

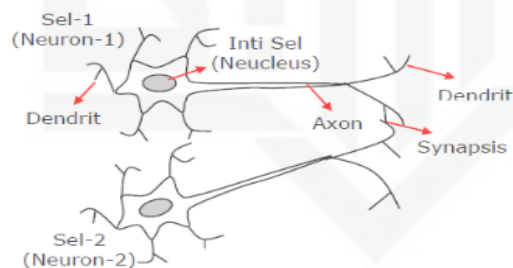
## BAB II

### LANDASAN TEORI

#### 2.1 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan merupakan bagian dari sistem kecerdasan buatan digunakan untuk memproses informasi yang didesain dengan menirukan cara kerja otak manusia dalam menyelesaikan masalah dengan melakukan proses belajar melalui perubahan bobot sinapsisnya ((Hermawan, 2006) dalam Anike, 2012).

Dalam syaraf biologis, setiap sel saraf (neuron) akan memiliki satu inti sel yang bertugas untuk melakukan pemrosesan informasi yang akan diterima oleh dendrit. Selain menerima informasi, dendrit juga menyertai axon sebagai keluaran dari suatu pemrosesan informasi. Informasi hasil olahan ini menjadi masukan bagi neuron lain. jika memenuhi batas tertentu, yang sering dikenal dengan nama nilai ambang (threshold) yang dikatakan teraktivasi ((Kusumadewi, 2010) dalam Hadjaratie, 2011 ).



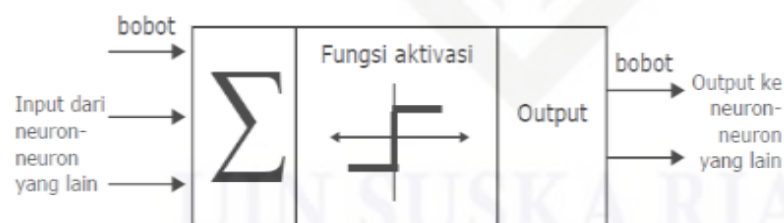
**Gambar 2.1 Syaraf Biologis**

Menurut ((Hermawan, 2006) dalam anike, 2012) karakteristik jaringan syaraf ditentukan oleh beberapa hal, yaitu :

- Pola hubungan antar neuron yang disebut dengan arsitektur jaringan
- Metode penentuan bobot-bobot sambungan yang disebut dengan pelatihan atau proses belajar jaringan
- Fungsi aktivasi

### 2.1.1 Komponen-Komponen Jaringan Syaraf Tiruan

Seperti halnya otak manusia, JST juga terdiri dari beberapa neuron yang berhubungan untuk mentransformasikan informasi yang terima melalui sambungan keluarnya. Hubungan ini dikenal dengan nama bobot (*weight*). Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Informasi (*input*) akan dikirim ke neuron dengan bobot kedatangan. *Input* ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini kemudian akan dibandingkan dengan nilai suatu ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila *input* tersebut melewati suatu nilai ambang tertentu, maka neuron tersebut akan diaktifkan, tapi kalau tidak, maka neuron tersebut tidak akan diaktifkan. Apabila neuron tersebut diaktifkan maka neuron tersebut akan mengirimkan *output* melalui bobot-bobot *output*nya ke semua neuron yang berhubungan dengannya. Neuron-neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layer*) yang saling berhubungan. Informasi akan dirambatkan mulai dari lapisan *input* sampai ke lapisan *output* melalui lapisan lainnya yang sering dikenal dengan lapisan tersembunyi (*hidden layer*), dan perambatannya tergantung algoritma pembelajarannya ((Kusumadewi, 2010) dalam Hadjaratie, 2011 ). Cara kerja neuron tersebut dapat digambarkan seperti gambar 2.2.



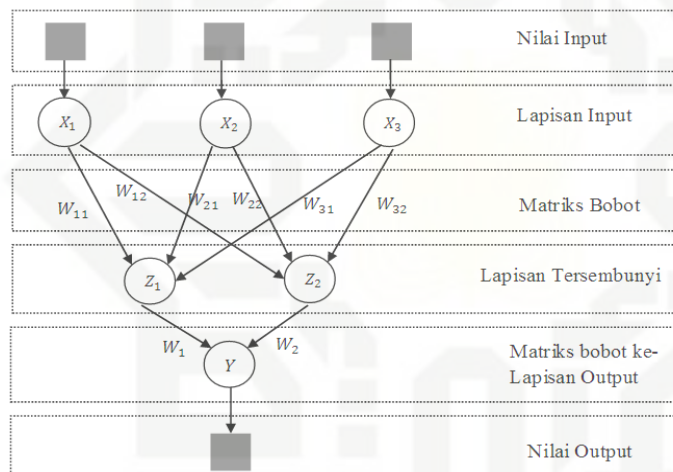
**Gambar 2.2 Struktur Neuron Jaringan Syaraf Tiruan**

