

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Non Statistik

2.1.1 Beras

Berdasarkan Peraturan Menteri Perdagangan RI Nomor 19/M-DAG/PER/3/2014 menjelaskan definisi beras secara umum, bahwa beras adalah biji-bijian baik berkulit, tidak berkulit, diolah atau tidak diolah yang berasal dari *Oriza Sativa*. Pada definisi ini beras mencakup gabah, beras giling, dan beras pecah kulit. Sedangkan definisi umum, beras merupakan bagian butir padi (gabah) yang telah dipisah dari sekam dan dedak atau bekatul (Kementan, 2015). Sehingga dapat dikatakan bahwa beras merupakan produk akhir dari gabah. Gabah tidak didefinisikan sebagai beras. Tanaman padi dapat tumbuh hingga setinggi 1-1,8 m. Daunnya panjang dan ramping dengan panjang 50-100 cm dan lebar 2-2,5 cm. Beras yang dapat dimakan berukuran panjang 5-12 mm dan tebal 2-3 mm (Kuswardani, 2013).

Beras sebagai menu pokok harian yang selalu dikonsumsi oleh hampir seluruh masyarakat Indonesia ini memiliki kandungan pati yang cukup besar dibandingkan dengan sereal. Selain itu, dalam beras juga mengandung vitamin, protein, mineral, dan air. Beras yang mengandung karbohidrat sangat dibutuhkan untuk seseorang yang memiliki banyak aktivitas karena karbohidrat berguna sebagai pemasok energi untuk tubuh (Ramadhanny, 2015).

2.1.2 Harga Beras

Harga komoditas beras merupakan harga yang pergerakannya terus dipantau dan diintervensi oleh pemerintah. Hal ini dilakukan karena harga beras memberi kontribusi pada ketahanan pangan, kemiskinan, stabilitas makro ekonomi dan pertumbuhan ekonomi Negara. Hingga saat ini pergerakan harga beras itu tentu dipengaruhi oleh 3 faktor, yang pertama adalah faktor ketersediaan beras yang diketahui bahwa ketersediaan beras itu berasal dari hasil produksi panen petani padi di daerah sentra produksi. Faktor ini juga dipengaruhi oleh beberapa hal seperti luas lahan panen, perubahan iklim yang berpengaruh terhadap hasil produksi, produktivitas, pergeseran musim tanam, musim panen, serta adanya serangan hama penyakit terhadap proses budidaya padi yang berpengaruh terhadap produksi.

Faktor kedua yaitu faktor permintaan dari konsumen, diketahui bahwasanya dengan adanya peningkatan dan penurunan permintaan konsumen bisa mempengaruhi harga beras terutama dalam menghadapi Hari Besar Keagamaan Nasional, adanya kepanikan atau kekhawatiran konsumen akan kelangkaan beras di pasar serta adanya perubahan pola konsumsi, preferensi dan diversifikasi pangan kebutuhan pokok konsumen. Faktor berikutnya yaitu faktor distribusi, mampu mempengaruhi kenaikan maupun penurunan harga beras. Dalam proses distribusi beras dapat mengeluarkan beberapa biaya seperti biaya distribusi, jarak dari sentra produksi ke konsumen, kemudian adanya gangguan dalam proses distribusi juga bisa menjadi faktor yang berpengaruh. Selain faktor-faktor diatas, kebijakan pemerintah juga ikut andil dalam pergerakan harga beras ini, yaitu

kebijakan impor dan ekspor beras, kebijakan pembelian dan penjualan beras dengan harga tertentu yang dilaksanakan oleh Bulog (Kemendag,2018).

2.2 Tinjauan Statistik

2.2.1 Analisis Deskriptif

Menurut Sugiyono (2017:147), analisis deskriptif adalah statistik yang digunakan untuk menganalisis data dengan cara mendeskripsikan atau menggambarkan data yang telah terkumpul sebagaimana adanya tanpa bermaksud membuat kesimpulan yang berlaku untuk umum atau generalisasi. Penelitian diskriptif tujuannya untuk menjelaskan atau mendeskripsikan suatu peristiwa, keadaan, objek apakah orang, atau segala sesuatu yang terkait dengan variabel-variabel yang bisa dijelaskan baik menggunakan angka-angka maupun kat-kata (Punaji,2010).

2.2.2 Peramalan

Peramalan adalah suatu pengambilan keputusan yang didasari dengan peristiwa masa lalu untuk memprediksi hasil masa depan. Selain itu peramalan juga bisa diartikan sebagai penggunaan data lampau untuk menentukan tres masa depan (Pakaja, et al, 2012).

