

BAB II

LANDASAN TEORI

2.1 Jaringan Saraf Tiruan

Jaringan Saraf Tiruan (*artificial neural networks*) atau disingkat JST adalah sistem komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel saraf biologi di dalam otak (Kristanto, 2004). Jaringan saraf merupakan salah satu representasi buatan dari otak manusia yang mensimulasikan proses pembelajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan saraf ini diimplementasikan dengan menggunakan program computer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran (Kusumadewi, 2003).

JST merupakan sistem pemrosesan informasi yang memiliki karakteristik serupa dengan jaringan saraf biologis dengan ciri-ciri:

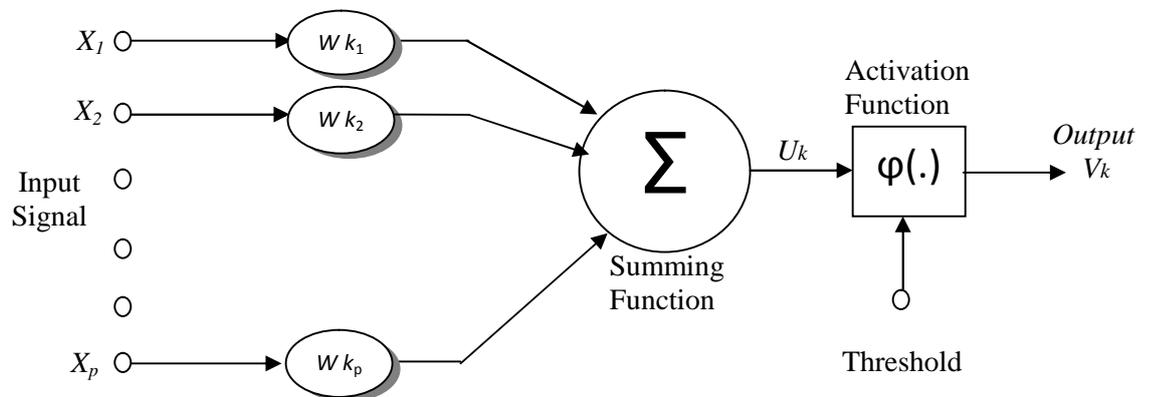
1. Pola hubungan antara elemen-elemen sederhana yakni *neuron*.
2. Metode penentuan bobot koneksi.
3. Fungsi aktivasinya.

JST mempunyai sifat dan kemampuan:

1. Akuisisi pengetahuan di bawah derau (*noise*) dan ketidakpastian (*uncertainty*).
2. Representasi pengetahuan yang fleksibel.
3. Pemrosesan pengetahuan yang efisien.
4. Toleransi kesalahan, dengan representasi pengetahuan terdistribusi dan pengkodean informasi yang redundan, kinerja system tidak menurun drastis berkaitan dengan responnya terhadap kesalahan.

2.1.1 Model Neuron

Satu sel saraf terdiri dari tiga bagian, yaitu fungsi penjumlahan (*summing function*), fungsi aktivasi (*activation function*), dan keluaran (*output*).



Gambar 2.1. Model *Neuron* (Sumber : Haykin, 2001)

Informasi (*input*) akan dikirim ke *neuron* dengan bobot tertentu. *Input* ini akan diproses oleh suatu fungsi yang akan menjumlahkan nilai bobot yang ada. Pada Gambar 2.1, hasil penjumlahan kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap *neuron*. Apabila *input* tersebut melewati suatu nilai ambang tertentu, maka *neuron* tersebut akan diaktifkan, jika tidak *neuron* tidak akan diaktifkan. Apabila *neuron* tersebut diaktifkan, maka *neuron* tersebut akan mengirimkan *output* melalui bobot-bobot *outputnya* ke semua *neuron* yang berhubungan dengannya.

Sehingga dapat disimpulkan bahwa *neuron* terdiri dari 3 elemen pembentuk yaitu :

1. Himpunan unit-unit yang dihubungkan dengan jalur koneksi. Jalur-jalur tersebut memiliki bobot yang berbeda-beda. Bobot yang bernilai positif akan memperkuat sinyal yang dibawa. Jumlah, struktur dan pola hubungan antar unit-unit tersebut akan menentukan arsitektur jaringan.
2. Suatu unit penjumlah yang akan menjumlahkan *input-input* sinyal yang sudah dikalikan dengan bobotnya.
3. Fungsi aktivasi yang akan menentukan apakah sinyal dari *input neuron* akan diteruskan ke *neuron* lain atau tidak.

2.1.2 Proses Pembelajaran

Umumnya, jika menggunakan Jaringan Saraf Tiruan, hubungan antara *input* dan *output* harus diketahui secara pasti dan jika hubungan tersebut telah

diketahui maka dapat dibuat suatu model. Hal lain yang penting adalah proses belajar hubungan *input/output* dilakukan dengan pembelajaran. Ada dua tipe pembelajaran yang dikenal yaitu :

1. Pembelajaran terawasi

Pada pembelajaran terawasi, metode ini digunakan jika *output* yang diharapkan telah diketahui sebelumnya. Biasanya pembelajaran dilakukan dengan menggunakan data yang telah ada.

2. Pembelajaran tak terawasi.

Pada metode pembelajaran yang tidak terawasi, tidak memerlukan target *output*. Pada metode ini tidak dapat ditentukan hasil seperti apa yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu range tertentu tergantung pada nilai *input* yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran seperti ini biasanya sangat cocok untuk pengelompokkan (klasifikasi) pola.

