

# Peramalan Jumlah Penjualan Menggunakan Jaringan Syaraf Tiruan *Backpropagation* Pada Perusahaan Air Minum Dalam Kemasan

<http://dx.doi.org/10.28932/jutisi.v5i2.1607>

Nur Fitrianiingsih Hasan ✉ #1, Kusriani\*2, Hanif Al Fatta#3

<sup>1</sup>Magister Teknik Informatika, Universitas AMIKOM Yogyakarta  
Jalan Ring Road Utara, Condong Catur, Sleman, Yogyakarta, Telp: (0274) 884201 – 207

1

<sup>2</sup>Magister Teknik Informatika, Universitas AMIKOM Yogyakarta  
Jalan Ring Road Utara, Condong Catur, Sleman, Yogyakarta, Telp: (0274) 884201 – 207  
<sup>2</sup> kusriani@amikom.ac.id

<sup>3</sup>Magister Teknik Informatika, Universitas AMIKOM Yogyakarta  
Jalan Ring Road Utara, Condong Catur, Sleman, Yogyakarta, Telp: (0274) 884201 – 207  
<sup>3</sup> hanif.a@amikom.ac.id

**Abstract** — The inhibition of the production and distribution of bottled water has become a serious problem in survival community and company, so a solution is needed for this problem, both short and long term solutions. One of the things that can be done by the company or management is the right amount of production and distribution by estimating sales. Estimated sales are the process of predicting which products will be sold in the future based on data that ever happened. This paper aims to determine the level of accuracy of Backpropagation ANN in estimating the sale of bottled drinking water. The ANN architecture used is 12-10-1 with MSE value 0,00043743 and MAPE value 6.88%. There is a difference from the actual data with the forecasting result which is equal to 2% but this is normal because in forecasting activities, the results are an approach.

**Keywords**— Artificial Neural Network, Backpropagation, Bottled Water, Sales Forecasting.

## I. PENDAHULUAN

Air merupakan kebutuhan hidup yang paling penting. Kebutuhan akan air bersih yang dapat dikonsumsi harus melalui tahap pengelolaan yang baik[1]. Dalam memenuhi kebutuhan air minum yang aman untuk di konsumsi, masyarakat pada umumnya mendapatkannya dari 2 sumber yaitu air yang dimasak / diolah sendiri dan air minum dalam kemasan (AMDK)[2].

Berdasarkan keputusan Menteri Perindustrian Republik Indonesia Nomor 11/M-IND/PER/3/2017 telah memberikan definisi yang jelas mengenai AMDK, yaitu air yang telah

diproses tanpa bahan pangan lainnya dan bahan tambahan pangan, dikemas, dan aman untuk diminum. Seiring dengan meningkatnya jumlah penduduk dari tahun ke tahun, tentunya kebutuhan AMDK akan semakin meningkat. Menurut Asosiasi Perusahaan Air Minum dalam Kemasan Indonesia atau ASPADIN (2015), pada tahun 2014 masyarakat Indonesia mengkonsumsi 23,1 miliar liter air minum dalam kemasan untuk dikonsumsi, jumlah tersebut meningkat 11,3% dari permintaan di tahun 2013 yang hanya sebesar 20,48 miliar liter air per tahun [3].

Perusahaan Air Minum ABC yang berada di Kota Jayapura merupakan salah satu BUMD di Jayapura yang mengelola dan mendistribusikan Air. Perusahaan ABC memiliki salah satu unit bisnis yang berkonsentrasi pada produksi dan pendistribusian AMDK. Sumber air yang digunakan Perusahaan ABC saat ini semuanya merupakan air permukaan atau air sungai. Sumber air tersebut memiliki daerah tangkapan air di pegunungan Cyclop, Sentani. Namun sejak 2018, secara kapasitas terjadi penurunan debit air terutama pada musim kemarau, tetapi jika masuk musim hujan justru terjadi peningkatan kekeruhan air. Produksi AMDK kemudian ikut terganggu karena banjir bandang yang baru saja terjadi di Sentani awal tahun 2019, beberapa pipa saluran air milik Perusahaan ABC mengalami kerusakan sehingga produksi dan pendistribusian AMDK menjadi terhambat, sehingga perlu adanya langkah solusi untuk masalah ini baik solusi jangka pendek maupun jangka panjang.

Salah satu yang dapat dilakukan perusahaan ataupun pihak manajemen agar jumlah produksi dan pendistribusian tepat adalah dengan melakukan *forecasting sales* (peramalan penjualan). Peramalan adalah seni dan ilmu memprediksi peristiwa-peristiwa yang akan terjadi dengan menggunakan data historis dan memroyeksikannya ke masa depan dengan beberapa bentuk model matematis[4]. Peramalan penjualan merupakan proses aktivitas memperkirakan produk yang akan dijual di masa mendatang dalam keadaan tertentu dan di buat berdasarkan data yang pernah terjadi dan mungkin akan terjadi [5].