Jenis-jenis peramalan (forecasting) berdasarkan horizon waktu dibagi menjadi tiga jenis yaitu (Herjanto, 2008):

a. Peramalan jangka panjang (long-range forecast)

Peramalan ini mencakup waktu yang lebih panjang atau lama (1-2 tahun)

b. Peramalan jangka menengah (medium-range forecast)

Peramalan ini mencakup waktu yang lebih pendek dari peramalan jangka panjang yaitu sekitar 2 bulan sampai dengan 1 tahun.

c. Peramalan jangka pendek (short-range forecast)

Peramalan ini adalah peramalan yang memakan waktu dekat misalnya memperhatikan target produksi harian suatu perusahaan.

2.2.3 Analisis Time Series

Time series merupakan serangkaian observasi terhadap suatu variabel yang diambil secara berurutan berdasarkan interval waktu yang tetap (Wei, 2006). Data time series merupakan jenis data yang terdiri dari satu objek tetapi meliputi beberapa periode waktu misalnya harian, mingguan, bulanan, tahunan, dan lain-lain. Data time series dianggap sangat berguna untuk memprediksi kejadian di masa depan. Hal ini diyakini pola yang ada pada masa lalu akan terulang kembali di masa mendatang (F.Al Huda, Ridok, & Dewi, 2014).

2.2.4 Data Training dan Testing

Sebelum proses pengujian, data akan dibagi menjadi data training dan data testing yang diperlukan untuk membangun jaringan syaraf tiruan. Pengembangan model jaringan syaraf tiruan dapat dilakukan dengan menggunakan data Training. Sedangkan untuk mengevaluasi kemampuan model peramalan yang digunakan adalah data Testing. Pemilihan data Training dan data Testing dapat mempengaruhi kinerja Jaringan Syaraf Tiruan (G.Zhang, Eddy Patuwo, & Y.Hu, 1998).

2.2.5 Normalisasi Data

Sebelum dilakukan proses pelatihan, maka diperlukan penskalaan pada data *training* dan data *testing* sehingga data tersebut masuk pada range tertentu atau disebut normalisasi data. Pada penelitian ini data yang dinormalisasi harus sesuai dengan fungsi aktivasi yang digunakan. Oleh karena penelitian ini menggunakan fungsi aktivasi biner yaitu data yang dinormalisasikan akan berada pada range 0 hingga 1. Menurut Siang (2005) persamaan yang digunakan untuk normalisasi yaitu :

$$x = \frac{0,8 (x_p - \text{Min}(x_p))}{(\text{Max}(x_p) - \text{Min}(x_p))} + 0,1 \quad (22)$$

Dimana :

x = input setelah dinormalisasi

x_p = nilai data asli yang belum dinormalisasi

$\text{Min}(x_p)$ = Nilai Minimum pada data set

$\text{Max}(x_p)$ = Nilai Maksimum pada data set

2.2.6 Jaringan Syaraf Tiruan (*Neural Network*)

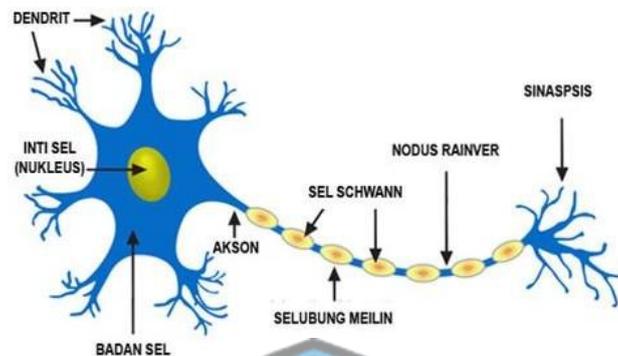
Jaringan syaraf tiruan (JST) atau sering dikenal dengan istilah neural network adalah sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf biologi manusia (Siang, 2005). Neural network telah diaplikasikan dalam berbagai bidang diantaranya pattern recognition, medical diagnostic, signal processing, dan peramalan. Pada dasarnya, neural network merupakan kumpulan dari elemen-elemen pemroses yang saling berhubungan, yang disebut dengan unit-unit atau syaraf-syaraf (Suhartanto,2007). Layaknya otak manusia dalam merespon kondisi lingkungan berbeda-beda, peranan JST dalam bidang penelitian dan pengembangan sangat penting di masa yang akan datang yang menuntut aspek optimasi dan aspek interaktif antara alat dan manusia (Muis, 2017).

Menurut Jong Jek Siang (2009), sistem jaringan syaraf tiruan ditentukan oleh 3 hal, yaitu:

1. Pola hubungan antar neuron atau bisa disebut arsitektur jaringan.
2. Metode untuk menentukan bobot penghubung (disebut metode training/learning/algorithm).
3. Fungsi aktivasi.

