan syaraf tiruan adalah sekumpulan sel neuron saling terhubung secara sederhana (Lagudu & Sarma, 2013). Sel-sel neuron yang ada diatur sehingga setiap sel dapat mengirimkan masukkan dari satu sel ke sel lainnya (Lagudu & Sarma, 2013). Keluaran yang dihasilkan dari jaringan ini didapatkan dari masukan yang disebarkan ke seluruh jaringan dari sel-sel yang aktif (Lagudu & Sarma, 2013). Jaringan syaraf tiruan ditentukan oleh tiga hal, yaitu (Siang, 2004):

1. Pola hubungan antar neuron (Arsitektur jaringan).
2. Metode untuk menentukan bobot penghubung (Metode pelatihan)
3. Fungsi aktivasi

2.4.1 Neuron

Neuron merupakan sel sama halnya seperti dengan neuron pada syaraf manusia. Neuron memproduksi sebuah keluaran (Lagudu & Sarma, 2013). Neuron

yang menghasilkan satu keluaran biasanya bersifat linier karena tidak ada fungsi aktivasi yang digunakan (Lagudu & Sarma, 2013). Pada jaringan syaraf tiruan, sebuah neuron terdiri dari tiga bagian yaitu fungsi penjumlahan atau fungsi masukan, fungsi aktivasi, dan keluaran (Siang, 2004).



Gambar 2.3 Neuron

Sumber: (Russell, 1995:568)

Pada gambar di atas dapat dilihat bahwa masukan akan dikirim ke neuron dengan bobot tertentu. Masukan ini akan diproses oleh suatu fungsi yang akan menjumlahkan perkalian nilai bobot dengan masukan. Hasil penjumlahan kemudian akan dihitung dalam fungsi aktivasi dan kemudian akan dibandingkan hasilnya dengan suatu ambang batas (*threshold*). Jika hasil keluaran melewati suatu batas ambang tertentu maka neuron tersebut akan diaktifkan, jika tidak neuron tersebut akan mengirimkan keluaran melalui bobot keluaran ke semua neuron yang berhubungan dengannya melalui penghubung (Agustin, 2012).

2.4.2 Layer

Pada jaringan syaraf tiruan, *layer* dapat berjumlah 2 yang berupa *input layer* dan *output layer* atau lebih. Setiap *layer* terdiri dari beberapa neuron, sesuai yang diinginkan. *Layer* penyusun jaringan syaraf tiruan dapat dibagi menjadi tiga, yaitu:

1. *Input Layer*

   Unit di dalam input layer disebut input unit. Unit ini menerima pola input data dari luar yang menggambarkan suatu permasalahan yang akan dipecahkan (Agustin, 2012).

2. *Hidden Layer*

   Unit di dalam *hidden layer* disebut *hidden unit* dimana keluarannya tidak dapat secara langsung diamati (Agustin, 2012). *Hidden layer* merupakan *layer* dari elemen pemrosesan yang tersembunyi dari koneksi langsung ke titik luar dari jaringan syaraf (Priddy & Keller, 2005). Semua koneksi ke dan dari *hidden layer* bersifat internal pada jaringan syaraf (Priddy & Keller, 2005).

3. *Output Layer*

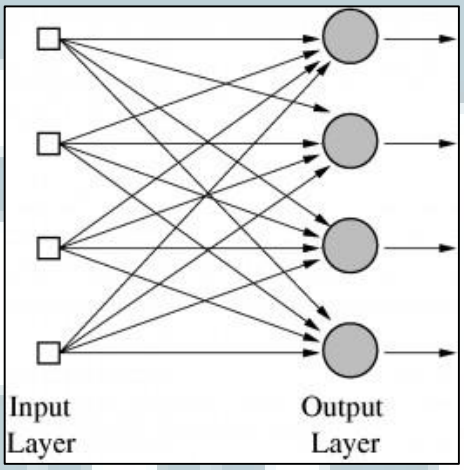
   Unit di dalam output layer disebut output unit. Keluaran dari lapisan ini merupakan solusi jaringan syaraf terhadap suatu permasalahan (Agustin, 2012).

2.4.3 Arsitektur Jaringan Syaraf Tiruan

Beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi antara lain (Siang, 2004).

1. Jaringan Layar Tunggal (*single layer network*)

   Dalam jaringan ini, sekumpulan neuron masukan dihubungkan langsung dengan sekumpulan neuron keluaran (Siang, 2004).



Gambar 2.4 *Single layer network*

Pada Gambar 2.8, terlihat bahwa satu neuron pada layar masukan akan dihubungkan langsung ke semua neuron pada lapisan keluaran tanpa dihubungkan dengan neuron lain.

2. Jaringan Layar Jamak (*multi layer network*)

Jaringan layar jamak merupakan perluasan dari layar tunggal (Siang, 2004). Dalam jaringan ini, selain unit masukan dan keluaran, ada unit lain yang sering disebut dengan layar tersembunyi (*hidden layer*) (Siang, 2004). Dimungkinkan pula ada beberapa layar tersembunyi (Siang, 2004). Sama seperti pada unit masukan dan keluaran, unit dalam satu layar tidak saling berhubungan (Siang, 2004).

