

Hak Cipta Dilindungi Undang-Undang

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

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. 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.

## BAB II

### LANDASAN TEORI

#### 2.1 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan adalah suatu sistem pemrosesan informasi yang terinspirasi oleh system saraf secara biologis, seperti proses informasi pada otak manusia. Cara kerja jaringan syaraf tiruan seperti cara kerja manusia, yaitu belajar melalui contoh. Sebuah JST dikonfigurasi untuk aplikasi tertentu, seperti pengenalan pola atau klasifikasi data, melalui proses pembelajaran. Dengan adanya kemampuan belajar, maka diharapkan jaringan syaraf tiruan ini dapat menyelesaikan masalah layaknya manusia. Pada proses pembelajaran jaringan syaraf tiruan ini dimasukkan pola-pola input atau output dan selanjutnya jaringan akan mendapatkan ajaran untuk dapat memberikan jawaban yang bisa diterima oleh pengguna (Sutojo dkk, 2011).

Dalam penerapannya JST memiliki kelebihan dan kekurangan (Sutojo dkk, 2011). Kelebihan dari JST adalah:

1. Belajar *Adaptive*, yaitu mampu melakukan pembelajaran untuk pekerjaan berdasarkan pengalaman awal yang dimiliki atau untuk pelatihan berdasarkan data yang diberikan.
2. *Self-Organisation*, yaitu mampu membuat organisasi secara tersendiri berdasarkan informasi yang diterima selama proses pembelajaran.
3. *Real Time Operation*, yaitu mampu melakukan proses perhitungan secara paralel sehingga dapat diperoleh keuntungan dari kemampuan ini berdasarkan perangkat keras yang telah dirancang dan di produksi secara khusus.

Selain memiliki kelebihan, JST juga memiliki kekurangan, yaitu:

2. Menjadi tidak efektif untuk digunakan sebagai operasi numerik dengan presisi yang tinggi.
3. Menjadi tidak efisien jika digunakan untuk perhitungan operasi algoritma aritmatik, operasi logika, dan simbolis.

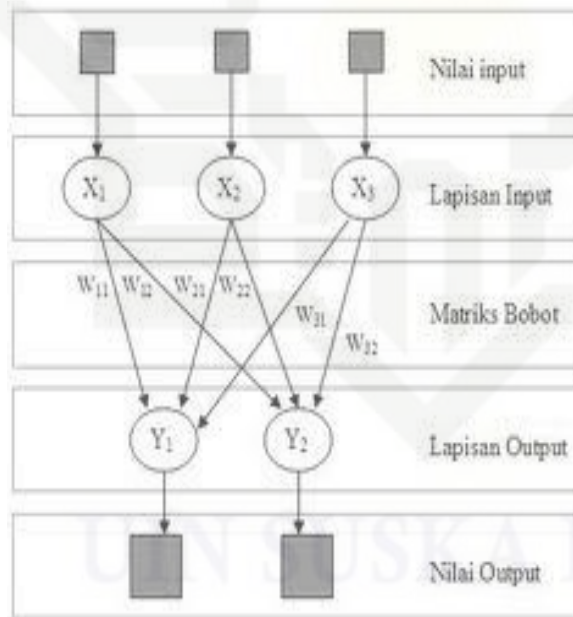
- Proses pengoperasian JST untuk jumlah data yang banyak atau besar, maka akan membutuhkan waktu yang sangat lama untuk proses pelatihannya.

### 2.1.1 Arsitektur Jaringan Syaraf Tiruan

Dalam JST memiliki beberapa arsitektur jaringan yang sering digunakan dalam berbagai aplikasi. Arsitektur JST tersebut, antara lain (Sutojo dkk,2011 ) :

#### 1. Jaringan Layan Tunggal (*Single Layer Network*)

Jaringan dengan lapisan tunggal terdiri dari 1 layer input dan 1 layer output. Setiap neuron/unit yang terdapat di dalam lapisan/layer input selalu terhubung dengan setiap neuron yang terdapat pada layer output. Jaringan ini hanya menerima input kemudian secara langsung akan mengolahnya menjadi output tanpa harus melalui lapisan tersembunyi. Contoh algoritma JST yang menggunakan metode ini yaitu : ADALINE, Hopfield, Perceptron.