### 2.1.2 Arsitektur Jaringan Syaraf Tiruan

Hubungan antar neuron pada jaringan syaraf tiruan mengikuti pola tertentu, tergantung pada arsitektur jaringan syarafnya ((Kusumadewi, 2010) dalam Hadjaratie, 2011).

## 1. Jaringan Syaraf dengan Banyak Lapisan (*Multi Layer*)

Arsitektur tipe ini memiliki satu atau lebih lapisan yang terletak di antara lapisan masukan dan lapisan keluaran, memiliki juga satu atau lebih lapisan tersembunyi. Umumnya, ada lapisan bobot-bobot yang terletak antara dua lapisan yang bersebelahan. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih sulit daripada lapisan dengan lapisan tunggal, tentu saja dengan pembelajaran yang lebih rumit juga. Contoh algoritma jaringan syaraf tiruan yang menggunakan jaringan lapisan banyak adalah *Madaline*, *Backpropagation*, dan *Neocognitron*. Cara kerja jaringan syaraf dengan banyak lapisan digambarkan pada Gambar 2.3.



**Gambar 2.3 Jaringan Syaraf Dengan Banyak Lapisan**

### 2.1.3 Fungsi Aktivasi

Ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan antara lain:

#### 1. Fungsi Sigmoid Bipolar

Fungsi ini hampir sama dengan fungsi sigmoid biner, hanya saja keluaran dari fungsi ini antara 1 sampai -1 terlihat pada Gambar 2.4. Sedangkan fungsi sigmoid bipolar dirumuskan seperti berikut ini:

$$y = f(x) = \frac{1 - e^{-x}}{1 + e^{-\sigma x}} \dots \dots \dots (2.1)$$

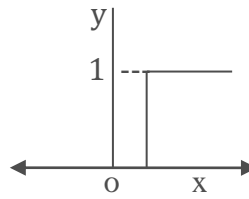
$$f'(x) = \frac{\sigma}{2} [1 + f(x)] [1 - f(x)]$$

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

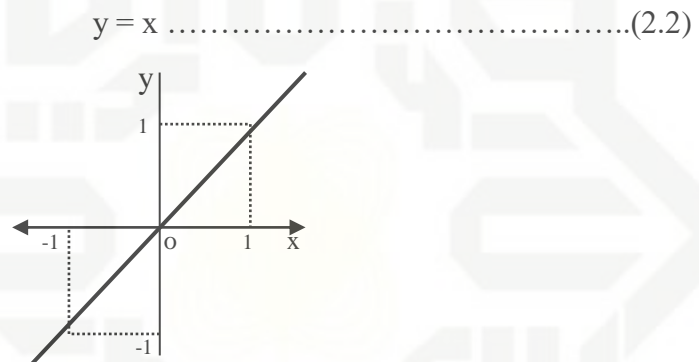
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.



**Gambar 2.4 Fungsi Aktivasi Sigmoid Bipolar**

2. Fungsi Linear (Identitas)

Fungsi linear memiliki nilai keluaran yang sama dengan nilai masukannya terlihat pada Gambar 2.5. Fungsi linear dirumuskan sebagai berikut:



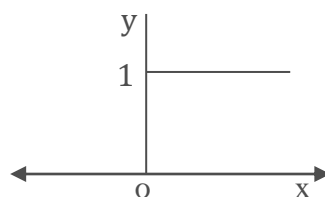
**Gambar 2.5 Fungsi Aktivasi Linear**

3. Fungsi Sigmoid Biner

Fungsi ini digunakan untuk jaringan syaraf yang dilatih menggunakan algoritma *Backpropagation*. Fungsi sigmoid biner yang tampak pada Gambar 2.6 memiliki nilai antara 0 sampai 1. Karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai keluaran yang terletak pada interval 0 sampai 1. Fungsi sigmoid biner dirumuskan sebagai berikut :

$$y = f(x) = \frac{1}{1+e^{-\sigma x}} \dots\dots\dots(2.3)$$

$$f'(x) = \sigma f(x)[1-f(x)]$$



**Gambar 2.6 Fungsi Aktivasi Sigmoid Biner**



Fungsi aktivasi yang digunakan dalam penelitian ini adalah fungsi sigmoid biner. Fungsi ini digunakan karena proses pembelajaran menggunakan algoritma *Backpropagation* membutuhkan nilai keluaran yang terletak pada interval 0 sampai 1.