2.1.3 Fungsi Transfer

Karakter dari Jaringan Saraf Tiruan tergantung atas bobot dan fungsi *input output* (fungsi transfer) yang mempunyai ciri tertentu untuk setiap unit. Fungsi ini terdiri dari 3 katagori yaitu :

1. Untuk linear units, Aktifitas *output* adalah sebanding dengan jumlah bobot *output*.
2. Untuk threshold units, *Output* diatur satu dari dua tingkatan tergantung dari apakah jumlah *input* adalah lebih besar atau lebih kecil dari nilai ambang.
3. Untuk sigmoid units, *Output* terus menerus berubah-ubah tetapi tidak berbentuk linear. Unit ini mengandung kesamaan yang lebih besar dari sel saraf sebenarnya dibandingkan dengan linear dan threshold unit, namun ketiganya harus dipertimbangkan dengan perkiraan kasar.

Untuk membuat Jaringan Saraf Tiruan untuk melakukan beberapa kerja khusus. Harus dipilih bagaimana unit-unit dihubungkan antara satu dengan yang

lain dan harus mengatur bobot dari hubungan tersebut secara tepat. Hubungan tersebut menentukan apakah mungkin suatu unit mempengaruhi unit yang lain. Bobot menentukan kekuatan dari pengaruh tersebut.

Dapat dilakukan pembelajaran terhadap 3 lapisan pada Jaringan Saraf Tiruan untuk melakukan kerja khusus dengan menggunakan prosedur dibawah ini :

1. Memperkenalkan Jaringan Saraf Tiruan dengan contoh pembelajaran yang terdiri dari sebuah pola dari aktifitas untuk unit-unit *input* bersama dengan pola yang diharapkan dari aktifitas untuk unit-unit *output*.
2. Menentukan seberapa dekat *output* sebenarnya dari Jaringan Saraf Tiruan sesuai dengan *output* yang diharapkan.
3. Mengubah bobot setiap hubungan agar Jaringan Saraf Tiruan menghasilkan suatu perkiraan yang lebih baik dari *output* yang diharapkan.

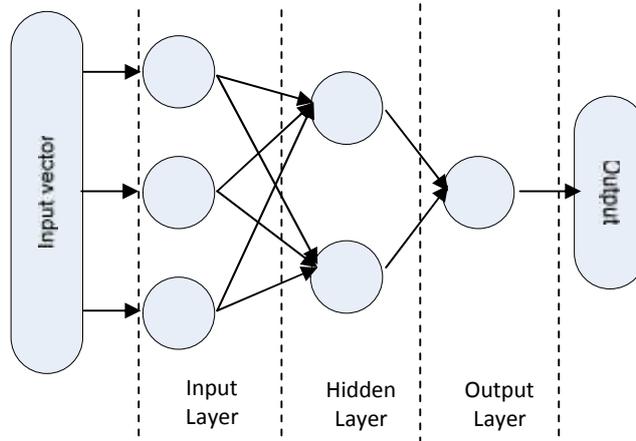
2.1.4 Arsitektur Jaringan Saraf Tiruan

Berdasarkan dari arsitektur (pola koneksi), Jaringan Saraf Tiruan dapat dibagi ke dalam dua katagori :

1. Struktur *feedforward*

Sebuah jaringan yang sederhana mempunyai struktur *feedforward* dimana signal bergerak dari *input* kemudian melewati *hidden layer* dan akhirnya mencapai unit *output* (mempunyai struktur perilaku yang stabil) seperti pada Gambar 2.2.

Tipe jaringan *feedforward* mempunyai sel saraf yang tersusun dari beberapa lapisan. Lapisan *input* bukan merupakan sel saraf. Lapisan ini hanya member pelayanan dengan mengenalkan suatu nilai dari suatu variabel. *Hidden layer* dan lapisan *output* sel saraf terhubung satu sama lain dengan lapisan sebelumnya. Kemungkinan yang timbul adalah adanya hubungan dengan beberapa unit dari lapisan sebelumnya atau terhubung semuanya (lebih baik).



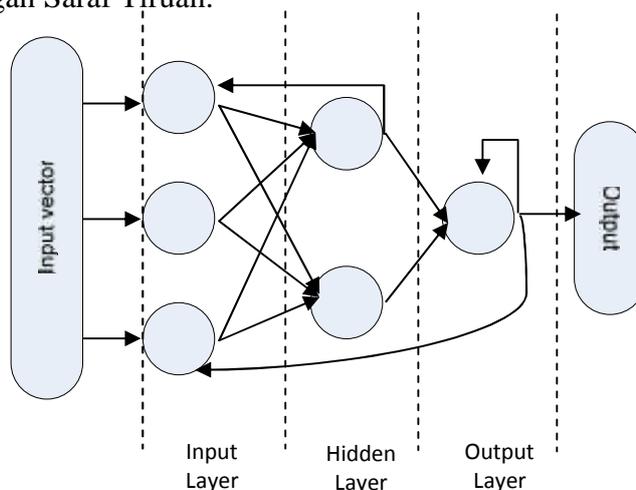
Gambar 2.2 Jaringan Saraf Tiruan *Feedforward* (Sumber : Suzuki, 2011)

Yang termasuk dalam struktur *feedforward* :

- *Single-layer perceptron*
- *Multilayer perceptron*
- *Radial-basis function networks*
- *Higher-order networks*
- *Polynomial learning networks*

2. Struktur *recurrent (feedback)*

Dalam jaringan *recurrent* mempunyai koneksi kembali dari *output* ke *input* (dari satu layer ke layer yang lain) seperti pada Gambar 2.3, hal ini menimbulkan ketidakstabilan dan dinamika yang sangat kompleks pada jaringan tersebut. Jaringan yang *recurrent* sangat menarik untuk diteliti dalam Jaringan Saraf Tiruan.



Gambar 2.3 Jaringan Saraf Tiruan *Recurrent* (Sumber : Suzuki, 2011)

Dengan *feedback* pada struktur *recurrent* dapat mempercepat proses iterasi. Adanya proses iterasi yang lebih cepat akan membuat update parameter dan kecepatan konvergensi menjadi lebih cepat.