Dalam syaraf biologis, setiap sel syaraf (neuron) akan memiliki satu sel yang bertugas untuk melakukan pemrosesan informasinya yang akan diterima oleh dendrit. Selain itu menerima informasi, dendrit juga menyertai axon sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini menjadi

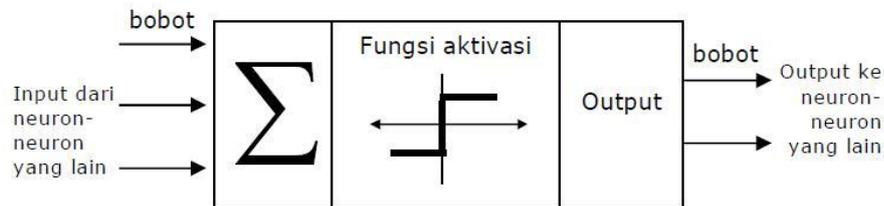
masukannya ke neuron lain (Hadjratie,2011). Berikut adalah bentuk sederhana dari sebuah neuron yang oleh para ahli dianggap sebagai satuan unit pemroses.



Gambar 2.1 Jaringan Syaraf Manusia

Jika dilihat, neuron-neuron buatan tersebut bekerja dengan cara yang sama dengan neuron-neuron biologis. Input yang datang akan diproses oleh suatu fungsi perambatan dengan menjumlahkan nilai-nilai dari semua bobot tersebut. Hasil dari penjumlahan tersebut kemudian akan dibandingkan dengan suatu nilai ambang (threshold) melalui suatu fungsi aktivasi pada setiap neuron. Apabila input tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan. Sebaliknya, jika input tidak melewati suatu nilai ambang tertentu maka neuron tidak akan diaktifkan. Apabila neuron tersebut tidak diaktifkan. Apabila neuron tersebut tidak diaktifkan maka neuron tersebut akan mengirimkan output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya, begitu seterusnya (Kusumadewi dan Hartati, 2010:70).

Cara kerja neuron tersebut dapat dilihat pada Gambar 2.2



Gambar 2.2 Struktur Neuron Jaringan

Pada jaringan syaraf, neuron –neuron berada dalam lapisan-lapisan (layer) yang disebut lapisan neuron. Menurut Puspitaningrum (2006:9), lapisan-lapisan penyusun jaringan syaraf tiruan dibagi menjadi tiga yaitu :

1. Lapisan input (Input layer)

Neuron-neuron yang berada di dalam lapisan input disebut neuron-neuron input. Neuron-neuron ini menerima input dari luar. Input yang dimasukkan merupakan penggambaran dari suatu masalah.

2. Lapisan tersembunyi (hidden layer)

Neuron-neuron di dalam lapisan tersembunyi disebut neuron-neuron tersembunyi. Output dari lapisan ini tidak secara langsung dapat diamati.

3. Lapisan Output (Output Layer)

Neuron-neuron pada lapisan output disebut neuron-neuron output. Keluaran atau output dari lapisan ini merupakan output jaringan syaraf tiruan terhadap suatu permasalahan.

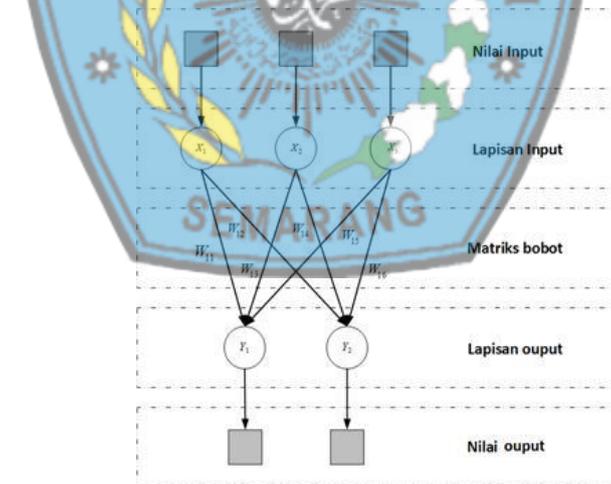
2.2.7 Arsitektur Jaringan Syaraf Tiruan

Jaringan syaraf tiruan memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur jaringan syaraf tiruan tersebut, antara lain (Agustin,2012) :

1. Jaringan Layer Tunggal (single layer network)

Jaringan dengan lapisan tunggal terdiri dari 1 layer input dan 1 layer output. Setiap neuron/unit yan terdapat di dalam lapsan/layer input selalu terhubung dengan setiap neural yang terdapat pada layer output. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalu lapisan tersembunyi.

Cara kerja jaringan saraf dengan lapisan tunggal digambarkan pada Gambar 2.3.