#### 2.1.4 Algoritma Pembelajaran

Tujuan utama proses pembelajaran adalah melakukan pengaturan terhadap bobot-bobot yang ada pada jaringan syaraf, sehingga diperoleh bobot akhir yang tepat sesuai dengan pola data yang dilatih. Selama proses pembelajaran akan terjadi perbaikan bobot-bobot berdasarkan algoritma tertentu. Nilai bobot akan bertambah jika informasi yang diberikan oleh neuron bersangkutan tersampaikan, sebaliknya jika informasi tidak tersampaikan maka bobot akan dikurangi (Anike, 2012).

Ada dua metode pembelajaran yaitu metode pembelajaran terawasi dan pembelajaran tidak terawasi

##### 1. Pembelajaran Terawasi

Metode pembelajaran disebut terawasi bila keluaran yang diharapkan telah diketahui sebelumnya. Misalkan dimiliki jaringan syaraf untuk mengenali pasangan pola dengan operasi AND, pada proses pembelajaran satu pola masukan akan diberikan ke satu neuron pada lapisan masukan. Pola dirambatkan di sepanjang jaringan syaraf sampai dengan neuron pada lapisan keluaran. Lapisan keluaran ini akan membangkitkan pola keluaran yang nantinya akan dicocokkan dengan pola keluaran targetnya. Apabila terjadi perbedaan antara pola keluaran hasil pembelajaran dengan pola target maka terjadi *error*, apabila nilai *error* masih cukup besar mengindikasikan masih perlu dilakukan pembelajaran lagi.

##### 2. Pembelajaran Tidak Terawasi

Metode ini tidak memerlukan target keluaran karena tidak dapat ditentukan hasil yang seperti apakah yang diharapkan selama proses pembelajaran. Saat proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu tergantung pada nilai keluaran yang diberikan, apabila nilai *error* masih cukup besar mengindikasikan masih diperlukan proses pembelajaran.

## 2.2 Algoritma Pembelajaran *Backpropagation*

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 telah ditentukan (Anike, 2012).

Tahapan prosedural algoritma *Backpropagation* pada JST ada 2, yaitu :

### A. Tahap Pelatihan *Backpropagation*

Pada pelatihan *Backpropagation* meliputi 3 fase, yaitu:

#### Fase I : Propagasi maju

Selama propagasi maju, sinyal masukan ( $= x_i$ ) dipropagasikan ke layar tersembunyi menggunakan fungsi aktivasi yang ditentukan. Keluaran dari setiap unit layar tersembunyi ( $= z_i$ ) tersebut selanjutnya dipropagasikan maju lagi ke layar tersembunyi di atasnya menggunakan fungsi aktivasi yang ditentukan. Demikian seterusnya hingga menghasilkan keluaran jaringan ( $= y_k$ ). Selanjutnya keluaran tersebut dibandingkan dengan target yang harus dicapai ( $= t_k$ ). Selisih ( $t_k - y_k$ ) adalah kesalahan yang terjadi. Jika kesalahan ini lebih kecil dari batas toleransi yang ditentukan maka iterasi dihentikan. Namun, jika kesalahan masih lebih besar dari batas toleransinya, maka bobot pada setiap garis akan dimodifikasikan untuk mengurangi kesalahan yang terjadi.

#### Fase II : Propagasi mundur

Berdasarkan kesalahan ( $t_k - y_k$ ), dihitung faktor  $\delta_k = (k = 1, \dots, m)$  yang dipakai untuk mendistribusikan kesalahan di unit ke semua unit tersembunyi yang terhubung langsung dengan  $y_k$ .  $\delta_k$  juga dipakai untuk mengubah bobot garis yang berhubungan langsung dengan unit keluaran. Dengan cara yang sama, dihitung faktor  $\delta_j$  di setiap unit

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

dilayar tersembunyi sebagai dasar perubahan bobot semua garis yang berasal dari unit tersembunyi di layar dibawahnya. Demikian seterusnya hingga faktor  $\delta$  di unit yang tersembunyi yang berhubungan dengan unit masukan dihitung.