2.2 Recurrent Neural Network

Recurrent Neural Network (RNN) adalah salah satu bagian dari model *artificial neural network*, yang mempunyai *feedback* yang berasal dari satu layer ke layer yang lain. Adapun yang termasuk dalam struktur *recurrent neural network* adalah sebagai berikut :

- *Competitive networks*
- *Self-Organizing maps*
- *Elman networks*
- *Hopfield networks*
- *Jordan networks*
- *Adaptive-resonance theory models*

2.3 Recurrent Neural Network Elman

Jaringan rekurens sederhana (*simple recurrent network*) merupakan variasi dari Multi-Layer Perceptron sering disebut juga sebagai jaringan Elman karena ditemukan oleh Jeff elman. Perbedaan utama yang terdapat pada struktur ini adalah adanya beberapa *unit* yang berdekatan dengan layer *input* yang terhubung ke *hidden layer* seperti *input unit* yang lain. *Unit-unit* tersebut mengandung isi dari salah satu layer yang ada ketika pola sebelumnya dilatih.

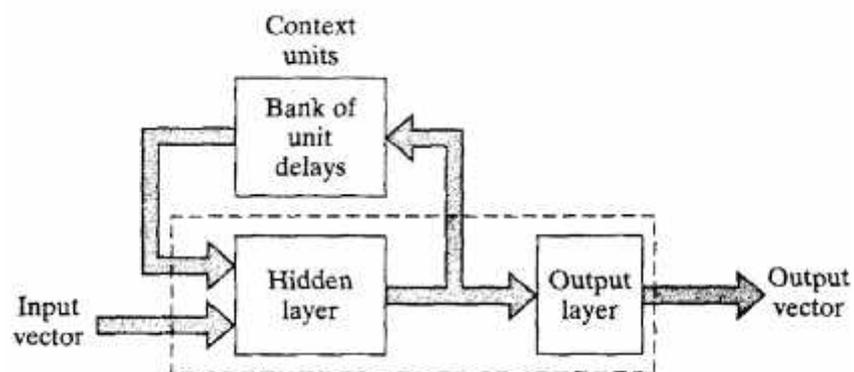
Pada tiap langkahnya, masukan disebarkan dengan cara standar feedforward lalu suatu learning rule (biasanya *back-propagation*) digunakan. Hasil koneksi balik yang tetap (*fixed back connection*) di unit konteks merupakan salinan hasil dari unit hidden sebelumnya.

RNN Elman terdiri atas satu atau lebih *hidden layer*. Lapisan pertama memiliki bobot-bobot yang diperoleh dari lapisan *input*, setiap lapisan akan menerima bobot dari lapisan sebelumnya. Jaringan ini menggunakan fungsi

aktivasi sigmoid bipolar untuk *hidden layer* dan fungsi linear (*purelin*) untuk lapisan keluaran.

Pada jaringan Elman ini, mempunyai fungsi aktivasi yang dapat berupa sembarang fungsi, baik yang kontinu maupun diskontinu. Delay yang terjadi pada hubungan antara lapisan *input* dengan *hidden layer* pertama pada waktu sebelumnya ($t-1$) dapat digunakan untuk saat ini (t) (Kusumadewi, 2004).

Keunikan RNN *Elman* adalah adanya koneksi umpan balik yang membawa informasi gangguan (*noise*) pada saat masukan sebelumnya yang akan diakomodasikan bagi masukan berikutnya dapat dilihat pada Gambar 2.4. Karena sifat dari umpan balik ini, unit dapat terus mendaur ulang informasi melalui jaringan hingga langkah-langkah beberapa waktu, dan dengan demikian menemukan representasi waktu yang abstrak. Oleh karena itu jaringan saraf *recurrent* tidak hanya merekam data masa lalu (haykin, 2001).



Gambar 2.4 Arsitektur *Simple Recurrent Network* Elman (Sumber : Haykin, 2001)

Sebuah Simple Recurrent Network (Elman, 1990) memiliki umpan balik aktivasi yang mewujudkan memori jangka pendek. Lapisan diperbarui tidak hanya dengan *input* eksternal jaringan tetapi juga dengan aktivasi dari propagasi maju sebelumnya. Umpan balik ini dimodifikasi dengan satu set bobot untuk memungkinkan adaptasi otomatis melalui belajar misalnya *backpropagation* (Boden, 2001).

Elman (1990) pada percobaannya menggunakan *backpropagation* terpotong. Dasar dari cara ini adalah menggunakan nilai keluaran dari *hidden*

layer, $y_{hidden} (t-1)$ hanya sebagai masukan tambahan. Galat dalam lapisan status hanya digunakan untuk memodifikasi bobot untuk masukan tambahan ini. Untuk menemukan batasan-batasan kata dalam aliran berkelanjutan dari fonem. Input ke jaringan mewakili fonem saat ini. *Output* mewakili menebak jaringan terbaik seperti apa fonem berikutnya secara berurutan. Peran unit konteks adalah untuk menyediakan jaringan dengan memori dinamis sehingga untuk mengkodekan informasi yang terkandung dalam urutan fonem, yang relevan dengan prediksi.

2.3.1 Algoritma Recurrent Neural Network Elman

pada waktu (t), yang aktivasi dari *context layer* adalah aktivasi (sinyal *output*) dari *hidden layer* di langkah waktu sebelumnya. *Weight* (bobot) dari *context layer* untuk *hidden layer* dilatih secara persis sama dengan bobot dari unit *input* ke unit tersembunyi. Dengan demikian, setiap langkah algoritma pelatihan sama seperti algoritma *backpropagation* standar (Fausset, 1994)

2.3.2 Propagasi Balik/ Backpropagation

Propagasi balik atau *backpropagation* merupakan suatu teknik pembelajaran/ pelatihan supervised learning yang paling banyak digunakan. Metode ini merupakan salah satu metode yang sangat baik dalam menangani masalah pengenalan pola-pola kompleks.