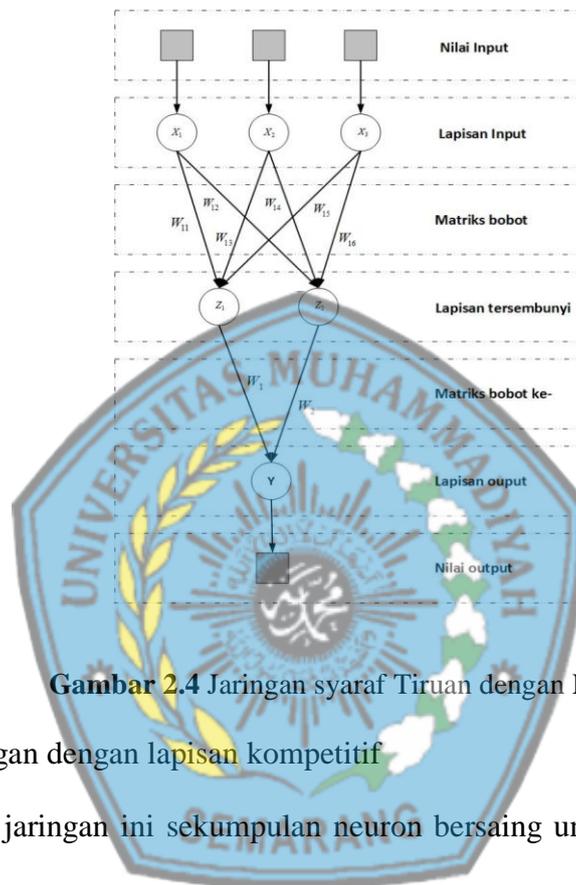
Gambar 2.3 Jaringan syaraf Tiruan dengan Lapisan Tunggal

2. Jaringan layar jamak (Multi Layer Network)

Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis layer yakni layer input, layer output, layer tersembunyi.

Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan

yang kompleks dibandingkan jaringan dengan lapisan tunggal. Namun proses pelatihan sering membutuhkan waktu yang cenderung lama. Cara kerja jaringan saraf dengan lapisan jamak digambarkan pada Gambar 2.4

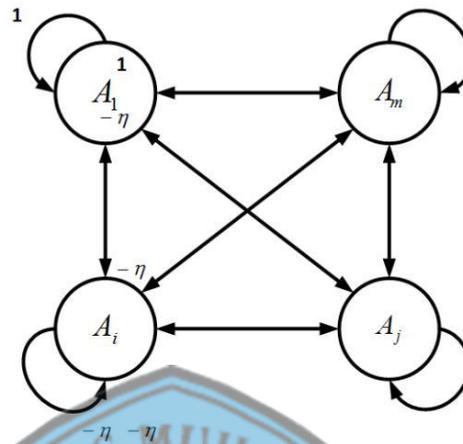


Gambar 2.4 Jaringan syaraf Tiruan dengan Lapisan Jamak

3. Jaringan dengan lapisan kompetitif

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif.

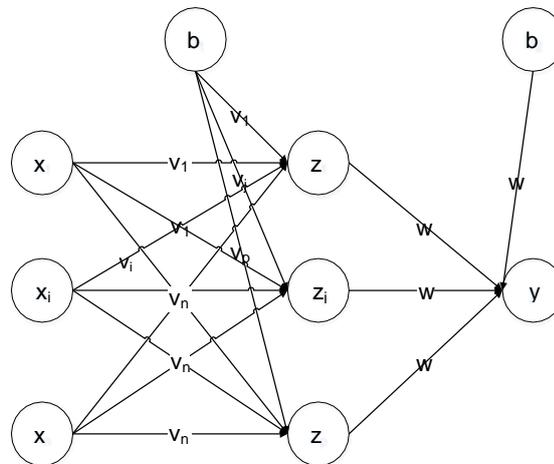
Cara kerja jaringan saraf dengan lapisan kompetitif digambarkan pada Gambar 2.5



Gambar 2.5 Jaringan syaraf Tiruan denga Lapisan Kompetitif

Desain Arsitektur Jaringan Pada Penelitian ini

Arsitektur yang akan digunakan dalam jaringan terdiri dari 3 layer yaitu input layer, hidden layer, dan output layer. Dimana pada input layer terdiri dari 12 node yaitu data aktual harga beras, hidden layer terdiri dari z_j ($j = 1, 2, \dots, 15$) dan untuk output layer terdiri dari 1 node yaitu forecast harga beras. Berikut desain arsitektur jaringannya :