Fase III : Perubahan bobot

Setelah semua faktor  $\delta$  dihitung, bobot semua baris dimodifikasi bersamaan. Perubahan bobot suatu garis didasarkan atas faktor  $\delta$  neuron di layar atasnya.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian (iterasi) dipenuhi. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan.

Namun Sebelum data dapat dilatih, data harus dinormalisasikan terlebih dahulu melalui persamaan normalisasi. Berikut Persamaan 2.4 untuk normalisasi dan Persamaan 2.5 untuk denormalisasi.

$$\text{Normalisasi} = (X - \text{Min}) / (\text{Max} - \text{Min}) \dots\dots\dots (2.4)$$

Keterangan :

X = data

Min = data minimum

Max = data maksimum

$$\text{Denormalisasi} = Y (\text{Max} - \text{min}) + \text{min} \dots\dots\dots (2.5)$$

Keterangan :

Y = hasil keluaran dari pelatihan

Min = data minimum

Max = data maksimum

Setelah data dinormalisasi maka dilakukan pelatihan dengan algoritma *Backpropagation*. Secara rinci pelatihan *Backpropagation* dilakukan melalui langkah-langkah berikut ini :

1. Inisialisasi bobot awal (ambil bobot awal dengan nilai random yang kecil, antara 0 sampai 1)
2. Selama kondisi berhenti tidak terpenuhi, lakukan langkah 3 sampai langkah 10

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

3. Untuk setiap pasangan pola pelatihan, lakukan langkah 4 sampai langkah 9  
Fase I. Feedforward

4. Setiap unit masukan ( $x_i$ ,  $i = 1, 2, 3, \dots, n$ ) menerima sinyal  $x_i$  dan meneruskan sinyal tersebut ke seluruh unit tersembunyi

5. Masing-masing unit di lapisan tersembunyi ( $Z_j$ ,  $j = 1, 2, 3, \dots, p$ ) dikalikan dengan bobotnya dan dijumlahkan serta ditambahkan biasnya:

$$z\_in_j = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots (2.6)$$

Keterangan:

$v_{0j}$  = nilai bobot bias

$z\_in_j$  = total sinyal masukan pada lintasan  $j$

$x_i$  = nilai masukan pada unit  $i$

$v_{ij}$  = bobot antara masukan unit  $i$  dan lapisan unit  $j$

Kemudian terapkan fungsi aktivasi untuk menghitung nilai sinyal *outputnya*:

$$z_j = \frac{1}{1+e^{-z\_in_j}} \dots\dots\dots (2.7)$$

Keterangan:

$z_j$  = keluaran pada lapisan unit  $j$

$z\_in_j$  = total sinyal pada lintasan  $j$

Kemudian kirimkan sinyal tersebut ke semua unit pada unit *output*.

6. Untuk Tiap-tiap unit *output* ( $Y_k$ ,  $k = 1, 2, 3, \dots, m$ ), jumlahkan bobotnya dengan sinyal masukan masing-masing:

$$y\_in_k = w_{0k} + \sum_{i=1}^p z_i w_{jk} \dots\dots\dots (2.8)$$

Keterangan:

$w_{0k}$  = nilai bias pada layer hidden

$y\_in_k$  = total sinyal masukan pada keluaran unit

$z_i$  = nilai masukan pada lapisan unit  $i$

$w_{jk}$  = bobot antara lapisan unit  $k$  dan keluaran unit  $j$

Kemudian terapkan fungsi aktivasi untuk menghitung sinyal *outputnya*:

$$y_k = \frac{1}{1+e^{-y\_in_k}} \dots\dots\dots (2.9)$$



Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Keterangan:

$y_k$  = Keluaran pada unit k

$y_{ink}$  = Total sinyal pada lintasan k

Fase II. Backward

- Setiap unit *output* ( $Y_k$ ,  $k = 1, 2, 3, \dots, m$ ) menerima sebuah pola target yang berhubungan dengan pola masukan pelatihan, kemudian hitung kesalahannya antara target dengan *output* yang dihasilkan jaringan:

$$\delta_k = (t_k - y_k) f'(y_{ink}) \dots \dots \dots (2.10)$$

Keterangan:

$\delta_k$  = Faktor kesalahan pada unit keluaran k

$Y_k$  = Keluaran pada keluaran unit k

kemudian hitung perubahan bobotnya (digunakan nanti untuk mengubah nilai  $\Delta W_{jk}$ )

$$\Delta w_{jk} = \alpha * \delta_k * z_j \dots \dots \dots (2.11)$$

Keterangan:

$\alpha$  = *Learning Rate*

Kemudian hitung perubahan biasnya (digunakan untuk memperbaiki nilai  $W_{0k}$ )

$$\Delta w_{0k} = \alpha * \delta_k \dots \dots \dots (2.12)$$

Kirimkan  $\delta_k$  ini ke unit-unit lapisan kanan.