3. Jaringan *Recurrent*

   Model jaringan *recurrent* mirip dengan jaringan layar tunggal ataupun ganda (Siang, 2004). Hanya saja, ada neuron keluaran yang memberikan sinyal kepada unit masukan (*feedback loop*) (Siang, 2004).

2.4.4 Fungsi Aktivasi

Dalam jaringan syaraf tiruan, fungsi aktivasi dipakai untuk menentukan keluaran suatu neuron (Siang, 2004). Beberapa fungsi aktivasi yang sering dipakai adalah sebagai berikut:

1. Fungsi *threshold* (batas ambang)

$$f(x) = \begin{cases} 1\ jika\ x\ \geq a \\ 0\ jika\ x\ < a \end{cases} \text{.....Rumus 2.5}$$

$$f(x) = \begin{cases} 1\ jika\ x\ \geq a \\ -1\ jika\ x\ < a \end{cases} \text{.....Rumus 2.6}$$

   Untuk beberapa kasus, fungsi *threshold* yang dibuat tidak berharga 0 atau 1, tapi berharga -1 atau 1 (sering disebut *threshold* bipolar) (Siang, 2004).

2. Fungsi sigmoid

$$f(x) = \frac{1}{1+ e^{-x}} \text{.....Rumus 2.7}$$

$$f'(x) = f(x)(1 - f(x)) \text{ .....Rumus 2.8}$$

Fungsi sigmoid sering dipakai karena nilai fungsinya yang terletak antara 0 dan 1 dan dapat diturunkan dengan mudah (Siang, 2004).

3. Fungsi identitas

$$f(x) = x$$ .....Rumus 2.9

Fungsi identitas sering dipakai apabila kita menginginkan keluaran jaringan berupa sembarang bilangan riil (Siang, 2004).

2.4.5 Pelatihan Standar Backpropagation

Pelatihan *Backpropagation* meliputi 3 fase (Siang, 2004). Fase pertama adalah fase maju. Pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase kedua adalah fase mundur. Selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, mulai dari haris yang berhubungan langsung dengan unit-unit di layar keluaran. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi (Siang, 2004). Berikut merupakan penjelasan dari fase-fase tersebut:

1. Propagasi maju

   Selama propagasi maju, sinyal masukan dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan (Siang, 2004). Keluaran dari setiap unit layar tersembunyi tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan (Siang, 2004). Demikian seterusnya hingga menghasilkan keluaran jaringan. Berikutnya, keluaran jaringan dibandingkan dengan target yang harus dicapai.

Selisih $t_k - y_k$ adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan, maka iterasi dihentikan. Akan tetapi apabila kesalahan masih lebih besar dari batas toleransinya, maka bobot setiap haris dalam jaringan akan dimodifikasi untuk mengurangi kesalahan yang terjadi (Siang, 2004).

2. Propagasi mundur

   Berdasarkan kesalahan $t_k - y_k$, dihitung faktor $\partial_k (k = 1,2, \dots, m)$ yang dipakai untuk mendistribusikan kesalahan di unit y_k ke semua unit tersembunyi yang terhubung langsung dengan y_k (Siang, 2004). ∂_k juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran.

   Dengan cara yang sama, dihitung ∂_j di setiap unit di layar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar di bawahnya. Demikian seterusnya hingga semua faktor ∂ di unit tersembunyi yang berhubungan langsung dengan unit masukan dihitung (Siang, 2004).

3. Perubahan bobot

   Setelah semua faktor ∂ dihitung, bobot semua garis dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor ∂ neuron di layar atasnya.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian terpenuhi (Siang, 2004). Untuk setiap pengulangan dalam pelatihan jaringan syaraf tiruan disebut *epoch*. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan, atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan (Siang, 2004).

Algoritma pelatihan untuk jaringan dengan satu layar tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut (Siang, 2004):

1. Initialisasi semua bobot dengan bilangan acak kecil.
2. Jika kondisi penghentian belum terpenuhi, lakukan langkah 3 – 10.
3. Untuk setiap pasang data pelatihan, lakukan langkah 4 – 9.
4. Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya.
5. Hitung semua keluaran di unit tersembunyi z_j $(j = 1,2, \dots, p)$.

$$z\_net_j = v_{j0} + \sum_{i=1}^{n} x_i v_{ji} \quad \text{.....Rumus 2.10}$$

$$z_j = f(z\_net_j) = \frac{1}{1+ e^{-z\_net_j}} \quad \text{.....Rumus 2.11}$$

6. Hitung semua keluaran jaringan di unit y_k $(k = 1,2, \dots, m)$.

$$y\_net_k = w_{k0} + \sum_{j=1}^{p} z_j w_{kj} \quad \text{.....Rumus 2.12}$$

$$y_k = f(y\_net_k) = \frac{1}{1+ e^{-y\_net_k}} \quad \text{.....Rumus 2.13}$$

7. Hitung faktor ∂ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k $(k = 1,2, \dots, m)$.

$$\partial_k = (t_k - y_k) f'(y\_net_k) = (t_k - y_k) y_k (1 - y_k) \quad \text{.....Rumus 2.14}$$