Di dalam jaringan propagasi balik, setiap unit yang berada di lapisan *input* terhubung dengan setiap unit yang ada di *hidden layer*. Setiap unit yang ada di *hidden layer* terhubung dengan setiap unit yang ada di lapisan *output*. Jaringan ini terdiri dari banyak lapisan (*multilayer network*). Ketika jaringan diberikan pola masukan sebagai pola pelatihan, maka pola tersebut menuju unit-unit *hidden layer* untuk selanjutnya diteruskan pada unit-unit di lapisan keluaran. Kemudian unit-unit lapisan keluaran akan memberikan respon sebagai keluaran JST. Saat hasil keluaran tidak sesuai dengan yang diharapkan, maka keluaran akan disebarkan mundur (*backward*) pada *hidden layer* kemudian dari *hidden layer* menuju lapisan masukan (Puspitaningrum, 2006).

Tahap pelatihan ini merupakan langkah untuk melatih suatu JST, yaitu dengan cara melakukan perubahan bobot. Sedangkan penyelesaian masalah akan

dilakukan jika proses pelatihan tersebut telah selesai, fase ini disebut fase pengujian (Puspitaningrum,2006).

2.3.3 Arsitektur *Backpropagation*

Setiap unit di dalam layer *input* pada jaringan *Backpropagation* selalu terhubung dengan setiap unit yang berada pada *hidden layer*, demikian juga setiap unit pada *hidden layer* selalu terhubung dengan unit pada layer *output*.

Jaringan *Backpropagation* terdiri dari banyak lapisan / *multilayer network* (Puspitaningrum,2006), yaitu :

1. Lapisan *input* (1 buah), yang terdiri dari 1 hingga n unit.
2. *hidden layer* (minimal 1 buah), yang terdiri dari 1 hingga p *hidden unit*.
3. Lapisan *output* (1 buah), yang terdiri dari 1 hingga m unit *output*.

2.3.4 Fungsi Aktivasi *Backpropagation*

Pada setiap layer pada jaringan saraf tiruan terdapat fungsi aktivasi. fungsi ini adalah fungsi umum yang akan digunakan untuk membawa *input* menuju *output* yang diinginkan. Fungsi aktivasi inilah yang akan menentukan besarnya bobot. Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan saraf tiruan. Fungsi Aktivasi yang digunakan pada *Backpropagation* antara lain :

1. Fungsi aktivasi Sigmoid Bipolar

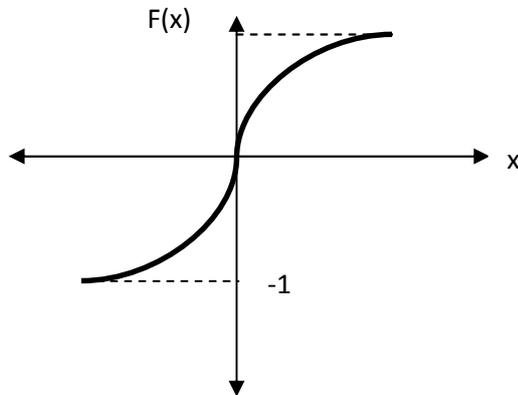
Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja *output* dari fungsi ini memiliki range antara 1 sampai -1 seperti pada Gambar

2.5. Fungsi sigmoid bipolar dirumuskan sebagai :

$$f(x) = \frac{2}{1 + e^{-x}} - 1 \quad 2.1$$

Dengan turunan fungsi :

$$F'(x) = \frac{1}{2} [1+f(x)][1-f(x)] \quad 2.2$$



Gambar 2.5 Fungsi Sigmoid Bipolar (sumber : Kristanto, 2004)

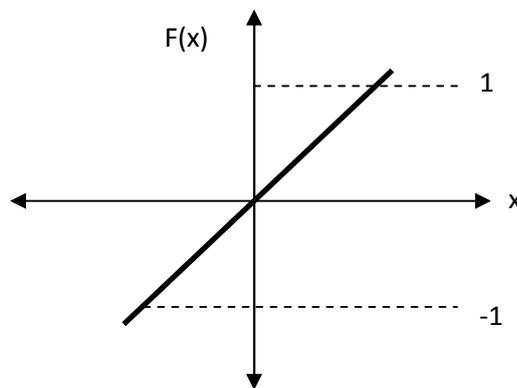
2. Fungsi Aktivasi Linear (Purelin)

Fungsi linear memiliki nilai *output* yang sama dengan nilai *input* dapat dilihat pada Gambar 2.6.

Fungsi linear dirumuskan sebagai :

$$y = x$$

2.3



Gambar 2.6 Fungsi Linear/ Purelin (Sumber : Kristanto, 2004)

2.3.5 Pelatihan *Backpropagation*

Aturan pelatihan jaringan propagasi balik terdiri dari 2 tahapan, *feedforward* dan *backward propagation*. Pada jaringan diberikan sekumpulan contoh pelatihan yang disebut set pelatihan. Set pelatihan ini digambarkan dengan sebuah *vector feature* yang disebut dengan *vector input* yang diasosiasikan dengan sebuah *output* yang menjadi target pelatihannya. Dengan kata lain set pelatihan

terdiri dari vector *input* dan juga vector *output* target. Keluaran dari jaringan berupa sebuah vector *output* actual. Selanjutnya dilakukan perbandingan antara *output* actual yang dihasilkan dengan *output* target dengan cara melakukan pengurangan diantara kedua *output* tersebut. Hasil pengurangan merupakan *error*. *Error* dijadikan sebagai dasar dalam melakukan perubahan dari setiap bobot yang ada dengan mempropagasikannya kembali (Puspitaningrum,2006).