Gambar 2.6 Arsitektur Jaringan syaraf Tiruan Penelitian

Secara umum dalam jaringan syaraf tiruan, mengaktivasi p unit tersembunyi di layer hidden melalui fungsi aktivasi f dan menghasilkan unit aktivasi tersembunyi z_j . Kemudian mengaktivasi unit output melalui fungsi aktivasi untuk menghasilkan output jaringan y . Secara simbolis output (keluaran) di unit y adalah sebagai berikut :

$$y = f \left(w_0 + \sum_{j=1}^p ((v_{j0} + xv_j)w_j) \right) = f \left(w_0 + \sum_{j=1}^p z_j w_j \right)$$

x = unit pada lapisan masukan (*input*)

y = unit pada lapisan keluaran (*output*)

z_j = unit ke- j pada lapisan tersembunyi (*hidden*)

f = fungsi aktivasi

p = banyak unit lapisan tersembunyi (*hidden*)

v_j = nilai penimbang bobot sambungan dari *input* ke *hidden*

w_j = nilai penimbang bobot sambungan dari *hidden* ke *output*

v_{j0} = nilai penimbang bobot sambungan pada bias untuk lapisan *hidden*

w_0 = nilai penimbang bobot sambungan pada bias untuk lapisan *output*

2.2.8 Fungsi Aktifasi

Fungsi aktivasi adalah salah satu parameter yang terpenting dalam jaringan syaraf tiruan. Performa jaringan syaraf tiruan dapat dipengaruhi oleh pemilihan fungsi aktivasi. Dalam jaringan syaraf tiruan, fungsi aktifasi dipakai untuk menentukan keluaran suatu neuron. Argumen fungsi aktivasi adalah net masukan (kombinasi linier masukan dan bobotnya). Jika $net = \sum X_i W_i$, maka fungsi

aktifasinya adalah $f(\text{net}) = f(\sum X_i W_i)$. Ada beberapa fungsi aktivasi yang sering digunakan pada jaringan syaraf tiruan (Siang, 2005: 26), yaitu :

1. Fungsi Sigmoid

Terdapat 2 buah fungsi sigmoid yaitu sigmoid biner (logsig) dan sigmoid bipolar (tansig) yang akan diperjelas dengan gambar 2.6 terkait grafik fungsi sigmoid dan fungsi bipolar.



Gambar 2.7 Grafik Fungsi Sigmoid Biner (a) dan Fungsi Sigmoid Bipolar (b)

(Hermawan, 2006:52)

Sigmoid biner memiliki nilai interval $(0,1)$ dan memiliki bentuk fungsi :

$$f(x) = \frac{1}{1+e^{-x}} \quad (1)$$

Dengan turunan

$$f'(x) = f(x)(1 - f(x)) \quad (2)$$

Sedangkan pada sigmoid bipolar bentuk fungsinya mirip dengan fungsi sigmoid biner, tetapi dengan nilai interval $(-1,1)$.

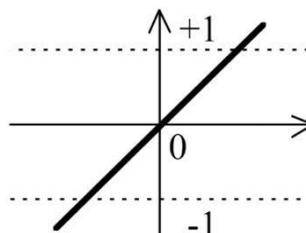
$$f(x) = \frac{1-e^{-x}}{1+e^{-x}} \quad (3)$$

Dengan turunan

$$f'(x) = \frac{1-f(x)}{2} \quad (4)$$

2. Fungsi Identitas

Fungsi identitas memiliki nilai keluaran sembarang bilangan riil, (bukan hanya pada interval $[0,1]$ atau $[-1,1]$). Dimana $f(x) = x$



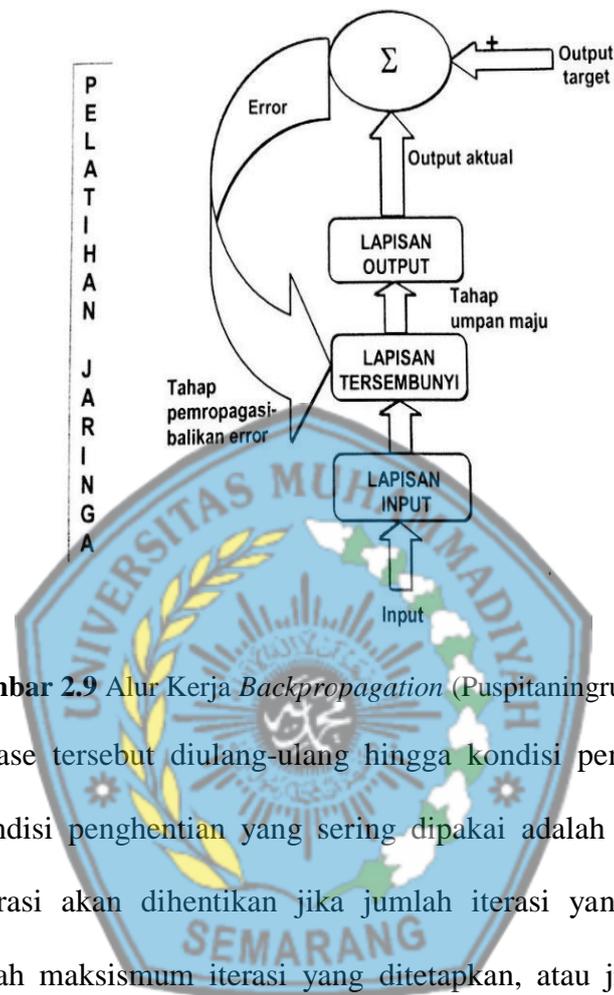
Gambar 2.8 Grafik Fungsi Identitas (Hermawan, 2006:54)