∂_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar di bawahanya. Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk merubah bobot w_{kj}) dengan laju percepatan α (Siang, 2004). α juga sering disebut *learning rate* dimana *learning rate* berfungsi untuk menentukan kecepatan jaringan syaraf tiruan untuk merubah bobotnya.

$$\Delta w_{kj} = \alpha\, \partial_k\, z_j \,; k = 1,2, \dots, m \,; j = 0,1, \dots, p \quad \text{.....Rumus 2.15}$$

8. Hitung faktor ∂ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j $(j = 1,2, \dots, p)$.

$$\partial\_net_j = \sum_{k=1}^{m} \partial_k w_{kj}$$ .....Rumus 2.16

Faktor ∂ unit tersembunyi :

$$\partial_j = \partial\_net_j \, f'(z\_net_j) = \partial\_net_j \, z_j \, (1 - z_j)$$ .....Rumus 2.17

Hitung suku perubahan bobot v_{ji}.

$$\Delta v_{ji} = \alpha \, \partial_j \, x_i \quad ; j = 1,2, \dots, p \; ; i = 0,1, \dots, n$$ .....Rumus 2.18

9. Hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran:

$$w_{kj}(baru) = w_{kj}(lama) + \Delta w_{kj} \; (k = 1,2, \dots, m \; ; j = 0,1, \dots, p)$$ Rumus 2.19

Perubahan bobot garis yang menuju ke unit tersembunyi:

$$v_{ji}(baru) = v_{ji}(lama) + \Delta v_{ji} \; (j = 1,2, \dots, m \; ; i = 0,1, \dots, p)$$ .....Rumus 2.20

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (langkah 5 dan 7) saja yang dipakai untuk menentukan keluaran jaringan.

2.4.6 Pemilihan Bobot dan Bias Awal

Bobot awal akan mempengaruhi apakah jaringan mencapai titik minimum lokal atau global, dan seberapa cepat konvergensinya (Siang, 2004).

Bobot yang menghasilkan nilai turunan aktivasi yang kecil sedapat mungkin dihindari karena akan menyebabkan perubahan bobotnya menjadi sangat kecil. Demikian pula nilai bobot awal tidak boleh terlalu besar karena nilai turunan fungsi

aktivasinya menjadi sangat kecil juga (Siang, 2004). Oleh karena itu dalam standar *Backpropagation*, bobot dan bias diisi dengan bilangan acak kecil.

Nguyen dan Widrow (1990) mengusulkan cara membuat inisialisasi bobot dan bias ke unit tersembunyi sehingga menghasilkan iterasi lebih cepat.

Misalnya, n adalah jumlah unit masukan , p adalah jumlah unit tersembunyi, dan β adalah faktor skala yang berupa $0.7\sqrt[n]{p}$. Algoritma inisialisasi Nguyen Widrow adalah sebagai berikut (Siang, 2004):

1. Inisialisasi semua bobot (v_{ji} (lama)) dengan bilangan acak dalam interval [-0.5,0.5].
2. Hitung $\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + \ldots + v_{jn}^2}$.....Rumus 2.21
3. Bobot yang dipakai sebagai inisialisasi = $v_{ji} = \frac{\beta\, v_{ji}(lama)}{\|v_j\|}$. .....Rumus 2.22
4. Bias yang dipakai sebagai inisialisasi = v_{j0} = bilangan acak antara $-\beta$ dan β.

2.4.7 Momentum

Pada standar *Backpropagation*, perubahan bobot didasarkan atas gradien yang terjadi untuk pola yang dimasukkan saat itu (Siang, 2004). Modifikasi yang dapat dilakukan adalah melakukan perubahan bobot yang didasarkan atas arah gradien pola terakhir dan pola sebelumnya (disebut momentum) yang dimasukkan. Jadi tidak hanya pola masukan terakhir saja yang diperhitungkan.

Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain. Apabila beberapa data terakhir yang diberikan ke jaringan memiliki pola serupa, maka perubahan bobot dilakukan secara cepat. Namun apabila data terakhir yang

dimasukkan memiliki pola yang berbeda dengan pola sebelumnya, maka perubahan dilakukan secara lambat.

Dengan penambahan momentum, bobot baru pada waktu ke (t+1) didasarkan atas bobot pada waktu t dan (t-1). Disini harus ditambahkan 2 variabel baru yang mencatat besarnya momentum untuk 2 iterasi terakhir. Jika μ adalah konstanta ($0 \leq \mu \leq 1$) yang menyatakan parameter momentum (*momentum rate)* maka bobot baru dihitung berdasarkan persamaan (Siang, 2004):

$$w_{kj}(t+1) = w_{kj}(t) + \alpha\, \partial_k\, z_j + \mu\, (w_{kj}(t) - w_{kj}(t-1)) \quad \text{..Rumus 2.23}$$

$$v_{ji}(t+1) = v_{ji}(t) + \alpha\, \partial_j\, x_i + \mu\, (v_{ji}(t) - v_{ji}(t-1)) \quad \text{.....Rumus 2.24}$$