Setiap perubahan bobot yang terjadi dapat mengurangi error. Siklus setiap perubahan bobot (*epoch*) dilakukan pada setiap set pelatihan hingga kondisi berhenti dicapai, yaitu bila mencapai jumlah epoch yang diinginkan atau hingga sebuah nilai ambang yang ditetapkan terlampaui. Algoritma pelatihan jaringan propagasi balik terdiri dari 3 tahapan yaitu (Puspitaningrum,2006) :

1. Tahap umpan maju (*feedforward*)
2. Tahap umpan balik (*backpropagation*)
3. Tahap pengupdatean bobot dan bias

Secara rinci algoritma pelatihan jaringan propagasi balik dapat diuraikan sebagai berikut :

- Langkah 0 inisialisasi bobot-bobot, konstanta laju pelatihan (η), toleransi error atau nilai bobot (bila menggunakan nilai bobot sebagai kondisi berhenti) atau set maksimal epoch (jika menggunakan banyaknya epoch sebagai kondisi berhenti).
- Langkah 1 selama kondisi berhenti belum dicapai, maka lakukan langkah ke-2 hingga langkah ke-9.
- Langkah 2 untuk setiap pasangan pola pelatihan, lakukan langkah ke-3 sampai langkah ke-8
- Tahap I *feedforward*
- Langkah 3 Setiap unit *input* x_i (dari unit ke-1 hingga unit ke-n pada lapisan *input*) mengirimkan sinyal *input* ke setiap *input* yang berada pada *hidden layer*.
- Langkah 4 masing-masing unit di *hidden layer* ($Z_j, j = 1, \dots, p$) / dari unit ke-1 hingga unit ke-p) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan dengan biasnya :

$$z_in_j = v_{oj} + \sum_{i=1}^n x_i v_{ij} \quad 2.4$$

$$z_j = f(z_in_j) \quad 2.5$$

dengan :

z_in_j adalah *input* jaringan ke z_j

v_{oj} adalah bias dari *input layer* ke *hidden layer*

x_i adalah unit *input* (masukan)

v_{ij} adalah bobot dari *input layer* ke *hidden layer*

z_j adalah sinyal *output* pada unit *hidden layer*

Langkah 5 masing-masing unit *output* ($Y_k, k=1,2,3,\dots,m$) dikalikan dengan bobot dan dijumlahkan serta ditambah dengan biasnya.

$$y_in_k = w_{ok} + \sum_{j=1}^p z_j w_{jk} \quad 2.6$$

$$y_k = f(y_in_k) \quad 2.7$$

dengan :

y_in_k adalah *input* jaringan ke y_k

w_{ok} adalah bias dari *hidden layer* ke *output layer*

z_j adalah unit *hidden layer*

w_{jk} adalah bobot dari *hidden layer* ke *output layer*

y_k adalah sinyal *output* pada unit *output layer*

Tahap II *Backpropagation*

Langkah 6 masing-masing unit *output* ($Y_k, k=1,2,3,\dots,m$) menerima pola target t_k sesuai dengan pola masukan atau *input* saat pelatihan kemudian informasi kesalahan atau error lapisan *output* (e_k) dihitung. e_k dikirim ke lapisan dibawahnya dan digunakan untuk menghitung besarnya koreksi bobot dan bias (w_{jk} dan w_{ok}) antara *hidden layer* dengan lapisan *output* :

$$e_k = (t_k - y_k) f'(y_in_k) = (t_k - y_k) y_k(1 - y_k) \quad 2.8$$

hitung suku perubahan bobot ΔW_{jk} (yang akan digunakan untuk merubah bobot W_{jk}) dengan laju pelatihan η .

$$\Delta w_{jk} = -\eta e_k z_j \quad ; k = 1,2,3,\dots,m ; j = 0,1,\dots,p \quad 2.9$$

Hitung perubahan bias

$$w_{0k} = \quad k \quad 2.10$$

Langkah 7 pada setiap unit di *hidden layer* (dari unit ke-1 hingga ke-p; $i=1, \dots, n; k=1, \dots, m$) dilakukan perhitungan informasi kesalahan *hidden layer* (δ_j). δ_j kemudian digunakan untuk menghitung besar koreksi bobot dan bias (Δv_{ji} dan Δv_{j0}) antara lapisan *input* dan *hidden layer*.

$$\delta_j = \sum_{k=1}^m w_{kj} \delta_k \quad 2.11$$

$\delta_j = \delta_j f'(z_{in_j})$ 2.12
 hitung suku perubahan bobot Δv_{ji} (yang digunakan untuk perbaikan bobot v_{ji}).

$$\Delta v_{ij} = \delta_j x_i \quad 2.13$$

Hitung perubahan bias (untuk memperbaiki V_{oj})

$$\Delta v_{0j} = \delta_j \quad 2.14$$

Tahap III Pengupdatean Bobot dan Bias

Langkah 8 masing-masing unit *output*/keluaran ($y_k, k=1, 2, 3, \dots, m$) dilakukan pengupdatean bias dan bobotnya ($j = 0, 1, 2, \dots, p$) sehingga menghasilkan bobot dan bias baru :

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad 2.15$$

Demikian juga untuk setiap unit tersembunyi mulai dari unit ke-1 sampai dengan unit ke-p dilakukan pengupdatean bobot dan bias :

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad 2.16$$

Langkah 9 : Uji Kondisi berhenti (akhir iterasi).

Pelatihan jaringan dikatakan berhasil jika pelatihan konvergen, dan gagal jika pelatihan divergen. Suatu pelatihan dikatakan konvergen jika galat pada setiap iterasi pelatihan selalu mengecil, sampai pada titik dimana nilai bobot pada setiap *neuron* telah mencapai nilai yang paling baik untuk data pelatihan yang diberikan.