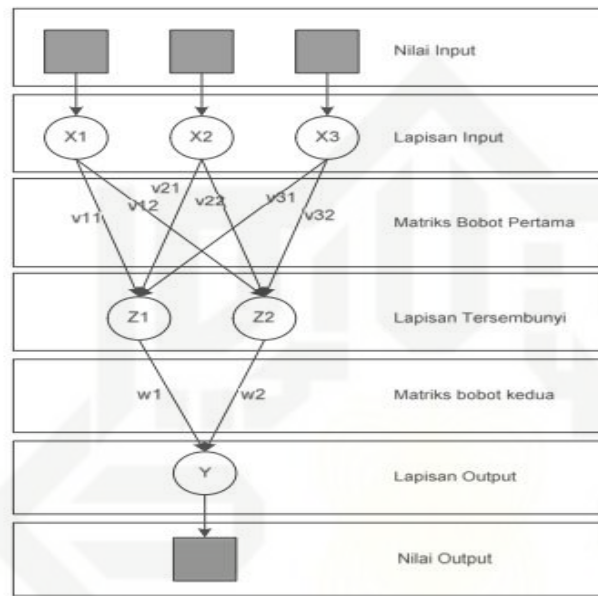


**Gambar 2. 1 Jaringan syaraf dengan lapisan tunggal (Sutojo dkk, 2011)**

#### 2. Jaringan Banyak Lapisan (*Multilayer Net*)

Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis *layer* yakni *layer input*, *layer output*, dan juga *layer tersembunyi*. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih

kompleks dibandingkan jaringan dengan lapisan tunggal. Namun, proses pelatihan sering membutuhkan waktu yang cenderung lama. Contoh algoritma Jaringan Syaraf Tiruan yang menggunakan metode ini yaitu : MADALINE, *backpropagation*, *Neocognitron* dan *Elman Recurrent Neural Network*.



**Gambar 2. 2 Jaringan syaraf dengan banyak lapisan (Sutojo dkk, 2011)**

### 3. Jaringan Lapisan Kompetitif (*Competitive Layer*)

Pada jaringan ini memiliki bobot yang telah ditentukan dan tidak memiliki proses pelatihan. Adapun alasan menggunakan jaringan ini ialah dapat digunakan untuk mengetahui neuron pemegang dari sejumlah neuron yang ada.. Contoh algoritma yang menggunakan metode ini adalah LVQ.

#### 2.1.2 Proses Pembelajaran Jaringan Saraf Tiruan

Proses pembelajaran Jaringan Saraf Tiruan dapat diklasifikasikan menjadi 2 bagian (Desiani, 2006), yaitu :

- Supervised Learning* (Pembelajaran Terawasi) yang menggunakan sejumlah pasangan data masukan dan keluaran yang diharapkan. Contoh tipe ini adalah metode *BackPropagation*, *Elman Recurrent Neural Network* (ERNN), jaringan *hopfield* dan perceptron.
- Unsupervised learning* (Pembelajaran tak terawasi) yang hanya menggunakan sejumlah pasangan data masukan tanpa ada contoh keluaran

Hak Cipta Dilindungi Undang-Undang

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

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. 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.

yang diharapkan. Contoh metode pembelajaran tak terawasi adalah jaringan kohonen (kohonen network).

## 2.2 Backpropagation Neural Network (BPNN)

BPNN adalah salah satu metode pada jaringan syaraf tiruan yang merupakan pembelajaran lanjut dari aturan metode perceptron. Algoritma ini diterapkan pada perceptron berlapis banyak (*multilayer perceptron*). Multilayer perceptron ini terdiri dari bagian input, bagian output dan lapisan diantara input dan output yang disebut dengan lapisan tersembunyi (*hidden layer*) (Widiastuti dkk, 2014).