2.2.9 Backpropagation Neural Network

Algoritma backpropagation merupakan bagian dari algoritma pembelajaran terawasi yang biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyi. Algoritma ini menggunakan error keluaran untuk mengubah nilai bobot-bobotnya dalam arah mundur (backward). Untuk mendapatkan error ini tahap perambatan maju (forward propagation) harus dikerjakan terlebih dahulu. Saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat didefinisikan seperti sigmoid.

Pelatihan pada backpropagation terdiri dari 3 fase (Siang, 2005 : 100-101), yaitu (1) Propogasi maju, (2) Propogasi mundur, (3) Perubahan bobot.

Untuk lebih jelasnya, algoritma pelatihan backpropagatin dapat dilihat pada gambar 2.8 di bawah ini.



Gambar 2.9 Alur Kerja *Backpropagation* (Puspitaningrum, 2006: 127)

Ketiga fase tersebut diulang-ulang hingga kondisi penghentian dipenuhi. 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 diizinkan.

Fungsi aktivasi dalam Backpropagation harus memiliki beberapa karakteristik penting, yaitu : kontinu, terdiferensial dengan mudah, dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah fungsi sigmoid biner yang memiliki biner yang memiliki range [0,1] (Siang, 2005). Fungsi lain yang sering dipakai adalah sigmoid bipolar yang bentuk fungsinya mirip sigmoid biner, tetapi dengan range [-1,1]

Parameter-parameter yang digunakan dalam Backpropagation adalah (Azhar & Riksakomara, 2017):

1. Learning Rate

Learning Rate digunakan untuk mendefinisikan pembelajaran model untuk setiap iterasi. Skala penentuan parameter learning rate antara 0 hingga 1.

2. Epoch

Epoch adalah banyaknya jumlah iterasi yang digunakan.

3. Momentum

Momentum adalah kemiringan batas kesalahan maksimum yang boleh dihasilkan oleh model. Skala momentum yang digunakan antara 0 sampai 0,9.

2.2.10 Conjugate Gradient Beale-Powell Restarts

Algoritma conjugate gradient dikembangkan oleh E.Stiefel dan M.R Hesstenes. Pertama kali metode ini digunakan untuk menyelesaikan persamaan matriks secara iteratif (Widyastuti, 2004). Masalah optimasi nonlinier tanpa kendala merupakan pencairan nilai minimum dari fungsi bernilai $f(x)$, yaitu :

$$\min f(x), x \in R^n$$

Dimana f adalah fungsi nonlinier yang kontinu dan terdiferensialkan dengan gradien $g_x = \nabla f(x)$ masalah tersebut dapat diselsaikan secara numerik, yaitu dengan cara iteratif. Iterasi yang dijalankan dinotasikan dengan $X_k, k = 1, 2, \dots$ Setiap iterasi dideskripsikan sebagai berikut :

$$x_{k+1} = x_k + \alpha_k d_k \quad (5)$$

$\alpha_k > 0$ adalah panjang langkah (steplength) yang ditentukan oleh suatu pencarian garis d_k didefinisikan oleh :

$$d_k = \begin{cases} -g_k & \text{if } k = 1 \\ -g_k + \beta_k d_k & \text{if } k > 1 \end{cases} \quad (6)$$

Dengan $g_k = g(x_k)$ adalah gradien dari f di titik x_k , β_k adalah parameter yang jika digunakan untuk meminimumkan fungsi kuadrat yang konveks ketat, maka arah pencarian d_k dan d_{k+1} merupakan konjugat berdasarkan Hessian dari fungsi objektif (Zhang, Zhou, dan Li, 2006). (Dai, Liao, dan Li, 2004) menyatakan bahwa jika fungsi objektifnya adalah fungsi kuadrat yang konveks ketat dan menggunakan pencarian garis eksak, maka metode conjugate gradient menghasilkan arah-arah pencarian yang konjugat satu dengan lainnya. Terdapat banyak formula β_k yang terkenal, salah satunya yaitu Powell dan Beale. Formula untuk algoritma Conjugate gradient beale-powell restarts (traincgb) yaitu :

$$\beta_k = \frac{g_{k+1}^T [g_{k+1} - g_k]}{d_k^T [g_{k+1} - g_k]} \quad (7)$$

Conjugate gradient beale-powell restarts (cgb) adalah pelatihan jaringan yang memperbarui nilai bobot dan bias sesuai dengan metode backpropagation. perhitungan yang dilakukan algoritma backpropagation membutuhkan waktu yang lama dalam proses training, maka digunakan conjugate gradient beale-powell restarts untuk mempercepat kinerja dari algoritma backpropagation. metode conjugate gradient (cg) merupakan salah satu metode optimasi yang arah pencariannya di dasarkan pada arah konjugasi yang nilainya orthogonal (wanto, 2017).