Sebaliknya, pelatihan dikatakan divergen jika galat pada pelatihan tidak cenderung mengecil menuju sebuah titik tertentu.

2.3.6 Inisialisasi Nguyen-Widrow.

Inisialisasi ini umumnya mempercepat proses pembelajaran dibandingkan dengan inisialisasi acak (Fauset 1994).

Inisialisasi Nguyen-Widrow didefinisikan sebagai persamaan berikut:

- a. Hitung harga faktor pengali β

$$\beta = 0.7 (p)^{1/n} \quad 2.17$$

Dengan :

β = Faktor pengali.

n = Jumlah *neuron* lapisan *input*.

p = Jumlah *neuron hidden layer*

- b. Untuk setiap unit tersembunyi ($j=1, 2, \dots, p$):

hitung v_{ij} (lama) yaitu bilangan acak antara -0.5 dan 0.5 (atau di antara –

Dan sampai). 2.18

- c. Hitung : $\| v_j \|$

$$\| v_j \| = \sqrt{v_{1j}^2 + v_{2j}^2 + \dots + v_{nj}^2} \quad 2.19$$

Pembaharuan bobot v_{ij} (lama) menjadi v_{ij} (baru) yaitu :

$$v_{ij} \text{ (lama)} = \frac{\beta v_{ij} \text{ (lama)}}{\| v_j \text{ (lama)} \|} \quad 2.20$$

- d. Set bias :

$$v_{oj} = \text{Bilangan random antara } -\beta \text{ sampai } \beta. \quad 2.21$$

2.4 Optimasi Pembelajaran Heuristik

Pada JST *backpropagation* dikenal optimasi teknik heuristik yaitu algoritma pelatihan yang berfungsi untuk lebih mempercepat proses pelatihan dan merupakan pengembangan dari suatu analisa kinerja pada algoritma *steepest (gradient) descent standard*. Tiga algoritma optimasi teknik heuristik (Kusumadewi,2004) yang sering dipakai :

- a. *Gradient Descent Adaptive Learning Rate.*
- b. *Gradient Descent Adaptive Learning Rate and Momentum.*
- c. *Resilient Backpropagation*

Pada Tugas akhir ini akan dibahas prediksi kebangkrutan perusahaan dalam jaringan saraf tiruan dengan menggunakan algoritma *Gradient Descent Adaptive Learning Rate.*

2.4.1 *Gradient Descent Adaptive Learning Rate.*

Teknik heuristik ini memperbaiki bobot berdasarkan gradient descent dengan laju pembelajaran yang bersifat adaptive. Pada *gradient descent standard*, selama proses pembelajaran, laju pembelajaran (α) akan terus bernilai konstan. Apabila laju pembelajaran terlalu tinggi, maka algoritma menjadi tidak stabil. Sebaliknya, jika laju pembelajaran terlalu kecil maka algoritma akan sangat lama dalam mencapai kekonvergenan. Pada kenyataannya, nilai laju pembelajaran yang optimal akan terus berubah selama proses pelatihan seiring dengan berubahnya nilai fungsi kinerja.

Pada *gradient descent adaptive learning rate*, nilai laju pembelajaran akan diubah selama proses pelatihan untuk menjaga agar algoritma ini senantiasa stabil selama proses pelatihan.

Langkah- langkah teknik heuristik ini adalah :

1. Hitung bobot dan bias baru lapisan *output* dengan menggunakan persamaan:

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \alpha \cdot \Delta w_{jk}$$

$$b_{2k}(\text{baru}) = b_{2k}(\text{lama}) + \alpha \cdot \Delta b_{2k}$$

dengan :

$$w_{jk} = \text{bobot pada lapisan } \textit{output}$$

$$b_{2k} = \text{bias lapisan } \textit{output}$$

2. Hitung bobot dan bias baru *hidden layer* dengan menggunakan persamaan :

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \alpha \cdot \Delta v_{ij}$$

$$b_{1j}(\text{baru}) = b_{1j}(\text{lama}) + \alpha \cdot \Delta b_{1j}$$

dengan :

v_{ij} = bobot pada *hidden layer*

b_{1j} = bias pada *hidden layer*

3. Hitung kinerja jaringan saraf baru (perf2) dengan menggunakan bobot-bobot baru tersebut.
4. Bandingkan kinerja jaringan saraf baru (perf2) kinerja jaringan saraf sebelumnya (perf).
5. Jika $\text{perf2}/\text{perf} > \text{max_perf_inc}$ (maksimum kenaikan kerja) maka laju pembelajaran (a) = $a * \text{lr_dec}$. 2.22
6. Jika $\text{perf2}/\text{perf} < \text{max_perf_inc}$ maka laju pembelajaran (a) = $a * \text{lr_inc}$. 2.23
7. Jika $\text{perf2}/\text{perf} = \text{max_perf_inc}$ maka bobot baru diterima sebagai bobot sekarang (Kusumadewi dikutip dari Salman, 2006).

2.5 Pengujian Jaringan

Perhitungan galat (*error*) merupakan pengukuran bagaimana jaringan saraf tiruan dapat belajar dengan baik. Perhitungan galat ini merupakan pengukuran ketepatan jaringan saraf tiruan terhadap data target pembelajaran. Galat pada keluaran jaringan saraf tiruan merupakan selisih antara keluaran sebenarnya (*current output*) dengan keluaran yang diinginkan (*desired output*) dari masukan data tertentu. Pada proses pembelajaran, data yang menjadi pembanding adalah data pembelajaran, sedangkan pada proses pengujian, data yang dipakai adalah data uji. Jumlah galat pada jaringan saraf tiruan dapat dihitung dengan menggunakan jumlah galat kuadrat (SSE/ *sum of squared error*), atau rata-rata galat kuadrat (MSE /*mean square error*) ataupun dengan menggunakan akar rata-rata galat kuadrat (RMSE/ *root mean square error*).