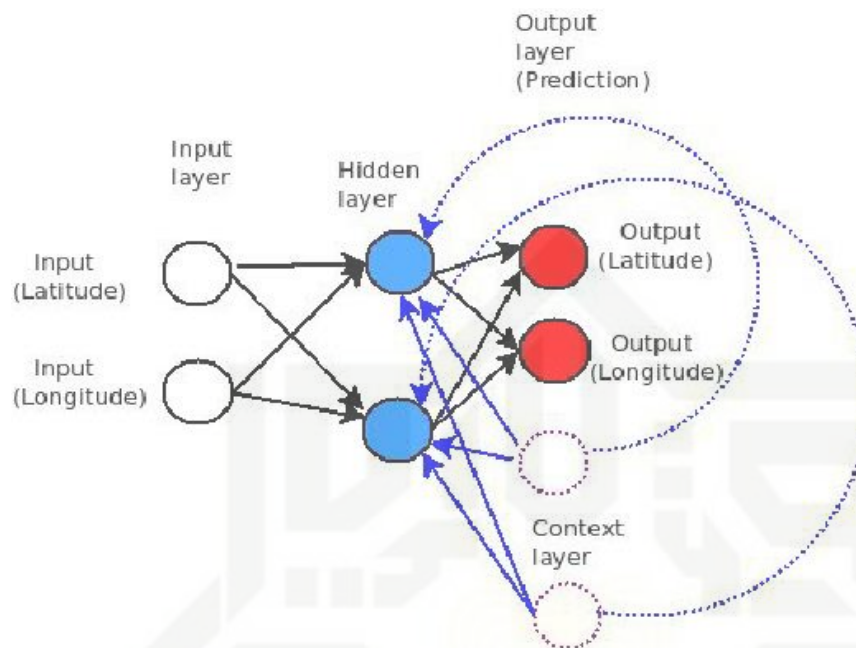
Backpropagation merupakan salah satu algoritma pembelajaran terawasi (*supervised learning*) dan digunakan oleh perceptron dengan banyak lapisan (*multilayer perceptron*) untuk mengubah bobot-bobot yang terhubung dengan neuron yang ada pada hidden layer. Algoritma *backpropagation* menggunakan *error* output untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error*, tahap perambatan maju (*forward propagation*) harus dikerjakan dahulu. Backpropagation merupakan sebuah teknik spesifik untuk implementasi penurunan gradien bobot pada multilayer perceptron (Haykin, 2009).

## 2.3 Elman Recurrent Neural Network (ERNN)

Elman Recurrent Neural Network (ERNN) adalah modifikasi *feed forward* dengan tambahan *layer neuron* yang berhubungan menyediakan pola keluaran jaringan untuk diumpan balik ke dirinya sendiri menjadi masukan dalam rangka menghasilkan keluaran jaringan berikutnya (Talahatu dkk, 2015)

ERNN 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. Fungsi aktivasi yang digunakan yaitu sigmoid biner pada *hidden layer* (lapisan tersembunyi) dan fungsi identitas (purelin) pada lapisan output (Maulida, 2011). Berikut gambar 2.3 *Arsitektur Elman Recurrent Neural Network*.





**Gambar 2. 3** Arsitektur ERNN (Sundaram dkk, 2015)

ERNN memiliki empat layer yaitu *input layer*, *hidden layer*, *output layer* dan *context layer*. ERNN dilatih secara *supervised* dengan menggunakan algoritma *Backpropagation* berdasarkan masukan dan target yang diberikan. ERNN memiliki kelebihan dari *Backpropagation* yaitu mempunyai *context layer* yang dapat mengingat state sebelumnya dari *hidden layer*. Adanya *context layer* juga dapat membuat iterasi dan kecepatan update parameter jauh lebih cepat (Sundaram dkk, 2015). Pada tabel 2.1 juga dijelaskan perbedaan ERNN dengan *Backpropagation*.