- Pada setiap unit pada lapisan tersembunyi ( $Z_j$ ,  $j = 1, 2, 3, \dots, p$ ) menjumlahkan nilai delta inputnya (dari neuron pada dilapisan atasnya):

$$\delta_{in_j} = \sum_{k=1}^m \delta_k w_{jk} \dots \dots \dots (2.13)$$

Kalikan nilai ini dengan turunan dari fungsi aktivasinya untuk menghitung informasi *error*:

$$\delta_j = \delta_{in_j} * z_j * (1 - z_j) \dots \dots \dots (2.14)$$

Kemudian hitung koreksi bobot (digunakan untuk mengubah nilai  $V_{ij}$ ) dengan persamaan berikut:

$$v_{ij} = \alpha * \delta_j * x_i \dots \dots \dots (2.15)$$

Kemudian hitung perubahan biasnya (digunakan untuk mengubah nilai  $V_{0j}$ ) dengan persamaan berikut:

$$\Delta v_{0j} = \alpha * \delta_j \dots\dots\dots (2.16)$$

### Update Bobot dan Bias

9. Untuk setiap neuron pada lapisan keluaran ( $Y_k$ ,  $k=1,2,\dots, m$ ) ganti nilai bobot biasnya ( $j=0,1,2,\dots,p$ )

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots\dots\dots (2.17)$$

Untuk setiap neuron pada lapisan tersembunyi ( $Z_j$ ,  $j=1,2,\dots,p$ ) ganti nilai bobot dan biasnya ( $i=0,1,2,\dots,n$ ) yang menuju keluaran lapisan. jumlahkan bobot masukan(lama) dengan jumlah koreksi bobot pada unit  $j$  dan  $i$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta V_{ij} \dots\dots\dots (2.18)$$

10. Pelatihan pola ini dilakukan secara berulang-ulang dengan menggunakan data pelatihan dan parameter yang telah ditentukan. Bobot-bobot baru ini digunakan sebagai bobot awal pada proses *epoch* berikutnya, proses ini dilakukan sampai sampai maksimum *epoch*.

### **B. Tahap Pengujian *Backpropagation***

Pengujian *Backpropagation* dilakukan melalui *feedforward*. Langkah - langkahnya sebagai berikut :

Langkah 0 : Inisialisasikan bobot (dari hasil pelatihan)

Langkah 1 : Untuk setiap vektor *input*, kerjakan langkah 2 - 4

Langkah 2 : Untuk  $i=1,\dots,n$ : set aktivasi unit *input*  $x_i$

Langkah 3 : Untuk  $j=1,\dots,p$ :

$$z_{in_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \dots\dots\dots (2.19)$$

$$z_j = \frac{1}{1+e^{-z_{in_j}}} \dots\dots\dots (2.20)$$

Langkah 4 : Untuk  $k=1,\dots,m$ :

$$y_{in_k} = w_0 + \sum_{i=1}^p z_j w_{jk} \dots\dots\dots (2.21)$$

$$y_k = \frac{1}{1+e^{-y_{in_k}}} \dots\dots\dots (2.22)$$

### 2.3 Backpropagation Pada Peramalan

Untuk dapat mengetahui hasil kinerja algoritma *Backpropagation* dapat diukur dengan melihat nilai akurasi dan *error*. Perhitungan akurasi dilakukan dengan persamaan akurasi sedangkan *error* akan diukur dengan *Mean Square Error* (MSE).

#### 2.3.1 Mean Square Error (MSE)

Mean Square Error (MSE) adalah fungsi kineja yang sering digunakan untuk metode *Backpropagation* yang dimana fungsi ini akan mengambil rata-rata kuadrat error yang terjadi antara output jaringan dan target (Sinuhaji, 2009).