Algoritma *Conjugate gradient* sebagai pelatihan jaringan syaraf tiruan dengan menggunakan fungsi aktivasi sigmoid biner adalah sebagai berikut (Adiwijaya, U.N, A & D.M, 2013).

- a. Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil.
- b. Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-13

Fase 1 : Propagasi Maju (*Feedforward*)

- c. Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya (unit lapisan tersembunyi)

- d. Langkah 4 : Menghitung semua keluaran di unit tersembunyi $Z_j (j = 1, 2, \dots, p)$

$$Z_{inj} = V_{j0} + \sum_{i=1}^n X_i V_{ji} \quad (8)$$

$$Z_j = f(Z_{inj}) = \frac{1}{1+e^{-y_{ink}}} \quad (9)$$

- e. Langkah 5 : Menghitung semua keluaran jaringan di unit $y_k (k = 1, 2, \dots, m)$

$$y_{ink} = w_{k0} + \sum_{j=1}^p Z_j w_{kj} \quad (10)$$

$$y_k = f(y_{ink}) = \frac{1}{1+e^{-y_{ink}}} \quad (11)$$

Fase II : Propagasi Mundur (*Backpropagation*)

- f. Langkah 6 : Menghitung faktor δ unit keluaran berdasarkan kesalahan disetiap unit keluaran $y_k (k = 1, 2, \dots, m)$

$$\delta_k = (t_k - y_k) f'(y_{ink}) = (t_k - y_k) y_k (1 - y_k) \quad (12)$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layer di bawahnya.

- g. Langkah 7 : Menghitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi $z_j (j = 1, 2, \dots, p)$ dengan menggunakan turunan dari fungsi aktivasi

$$\delta_{inj} = \sum_{k=1}^m \delta_k w_{kj} \quad (13)$$

Faktor δ unit tersembunyi :

$$\delta_k = \delta_{inj} f'(z_{inj}) = \delta_{inj} z_j (1 - z_j) \quad (14)$$

Menghitung suku perubahan bobot v_{kj} (yang akan dipakai nanti untuk mengubah bobot v_{kj}) dengan laju kecepatan α .

$$\Delta v_{ji} = \alpha \delta_j x_i ; j = 1, 2, \dots, m ; i = 0, 1, \dots, p$$

- h. Langkah 8 : Menghitung gradient di unit output dari fungsi objektif yang sudah ditentukan

$$g_{k+1} = \frac{1}{N} \sum_{n=1}^p \delta_{nk} y_{nk} \quad (15)$$

- i. Langkah 9 : Menghitung gradient di unit tersembunyi

$$g_{k+1} = \frac{1}{N} \sum_{n=1}^p \delta_{nj} y_{nj} \quad (16)$$

- j. Langkah 10 : Menghitung parameter β untuk semua neuron di unit tersembunyi dan unit output. Dengan menggunakan persamaan formula

Powell dan Beale :

$$\beta_k = \frac{g_{k+1}^T [g_{k+1} - g_k]}{d_k^T [g_{k+1} - g_k]} \quad (17)$$

Dimana

β_k = nilai parameter pada iterasi saat ini

g_{k+1} = gradient pada iterasi saat ini

g_k = gradient pada iterasi sebelumnya

d_k = direction pada iterasi sebelumnya

- k. Langkah 11 : Menghitung *direction* untuk semua neuron di unit tersembunyi (*hidden*) dan unit keluaran (output).

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad (18)$$

Dimana

d_{k+1} = direction pada iterasi saat ini

g_{k+1} = gradient pada iterasi saat ini

β_k = β merupakan nilai parameter pada iterasi sebelumnya

d_k = direction pada iterasi sebelumnya

Untuk direction awal :

$$d_1 = -g_1$$

- l. Langkah 12 : Menghitung parameter α untuk semua neuron di unit tersembunyi dan unit output yang merupakan seberapa besar langkah yang diambil untuk setiap *direction*. Parameter ini dapat dicari dengan teknik line search.