**Tabel 2. 1** Perbedaan ERNN dengan *Backpropagation*

<i>Elman Recurrent Neural Network (ERNN)</i>	<i>Backpropagation Neural Network</i>
4 layer (input, hidden, output, context)	3 layer (input, hidden , output)
Terdapat <i>feedback</i> di hidden	Tidak ada

### 2.3.1 Algoritma Elman Recurrent Neural Network (ERNN)

Secara umum langkah dalam pelatihan ERNN sama dengan *Backpropagation* biasa, hanya saja dalam ERNN menggunakan nilai keluaran dari hidden layer,  $y_{hidden}(t-1)$  sebagai masukan tambahan yang disebut *context layer*.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Galat dalam hidden layer hanya digunakan untuk modifikasi bobot untuk masukan tambahan ini (Maulida, 2011).

**Feed Forward**

1. Memeberi nilai awal secara random bagi seluruh weight antara input-hidden layer dan hidden layer-output.
2. Tiap unit hidden layer  $net_j(t)$  ditambah dengan input  $x_i$  yang dikali dengan bobot  $v_{ji}$  dan dikombinasikan dengan context layer  $y_h(t-1)$  yang dikali dengan bobot  $u_{jh}$  dijumlah dengan bias  $\theta$ .

$$y_h = \left( \sum_i^n x_i(t) v_{ji} \right) \dots \dots \dots (2.1)$$

$$net_j(t) = \left( y_h + \sum_h^m y_h(t-1) u_{jh} \right) + \theta_j \dots \dots \dots (2.2)$$

Keterangan:

- $x_i$  = input dari 1,.....,n
- $v_{ji}$  = bobot dari input ke *hidden layer*
- $y_h$  = hasil *copy* dari *hidden layer* waktu ke(t-1)
- $u_{jh}$  = bobot dari *context* ke *hidden layer*
- $\theta$  = bias
- n = jumlah *node* masukan
- i = *node input*
- m = jumlah *node hidden*
- h = *node context*

3. fungsi pengaktif neuron yang digunakan sigmoid biner

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

4. Tiap unit output  $y_k$  ditambah dengan nilai keluaran *hidden layer*  $y_j$  yang dikali bobot  $w_{ji}$  dan dijumlah dengan bias bagian hidden layer  $\theta$ . Untuk mendapatkan keluaran, maka  $net_k$  dihitung dalam fungsi pengaktif menjadi  $y_k$ .

$$net_k(t) = \left( \sum_j^m y_j(t)w_{kj} \right) + \theta_k \dots\dots\dots(2.4)$$

$$y_k(t) = g(net_k(t)) \dots\dots\dots(2.5)$$

Keterangan:

$y_j$  = hasil fungsi  $net_j$

$w_{kj}$  = bobot dari *hidden* ke *output layer*

$\theta_k$  = bias

$y_k$  = hasil fungsi  $net_k$

$g(net_k(t))$  = fungsi  $net_k(t)$

**Elman Backpropagation**

5. Tiap unit output menerima pola target  $t_k$  sesuai dengan pola masukan saat pelatihan dan dihitung error-nya dan diperbaiki nilai bobot-nya.

Perhitungan Error dalam turunan Fungsi pengaktif

$$\delta_k = g'(net_k)(t_k - y_k) \dots\dots\dots(2.6)$$

Keterangan:

$g'(net_k)$  = fungsi turunan  $g(net_k)$

$t_k$  = target

$y_k$  = hasil fungsi  $g(net_k)$

Menghitung perbaikan bobot

$$\Delta\theta_k = \alpha\delta_k y_j \dots\dots\dots(2.7)$$

Keterangan:

$\Delta w_{kj}$  = perbaikan nilai bobot dari *hidden* ke *output layer*

$\alpha$  = konstanta *learning rate* / laju pembelajaran

Menghitung perbaikan kolerasi

$$\Delta\theta_k = \alpha\delta_k \dots\dots\dots(2.8)$$

Keterangan:

$\Delta\theta_k$  = hasil perbaikan nilai bias

Dan menggunakan nilai  $\delta_k$  pada semua unit lapisan sebelumnya.