Berikut ini adalah cara menghitung SSE, MSE dan RMSE untuk kumpulan data d , pada jaringan saraf tiruan dengan kumpulan *neuron* keluaran outputs:

$$SSE = \sum_d \sum_{k \in outputs} t_k - o_k^2 \quad 2.24$$

$$MSE = \frac{\sum_d \sum_{k \in outputs} t_k - o_k^2}{n_d n_{outputs}} \quad 2.25$$

$$RMSE = \frac{\sqrt{\sum_d \sum_{k \in outputs} (t_k - o_k)^2}}{n_d n_{outputs}} \quad 2.26$$

Dengan:

t_k adalah nilai target pada *neuron* keluaran ke- k

o_k adalah nilai output pada *neuron* keluaran ke- k

n_d adalah jumlah data pembelajaran

n_{output} adalah jumlah *neuron* keluaran

2.6 Kebangkrutan Perusahaan

Beberapa pembahasan mengenai teori pendukung ilmu ekonomi dalam penelitian ini yaitu :

2.6.1 Kesulitan Keuangan dan Kebangkrutan

Financial distress (kesulitan keuangan) terjadi sebelum kebangkrutan benar-benar dialami oleh perusahaan. Plat and Plat (2002) mendefinisikan *financial distress* sebagai tahap penurunan kondisi keuangan yang terjadi sebelum terjadinya kebangkrutan ataupun likuidasi. Berdasarkan pengertian Foster (1988) dan Plat(2002) bahwa kesulitan keuangan adalah suatu situasi dimana arus kas operasi perusahaan tidak memadai untuk melunasi kewajiban-kewajiban lancar (seperti hutang dagang atau beban bunga) dan perusahaan terpaksa melakukan tindakan perbaikan (dikutip dari Ramadhani, 2009).

Melalui informasi mengenai terjadinya *financial distress* pada laporan keuangan perusahaan maka dapat dijadikan sebagai peringatan dini atas kebangkrutan sehingga manajemen dapat melakukan tindakan secara cepat untuk mencegah masalah sebelum terjadinya kebangkrutan.

Kebangkrutan sering disebut likuidasi perusahaan atau penutupan perusahaan atau insolvensi atau pailit. Sebuah perusahaan juga dapat dikatakan bangkrut apabila nilai aktiva perusahaan tersebut sudah berada dibawah angka

nominal utang atau pinjaman. Erick L.Kohler (1994) dalam *A Dictionary Of Accountants* merumuskan bahwa pengertian kebangkrutan adalah suatu kondisi dimana yang bersangkutan tidak mampu membayar kewajibannya dan secara hukum dinyatakan bangkrut dan diharuskan untuk menjual kekayaannya/asetnya untuk membayar kewajibannya. Pekerjaan tersebut dapat dilakukan oleh debitur maupun kreditur.

Sebab kebangkrutan bisa terjadi karena perusahaan tersebut gagal atau tidak mampu lagi memenuhi kewajiban-kewajiban kepada debitur. Perusahaan mengalami kekurangan dan ketidakcukupan dana untuk menjalankan atau melanjutkan usahanya sehingga tujuan ekonomi yang ingin dicapai oleh perusahaan tidak dapat dicapai yaitu profit, sebab dengan laba yang diperoleh perusahaan bisa digunakan untuk mengembalikan pinjaman, membiayai operasi perusahaan dan kewajiban-kewajiban yang harus dipenuhi bisa ditutup dengan laba atau aktiva yang dimiliki.

2.6.2 Laporan Keuangan

Menurut Munawir dikutip dari jurnal Sarjono (2006), laporan keuangan merupakan dua daftar yang disusun oleh akuntan pada akhir periode untuk suatu perusahaan. Kedua daftar itu adalah daftar neraca atau daftar posisi keuangan dan daftar pendapatan atau daftar laba rugi. Pada waktu akhir-akhir ini sudah menjadi kebiasaan bagi perseroan-perseroan untuk menambahkan daftar ketiga yaitu daftar surplus atau daftar laba yang tak dibagikan (laba yang ditahan). Laporan keuangan tidak saja dapat menganalisa laporan keuangan masa lalu saja namun dapat memprediksi kondisi keuangan perusahaan dimasa mendatang.

2.6.3 Rasio Keuangan

Salah satu teknik yang digunakan untuk menganalisa laporan keuangan adalah dengan menggunakan rasio keuangan. Rasio keuangan dapat digunakan untuk membantu mengevaluasi laporan keuangan. Alasan utama digunakan rasio keuangan kerana laporan keuangan lazimnya berisi informasi-informasi penting mengenai kondisi dan prospek perusahaan tersebut di masa datang .

Menurut Harahap (2006:297) di dalam buku Analisis Kritis Atas Laporan Keuangan bahwa: “Rasio keuangan adalah angka yang diperoleh dari hasil

perbandingan dari satu pos laporan keuangan dengan pos lainnya yang mempunyai hubungan yang relevan dan signifikan atau berarti”. Rasio keuangan dapat menjadi suatu perhitungan rasio dengan menggunakan laporan keuangan yang dapat berfungsi sebagai alat ukur dalam menilai kinerja perusahaan. Rasio ini dapat digunakan untuk mengetahui apakah terdapat penyimpangan-penyimpangan dengan cara membandingkan rasio keuangan dengan tahun-tahun sebelumnya.

Terkait model prediksi *financial distress* dengan menggunakan rasio keuangan, menurut Rifqi (2009) ada beberapa model yang mencoba membantu calon-calon investor dan kreditur dalam memilih perusahaan tempat menaruh dana supaya tidak terjebak dalam masalah *financial distress* tersebut. Model tersebut antara lain dikemukakan oleh Beaver, Altman, Springate, Ohlson, dan Zmijewski. Model yang telah diteliti oleh Altman telah menemukan lima rasio keuangan yang dapat digunakan untuk mendeteksi kebangkrutan perusahaan beberapa saat sebelum perusahaan tersebut bangkrut.