Adapun rumus untuk menghitung MSE adalah:

$$MSE = \frac{\sum e_i^2}{n} \dots\dots\dots (2.23)$$

$e_i^2$  = Selisih antara nilai target dengan hasil peramalan

$n$  = jumlah data

#### 2.3.2 Akurasi

Pengujian akurasi untuk persen ketepatan peramalan produksi Tandan Buah Segar ini dilakukan dengan persamaan mengurangi persentase keseluruhan data dengan tingkat kesalahan (Fattah, 2016).

$$Akurasi = 100\% - \left( \frac{x}{Target} \right) \times 100\% \dots\dots\dots (2.24)$$

$\bar{x}$  = selisih target dari hasil peramalan

### 2.4 Peramalan

Peramalan adalah perhitungan yang objektif dengan menggunakan data-data masa lalu, untuk menentukan sesuatu di masa yang akan datang. Untuk melakukan peramalan memerlukan ilmu pengetahuan statistik dan teknologi ((Sumayang, 2003) dalam Syamsuar, 2014)

Langkah-langkah yang harus dilakukan dalam menjamin efisiensi untuk melakukan peramalan adalah sebagai berikut :

- Menentukan tujuan peramalan
- Memilih item yang akan diramalkan
- Menentukan rentang waktu peramalan

- d. Memilih model peramalan
- e. Mengumpulkan dan menganalisis data
- f. Validasi model peramalan
- g. Membuat peramalan
- h. Implementasi hasil peramalan
- i. Memantau kehandalan hasil peramalan

Teknik pengumpulan data yang relevan sangat diperlukan dalam sebuah peramalan agar hasil yang didapatkan sesuai dengan yang diharapkan. Menurut ((Jumingan ,2009) dalam Jannati, 2015) teknik peramalan berdasarkan sifatnya dapat dibedakan menjadi dua yaitu :

1. Teknik peramalan kualitatif

Teknik kualitatif merupakan teknik peramalan yang bersifat subjektif berdasarkan pendapat dari suatu pihak atau berdasarkan hasil penelitian questioner yang telah dilakukan. Data pada teknik ini tidak dapat direpresentasikan secara tegas ke dalam suatu angka atau nilai.

2. Teknik peramalan kuantitatif

Teknik kuantitatif merupakan teknik peramalan berdasarkan data masa lalu atau data historis dan dapat dibuat dalam bentuk angka.

## 2.5 Produksi Tandan Buah Segar (TBS)

Kelapa sawit termasuk tanaman daerah tropis yang tumbuh baik antara garis lintang  $13^0$  Lintang Utara dan  $12^0$  Lintang Selatan, terutama di kawasan Afrika, Asia, dan Amerika Latin. Indonesia menjadi salah satu negara yang cocok untuk pembudidayaan kelapa sawit karena struktur tanah dan curah hujannya. Bagian yang paling utama pada industri kelapa sawit adalah buah dari kelapa sawit (TBS). Bagian daging buah menghasilkan minyak kelapa sawit mentah yang diolah menjadi bahan baku minyak goreng. Produksi TBS kelapa sawit merupakan hasil panen buah kelapa sawit dari suatu area perkebunan kelapa sawit. Produksi TBS kelapa sawit merupakan sumber penghasilan utama dalam perusahaan yang bergerak di bidang perkebunan sawit. Penyusunan biaya produksi diperlukan oleh perusahaan agar mendapatkan produksi TBS kelapa sawit yang tinggi. Jika produksi TBS kelapa sawit tinggi maka keuntungan



perusahaan dari penjualan hasil produksi akan meningkat. Penyusunan biaya produksi perusahaan memerlukan peramalan produksi kelapa sawit sebagai target produksi agar anggaran yang dikeluarkan oleh perusahaan menjadi optimal karena pengeluaran perusahaan disesuaikan berdasarkan kemungkinan produksi perusahaan dalam setahun (Jannati, 2015).

Kacaribu dkk., (2013) menyatakan bahwa variabel curah hujan, jumlah pohon, umur rata-rata berpengaruh signifikan terhadap besar kecilnya produksi kelapa sawit. Sedangkan dosis pupuk tidak berpengaruh signifikan terhadap produksi kelapa sawit.

Menurut Manalu (2008) Produktivitas kelapa sawit sangat erat hubungannya dengan curah hujan. Faktor curah hujan berpengaruh dalam hal penyerapan unsur hara oleh akar, membantu perkembangan bunga betina, membantu pemasakan buah menjadi lebih sempurna dan berpengaruh terhadap berat janjang .