6. Tiap bobot yang menghubungkan unit output dengan unit hidden layer dikali  $\delta_k$  dan dijumlahkan sebagai masukan unit berikutnya.

$$\delta_{net_j} = \sum \delta_k w_{kj} \dots \dots \dots (2.9)$$

Selanjutnya dikalikan dengan turunan dari fungsi aktivasi untuk menghitung galat.

$$\delta_j = \delta_{net_j} f'(net_j) \dots \dots \dots (2.10)$$

Keterangan:

$f'(net_j)$  = fungsi turunan  $net_j$

Kemudian menghitung perbaikan bobot

$$\Delta v_{kj} = \alpha \delta_j x_i \dots \dots \dots (2.11)$$

Keterangan:

$\Delta v_{kj}$  = hasil perbaikan nilai bobot

Menghitung perbaikan kolerasi

$$\Delta \theta_j = \alpha \delta_j \dots \dots \dots (2.12)$$

Keterangan:

$\Delta \theta_j$  = hasil perbaikan nilai bias

7. Tiap unit output diperbaiki bobot dan biasnya

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \dots \dots \dots (2.13)$$

Keterangan:

$w_{kj}(\text{baru})$  = nilai bobot baru dari input ke hidden layer

$w_{kj}(\text{lama})$  = nilai bobot lama dari input ke hidden layer

Tiap unit hidden layer diperbaiki bobot dan biasnya

$$v_{kj}(\text{baru}) = v_{kj}(\text{lama}) + \Delta v_{kj} \dots \dots \dots (2.14)$$

Keterangan:

$v_{kj}(\text{baru})$  = nilai bobot baru dari hidden ke output layer

$v_{kj}(\text{lama})$  = nilai bobot lama dari hidden ke output layer

8. Tiap output dibandingkan dengan target  $t_k$  yang diinginkan, untuk memperoleh error (E) keseluruhan

$$E(t) = \frac{1}{2} \sum_{i=1}^k (t_k - y_k)^2 \dots \dots \dots (2.15)$$

Keterangan:



$E(t)$  = hasil nilai *error* keseluruhan

9. Lakukan pengujian kondisi pemberhentian (akhir iterasi).

Proses pelatihan yang dikatakan berhasil yaitu apabila nilai *error* pada saat iterasi pelatihan nilainya selalu mengecil hingga diperoleh nilai bobot yang baik pada setiap neuron untuk data pelatihan yang diberikan. Sedangkan proses pelatihan yang dikatakan tidak berhasil yaitu apabila nilai *error* pada saat iterasi pelatihan tidak memberikan nilai yang cenderung mengecil.

### 2.3.2 Fungsi Aktivasi

Fungsi aktivasi (batas ambang atau *threshold*) merupakan fungsi untuk menentukan nilai keluaran (*output*) berdasarkan nilai total masukan (*input*) pada *neuron* (Kusumadewi, 2004). Beberapa fungsi aktivasi yang sering dipakai adalah:

#### 1. Fungsi Sigmoid Biner

Pada fungsi sigmoid biner, nilai output terletak pada interval range 0 sampai dengan 1. Dapat didefinisikan sebagai berikut (sayekti dkk,2012) :

$$y = f(x) = \frac{1}{1+e^{-x}} \quad (2.16)$$

Dengan turunan

$$y' = f'(x) = f(x)[1 - f(x)] \quad (2.17)$$

#### 2. Fungsi Sigmoid Identitas (*Purelin*)

Fungsi identitas digunakan jika menginginkan keluaran jaringan berupa sembarangan bilangan real (bukan hanya pada interval [0,1] atau [-1,1]). Artinya fungsi aktivasi *purelin* digunakan untuk menghasilkan nilai output yang mendekati target yang diinginkan dengan formula yaitu nilai output sama dengan nilai input (sayekti dkk,2012)

$$f(x) = y \quad (2.18)$$

### 2.3.3 Normalisasi

Normalisasi adalah proses transformasi nilai menjadi kisaran 0 dan 1. Normalisasi merupakan proses penskalaan nilai atribut dari data sehingga bisa jatuh pada range tertentu. Tujuan dari normalisasi data adalah untuk mendapatkan data dengan ukuran yang lebih kecil, mewakili data asli tanpa kehilangan karakteristiknya. Rumus dari normalisasi yaitu:

$$X^* = \frac{X - \min(X)}{\max(X) - \min(X)} \quad (2.19)$$

Keterangan:

$X^*$  = nilai setelah dinormalisasi

$X$  = nilai sebelum dinormalisasi

$\min(X)$  = nilai minimum dari fitur

$\max(X)$  = nilai maksimum dari fitur

### 2.3.4 Denormalisasi

Denormalisasi adalah proses dimana data dikembalikan ke nilai asalnya. Berikut persamaan 2.19 untuk proses denormalisasi.

$$\text{Denormalisasi} = Y(\text{Max} - \text{Min}) + \text{min} \dots \dots \dots (2.20)$$

### 2.3.5 Mean Square Error (MSE)

MSE merupakan cara yang digunakan untuk mengukur tingkat kesalahan dan error pada sebuah model peramalan (sanny dkk, 2013).

$$MSE = \frac{\sum ET^2}{n} \dots \dots \dots (2.21)$$

Dengan:

$x_t$  : nilai aktual pada waktu ke-t

$f_t$  : nilai dugaan pada waktu ke-t

$n$  : jumlah data yang diprediksi

### 2.3.6 Pengujian Tingkat Akurasi

Akurasi adalah kedekatan hasil pengukuran atau rata-rata hasil pengukuran ke nilai yang sebenarnya (Permana dkk, 2014). Rumus akurasi adalah sebagai berikut :

$$Akurasi = 100 - \left(\frac{y-t}{t}\right) \times 100 \dots \dots \dots (2.22)$$

$$Rata - Rata Akurasi = \frac{jumlah\ data\ uji\ sesuai\ target}{jumlah\ data\ uji} \times 100\% \dots \dots \dots (2.23)$$

## 2.4 Peramalan Permintaan Koran

Peramalan adalah suatu usaha untuk meramalkan keadaan di masa mendatang melalui pengujian keadaan di masa lalu, sedangkan peramalan permintaan merupakan tingkat permintaan produk yang diharapkan akan terealisasi untuk jangka waktu tertentu pada masa yang akan datang (Nasution, 2008). Peramalan merupakan bagian awal dari suatu proses pengambil keputusan. Dalam melakukan peramalan ada hal yang harus terlebih dahulu diketahui yaitu apa sebenarnya permasalahan dalam pengambilan keputusan ini. Peramalan adalah pemikiran terhadap suatu besaran, misalnya permintaan terhadap satu atau beberapa produk pada periode yang akan datang. Pada dasarnya peramalan hanya merupakan suatu perkiraan, setiap pengambilan keputusan yang menyangkut keadaan di masa yang akan datang, maka pasti ada peramalan yang melandasi pengambilan keputusan tersebut (Firdausi dkk, 2015).

Dalam melakukan peramalan permintaan ada beberapa faktor yang mempengaruhi hasil peramalan permintaan (Munawar, 2003) yaitu :

1. Volume Penjualan  
 Volume penjualan adalah jumlah penjualan yang tercapai dalam suatu perusahaan.
2. Biaya Promosi  
 Biaya promosi adalah biaya yang dikeluarkan dalam mempromosikan suatu produk dalam rangka untuk meningkatkan penjualan produk.
3. Harga Jual  
 Harga jual merupakan harga suatu produk.
4. Harga Jual Pesaing  
 Harga jual pesaing mempengaruhi peramalan permintaan karna dapat membandingkan harga jual suatu perusahaan. Dengan demikian dapat melihat banyaknya permintaan dengan mengetahui harga jual.

**Hak Cipta Dilindungi Undang-Undang**

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

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

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

## 5. Biaya Distribusi

Biaya distribusi merupakan biaya yang dikeluarkan dalam mendistribusikan produk pada suatu perusahaan.

Lima faktor tersebut saling berkaitan dan mempengaruhi peramalan permintaan. Dalam penelitian ini ada beberapa faktor yang mempengaruhi hasil peramalan permintaan Koran. Faktor tersebut diinisialisasikan ke dalam *variabel-variabel* yaitu produksi, harga, biaya promosi, biaya distribusi dan pengembalian.