Menurut Haryadi (2006), rasio-rasio keuangan atau variable-variabel yang dapat digunakan adalah Diskriminan Altman. Diskriminan Altman dapat digunakan sebagai sistem peringatan dini dalam hal memprediksi kinerja keuangan perusahaan, kinerja keuangan yang buruk dapat memicu kebangkrutan. Sama halnya menurut Adnan dan Kurniasih (2005), bahwa pendekatan Model Altman dapat membuktikan secara empiris rasio keuangan sehingga dapat digunakan sebagai alat untuk memprediksi kebangkrutan.

Beberapa Penelitian terkait dengan Model Altman dan model lainnya yaitu menurut Hadi dan Anggraeni (2008), bahwa model prediksi Altman merupakan prediktor terbaik di antara ketiga prediktor yang dianalisa yaitu Altman model, Zmijewski model dan Springate model, tetapi selisih dengan Springate tidak terlalu jauh. Springate model masih memberikan hasil prediksi yang lebih baik dibandingkan Zmijewski model. Yessy (2011), Model Z-Score (Altman) dapat digunakan untuk mengukur tingkat kesehatan keuangan suatu perusahaan melalui informasi dari laporan keuangan dengan tingkat nilai yang cukup akurat dibandingkan dengan model X-Score (Zmijewski). Ulfa (2007), Dalam

penelitiannya berjudul Analisis Perbedaan Prediksi Kebangkrutan Model Zevgren (Logit), Altman (Z-Score) dan Zmijewski (X-Score) Pada Perusahaan Jasa Transportasi yang Listing di Bursa Efek Jakarta Periode 2001-2005 menunjukkan bahwa akurasi dari kedua analisis ini berbeda pada waktu yang digunakan untuk memprediksi kebangkrutan perusahaan. Zmijewski (X-Score) melaporkan bahwa metodenya memiliki akurasi sebesar 85-92% dan Altman (Z-Score) melaporkan akurasi metode sebesar 95%.

Dengan demikian formula yang ditemukan Altman bisa digunakan sebagai salah satu alat ukur yang handal dalam memprediksi kebangkrutan sebuah perusahaan.

2.6.4 Analisis Diskriminan Altman

Prediksi kesulitan keuangan banyak dilakukan dengan menggunakan indikator berupa rasio-rasio keuangan. Rasio keuangan yang populer digunakan adalah model Altman. Adnan M & Taufiq M, Jurnal Ekonomi & Auditing (2005) menjelaskan, adapun variable-variabel atau rasio-rasio keuangan yang digunakan dalam analisis Diskriminant Altman adalah :

1. $X1 = \text{Net Working Capital} / \text{Total Assets}$

Rasio ini menunjukkan kemampuan perusahaan untuk menghasilkan *Net Working Capital* (modal kerja bersih) dari *Total Assets* (total aktiva) yang dimilikinya. Rasio ini dihitung dengan membagi modal kerja bersih dengan *total aktiva*. Modal kerja bersih diperoleh dengan cara aktiva lancar dikurangi dengan kewajiban lancar. Modal kerja bersih yang negatif kemungkinan akan menghadapi masalah dalam hal menutupi kewajiban jangka panjang karena tidak tersedianya aktiva lancar yang cukup.

$$\text{Metode perhitungan : } \frac{\text{net working capital}}{\text{total assets}} \quad 2.27$$

2. $X2 = \text{Retained Earnings} / \text{Total Assets}$

Rasio ini menunjukkan kemampuan perusahaan untuk menghasilkan laba ditahan dari total aktiva perusahaan. Laba ditahan merupakan laba yang tidak dibagikan kepada pemegang saham.

Metode perhitungan :
$$\frac{\textit{retained earnings}}{\textit{total assets}}$$
 2.28

3. $X3 = \textit{Earnings Before interest and tax} / \textit{Total Assets}$

Rasio ini menunjukkan kemampuan perusahaan untuk menghasilkan laba dari aktiva perusahaan sebelum pembayaran bunga dan pajak. *Earning Before Interest and Tax* atau sering disebut EBIT didapat dari laba sebelum pajak penghasilan.

Metode perhitungan :
$$\frac{\textit{EBIT}}{\textit{total assets}}$$
 2.29

4. $X4 = \textit{Market Value of Equity} / \textit{Book Value of Debt}$

Rasio ini menunjukkan kemampuan perusahaan untuk memenuhi kewajiban-kewajiban dari nilai pasar modal sendiri (saham biasa). Nilai pasar modal sendiri diperoleh dengan mengalikan jumlah lembar saham yang biasa beredar dengan harga pasar per-lembar saham biasa (*total equity*). Nilai buku hutang diperoleh dengan menjumlahkan kewajiban lancar dengan kewajiban jangka panjang (*total liabilities*).

Metode perhitungan :
$$\frac{\textit{market value of equity}}{\textit{book value of debt}}$$
 2.30

5. $X5 = \textit{Sales} / \textit{Total Assets}$

Rasio ini mencerminkan efisiensi manajemen dalam menggunakan keseluruhan aktiva perusahaan untuk menghasilkan penjualan dan mendapatkan laba.

Metode perhitungan :
$$\frac{\textit{sales}}{\textit{total assets}}$$
 2.31

Menurut Altman (1968), bahwa laporan keuangan yang paling baik untuk memprediksi kebangkrutan adalah tidak lebih dari tiga tahun sebelum kebangkrutan. Laporan keuangan satu tahun sebelum terjadi kebangkrutan adalah yang paling baik digunakan karena lebih menggambarkan kondisi sebuah perusahaan sebelum terjadinya kebangkrutan.