Yohansyah dan Lubis (2014) menyatakan bahwa hari hujan berpengaruh nyata terhadap produktivitas tanaman kelapa sawit dan memiliki nilai koefisien regresi yang negatif sebesar -0.0366 yang berarti bahwa setiap bertambahnya 1 hari hujan, produktivitas kelapa sawit akan menurun sebesar -0.0366 ton/ha dengan asumsi variabel lain dianggap konstan. Satu hari hujan adalah periode 24 jam terkumpulnya curah hujan setinggi 500 mm atau lebih dan curah hujan dengan tinggi kurang dari ketentuan tersebut, hari hujan dianggap nol tetapi curah hujan tetap diperhitungkan. Frekuensi hari hujan yang rendah akan menyebabkan terjadinya defisit air. Defisit air sangat berpengaruh dalam produksi tandan buah segar kelapa sawit karena berpengaruh terhadap pembungaan.

Berdasarkan beberapa penelitian diatas dapat disimpulkan bahwa produktivitas kelapa sawit dipengaruhi oleh beberapa faktor, yaitu curah hujan, hari hujan, jumlah pohon, rata-rata umur pohon. Variabel ini berpengaruh signifikan terhadap besar kecilnya jumlah produksi TBS yang dihasilkan.

## 2.6 Kajian Pustaka

Beberapa penelitian yang berkaitan dengan peramalan produksi TBS kelapa sawit dan algoritma *Backpropagation* :

**Tabel 2.1 Penelitian terkait peramalan produksi TBS**

No	Tahun	Nama	Judul Penelitian	Hasil Penelitian
1	2015	Rini Jannati	Prediksi produksi panen kelapa sawit menggunakan jaringan syaraf Radial Basis Function (RBF)	Jumlah variabel = 1, <i>input node</i> = 5 , nilai <i>learning rate</i> = 0.75, dan nilai <i>epoch</i> maksimal = 3000 dengan hasil <i>error</i> MAPE= 11.75%.
2	2013	Riski Agustian Kacaribu, Margaretha Ohyver, Bayu Kanigoro,	Aplikasi peramalan produksi kelapa sawit dengan metode regresi ganda dan exponential smoothing	MAPE regresi = 21.068% MAPE ES pertahun = 12.78%. Dari 4 variabel umur rata-rata, jumlah pohon, curah hujan, dan dosis pupuk, hanya variabel dosis pupuk yang tidak signifikan berpengaruh terhadap besar kecilnya produksi kelapa sawit.
3.	2013	Aindra Budiar	Peramalan produksi TBS kelapa sawit dengan model arima dan model fungsi transfer <i>input</i> ganda	Selisih MAPE dan MAD sebesar 0.53 % dan 7040.34 kg.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

**Tabel 2.2 Penelitian terkait algoritma *Backpropagation***

No	Tahun	Nama	Judul Penelitian	Hasil Penelitian
1.	2015	Lestari Handayani, Muhammad Adri	Penerapan JST ( <i>Backpropagation</i> ) untuk Prediksi Curah Hujan.	Menggunakan parameter <i>epoch</i> 1000, <i>learning rate</i> 0.01 dan <i>error</i> 0.001 diperoleh akurasi sebesar 96%.
2.	2013	Yudha Tirto Pramonoaji, Stefanus Santosa, Ricardus Anggi Pramunendar	prediksi produksi air PDAM dengan Jaringan Syaraf Tiruan <i>Backpropagation</i>	Nilai <i>error</i> 0.0000, MSE 0.074416, MAE 0.102487, dan rata-rata nilai akurasi sebesar 95,56%.
3.	2013	Candra Dewi, M.Muslikh	Perbandingan akurasi <i>Backpropagation</i> neural network dan anfis untuk memprediksi cuaca.	BPNN memiliki tingkat akurasi yang tinggi jika dibandingkan dengan ANFIS dan memiliki nilai RMSE yang lebih rendah.
4.	2010	Andrijasa, Mistianingsih	Penerapan jaringan syaraf tiruan untuk memprediksi jumlah pengangguran diprovinsi kalimantan timur.	RMSE sebesar 13.700152. hasil prediksi jumlah pengangguran tahun 2009 adalah 133.104. sedangkan hasil prediksi pengangguran tahun 2009 yang dilakukan oleh BPS adalah 139.830.