Sebuah perusahaan sangatlah penting melakukan prediksi agar meminimalisir kerugian. JST dapat digambarkan sebagai sebuah simulasi dari korelasi model syaraf biologi. Model syaraf ditunjukkan dengan kemampuannya dalam emulasi, analisa, prediksi, dan asosiasi (Kristanto, 2004). Hal tersebut merupakan salah satu alasan menggunakan ilmu Jaringan Syaraf Tiruan dalam memprediksi atau meramalkan permintaan Koran Haluan Riau.

## 2.5 Kajian Pustaka

Kajian pustaka berisikan tentang penelitian sebelumnya yang berkaitan dengan penelitian yang dilakukan, dapat dilihat pada Tabel 2.1:

**Tabel 2. 2 Tabel Referensi Penelitian Terkait**

No	Penulis & Tahun	Judul	Metode	Kesimpulan
1.	Sayyidina Abdul Fattah RA (2016)	Implementasi Jaringan Syaraf Tiruan (JST) Backpropagation untuk Peramalan Permintaan Koran (Studi Kasus: PT.RIAU POS Media Group Pekanbaru	<i>Backpropagation</i>	Penelitian ini menghasilkan akurasi terbaik pada persentasi data 90% data latih dan 10% data uji dengan akurasi 95, 21% dan nilai learning rate = 0.5.
2.	Julian Talahatu, Njoto Benarkah dan Jimmy (2015)	Penggunaan Aplikasi Sistem Jaringan Syaraf Tiruan Berulang Elman untuk Prediksi Pergerakan Saham	<i>Berulang Elman (Elman Recurrent Neural Network)</i>	Konfigurasi jaringan dapat memberikan hasil yang berbeda. Aplikasi yang dibuat dapat memberikan hasil prediksi yang lebih stabil dibandingkan dengan menggunakan library encog, namun program



Hak Cipta Dilindungi Undang-Undang

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

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

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

				yang dibuat dengan library encog memiliki waktu proses yang lebih sedikit.
3.	Agus Aaan Jiwa Permana, Widodo Prijodiprodjo (2014)	Sistem Evaluasi Kelayakan Mahasiswa Magang Menggunakan <i>Elman Recurrent Neural Network</i>	<i>Elman Recurrent Neural Network</i>	Hasil tingkat akurasi yang didapat dengan menggunakan metode Elman Recurrent Neural Network adalah 90,91% dengan menggunakan momentum 0,85.
4.	Rafika Putri (2014)	Aplikasi Prediksi Kebangkrutan dengan Implementasi Jaringan Syaraf Tiruan <i>Recurrent Model Elman</i>	<i>Recurrent model Elman (Elman Recurrent Neural Network)</i>	Penerapan Jaringan Syaraf Tiruan <i>Recurrent model Elman</i> ini telah berhasil untuk memberikan kemungkinan atas kondisi perusahaan berdasarkan laporan keuangan masing-masing perusahaan, ini dapat dilihat dari hasil pengujian dengan menggunakan 10 unit <i>hidden layer</i> yang mana 9 dari 10 perusahaan yang diuji mendekati kondisi perusahaan yang sebenarnya.
5.	Dea Lucky Sani (2014)	Penerapan Elman- <i>Recurrent Neural Network</i> Pada Peramalan Konsumsi Listrik Jangka Pendek di PT. PLN APP Malang	<i>Elman Recurrent Neural Network</i>	Metode Elman-RNN menghasilkan jaringan terbaik untuk meramalkan konsumsi listrik per jam di Blimbing dengan hasil RMSE testing 147.04, dan menghasilkan MAPE sebesar 2,97% yang merupakan nilai MAPE terkecil.
6.	Ana Maulida	Penggunaan <i>Elman Recurrent Neural Network</i> dalam peramalan suhu udara sebagai faktor yang mempengaruhi Kebakaran Hutan	<i>Elman Recurrent Neural Network</i>	Penelitian ini menggunakan parameter hidden node 5 dan LR 0,3 menghasilkan RMSE 0,51 dan MAPE 1,55%.