Fase III : Perubahan bobot (*Weight Update*)

- m. Langkah 13 : Menghitung semua perubahan bobot

$$w_{t+1} = w_t + \alpha_{t+1} d_t + 1$$

Dimana:

w_{t+1} = update bobot

w_t = bobot sebelumnya

α_{t+1} = nilai alfa saat ini

d_t = direction pada iterasi saat ini

x = unit pada lapisan masukan (*input*)

z_j = unit ke- j pada lapisan tersembunyi (*hidden*)

y = unit pada lapisan keluaran (*output*)

f = fungsi aktivasi

p = banyak unit lapisan tersembunyi (*hidden*)

v_j = nilai penimbang bobot sambungan dari *input* ke *hidden*

w_j = nilai penimbang bobot sambungan dari *hidden* ke *output*

v_j0 = nilai penimbang sambungan pada bias untuk lapisan *hidden*

$w0$ = nilai penimbang sambungan pada bias untuk lapisan *output*

z_{netj} = keluaran untuk lapisan tersembunyi

y_{net} = masukan untuk lapisan keluaran

δ_j = factor pengaturan nilai penimbang sambungan pada *hidden*

δ_k = faktor pengaturan nilai penimbang sambungan pada *output*

α = konstanta laju pelatihan (*learning rate*) $0 < \alpha < 1$

g_k = $g(x_k)$ adalah gradien dari f di titik x_k

2.2.11 Evaluasi Kinerja Peramalan

Evaluasi kinerja peramalan dilakukan setelah mendapatkan hasil prediksi melalui proses yang sebelumnya dijalankan. Peramalan atau prediksi dapat digunakan untuk mengetahui masa depan hingga mendekati realita, karena pada dasarnya masa depan adalah hal yang tidak bisa diketahui secara pasti. Penyimpangan nilai atau perbedaan nilai dengan kenyataan selalu ada sebab hasil peramalan tidak dapat dipastikan benar seluruhnya. Evaluasi kinerja peramalan dalam penelitian ini menggunakan perhitungan MSE (*Mean Square Error*). Mean Square Error adalah nilai rata-rata dari penjumlahan kuadrat kesalahan lalu dibagi dengan jumlah observasi. Berikut ini adalah rumusnya (Syariz, 2015):

$$MSE = \frac{\sum_{i=1}^n (y_t - \hat{y}_t)^2}{n} \quad (21)$$

Dimana :

n : jumlah periode peramalan

\hat{y}_t : nilai target pada periode ke t

y_t : nilai output pada periode ke t

Ukuran kesalahan kedua yang digunakan dalam penelitian ini adalah nilai rata-rata kesalahan persentase absolut (*Mean Absolute Percentage Error*). *Mean Absolute Percentage Error* (MAPE) dihitung dengan menggunakan kesalahan absolut pada tiap periode dibagi dengan nilai observasi yang nyata untuk periode itu. Kemudian, merata-rata kesalahan persentase absolut tersebut. Pendekatan ini berguna ketika ukuran atau besar variabel ramalan itu penting dalam mengevaluasi ketepatan ramalan. MAPE mengindikasikan seberapa besar kesalahan

dalam meramal yang dibandingkan dengan nilai nyata. Nilai MAPE dapat dihitung dengan rumus:

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|x_t - \hat{x}_t|}{x_t} \times 100\%$$

Dimana :

x_t : Data aktual ke t

\hat{x}_t : Data prediksi ke- t

n : Ukuran pengamatan

Semakin kecil nilai MAPE menunjukkan bahwa persentase kesalahan yang dihasilkan oleh model juga semakin kecil. Menurut Chang *et al* (2007) dalam Halimi *et al* (2013) nilai evaluasi yang dihasilkan mempunyai kriteria MAPE seperti berikut:

- a. $MAPE < 10\%$: Kemampuan peramalan sangat baik
- b. $10\% \leq MAPE < 20\%$: Kemampuan peramalan baik
- c. $20\% \leq MAPE < 50\%$: Kemampuan peramalan cukup
- d. $MAPE \geq 50\%$: Kemampuan peramalan buruk

2.2.12 Denormalisasi Data

Denormalisasi atau mengembalikan data dilakukan untuk mengkonversikan kembali hasil normalisasi (Output) yang dihasilkan oleh jaringan berkisar antara 0 hingga 1 menjadi harga material normal yang sebenarnya. Menurut (Siang, 2005)

berikut merupakan persamaan yang digunakan untuk denormalisasi data di dalam interval [0,1] :

$$x = \frac{(x_n - 0,1)(Max(x_p) - Min(x_p))}{0,8} + Min(x_p) \quad (23)$$

Dimana :

x : nilai denormaisasi

x_n : nilai data normalisasi

$Min(x_p)$: nilai minimum pada data asli

$Max(x_p)$: nilai maksimum pada data asli