Banyak teknik dan metode yang diterapkan dalam proses peramalan. Dalam melakukan peramalan diperlukan perhitungan yang akurat sehingga diperlukan peramalan yang tepat [6]. Metode Jaringan Syaraf Tiruan (JST) dengan algoritma Backpropagation dalam ilmu AI dapat diterapkan untuk meramal penjualan dengan menentukan arsitektur jaringan yang tepat sehingga hasil peramalan dapat mendekati nilai aktual. Jaringan perambatan galat mundur (*backpropagation*) merupakan salah satu algoritma JST yang dapat digunakan dalam menyelesaikan masalah-masalah rumit berkaitan dengan identifikasi, peramalan, pengenalan pola dan sebagainya.

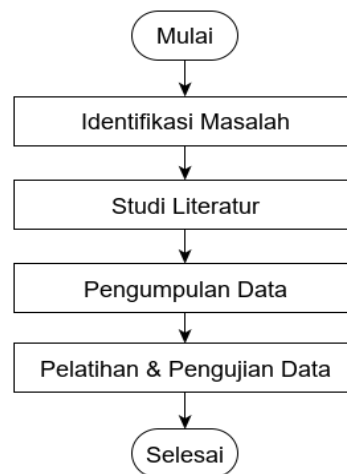
Pada penelitian sebelumnya JST *Backpropagation* digunakan dalam meramal penjualan pada perusahaan retail [7], meramalkan penjualan mobil menggunakan JST *Backpropagation* dan di kombinasikan dengan validasi *Certainty Factor*[8], penerapan JST *Backpropagation* untuk meramal penjualan Coklat dan membandingkan hasil peramalan dengan data *testing* dan peramalan dengan data *time series* hasilnya lebih baik dengan data *testing* JST *Backpropagation*[9], memprediksi jumlah produksi daging sapi menggunakan JST *Backpropagation*[10].

Penelitian yang akan dilakukan bertujuan untuk mengetahui tingkat akurasi dari penggunaan JST *Backpropagation* dalam meramal penjualan AMDK pada Unit Bisnis AMDK Perusahaan ABC Kota Jayapura menggunakan JST *Backpropagation*. Data yang dijadikan sebagai data prediksi adalah data penjualan produk AMDK 600ml tahun 2017-2018. Data penjualan tahun 2017 digunakan untuk data *training* kemudian data tahun 2018 digunakan sebagai data *testing*. *Testing* dilakukan untuk menguji arsitektur yang dipilih untuk peramalan. Arsitektur yang digunakan adalah arsitektur peramalan penjualan AMDK yang telah dilakukan analisis pada penelitian [11] yang menghasilkan arsitektur terbaik untuk peramalan penjualan AMDK.

Hasil penelitian ini diharapkan dapat menghasilkan peramalan yang baik sehingga dapat digunakan Unit Bisnis AMDK Perusahaan ABC Kota Jayapura sebagai dasar dalam pengambilan keputusan untuk menentukan jumlah produksi AMDK.

## II. METODE PENELITIAN DAN LANDASAN TEORI

Pada penelitian ini dilakukan beberapa tahapan penelitian dimulai dari identifikasi masalah, Studi literatur yaitu mencari teori-teori yang mendukung untuk penelitian ini, mengumpulkan data penelitian dan melakukan analisa terhadap data *training* dan *testing* untuk kebutuhan peramalan penjualan dengan algoritma *backpropagation*. Berikut alur penelitian dapat dilihat pada gambar 1.



Gambar 1. Alur Penelitian

Dalam penelitian ini terdapat beberapa teori yang menjadi landasan yaitu mengenai peramalan penjualan dan semua pembahasan teori, aplikasi, serta tahapan dalam metode JST *Backpropagation*.

### A. Peramalan Penjualan

Peramalan penjualan merupakan proses aktifitas memperkirakan produk yang akan dijual di masa mendatang dalam keadaan tertentu dan di buat berdasarkan data yang pernah terjadi atau data masa lalu[5].

Banyak teknik dan metode yang diterapkan dalam proses peramalan. Banyak teknik dan metode peramalan. Dalam melakukan peramalan diperlukan perhitungan yang akurat sehingga diperlukan peramalan yang tepat [6]. Metode peramalan di bagi menjadi 2 yaitu metode statistika dan metode jaringan syaraf. Terdapat 2 pendekatan untuk melakukan peramalan yaitu dengan pendekatan kualitatif dan pendekatan kuantitatif [12].

### B. Jaringan Syaraf Tiruan

Definisi jaringan syaraf tiruan (JST) menurut [13], JST adalah system komputasi dimana arsitektur dan operasi diilhami dari pengetahuan tentang sel syaraf biologis di dalam otak manusia. JST dapat digambarkan sebagai sebagai model matematis dan komputasi untuk fungsi aproksimasi nonlinear, klasifikasi data, cluster, prediksi dan regresi. Menurut [14] dengan kemampuan JST yang sangat baik, beberapa aplikasi JST memang sangat cocok untuk

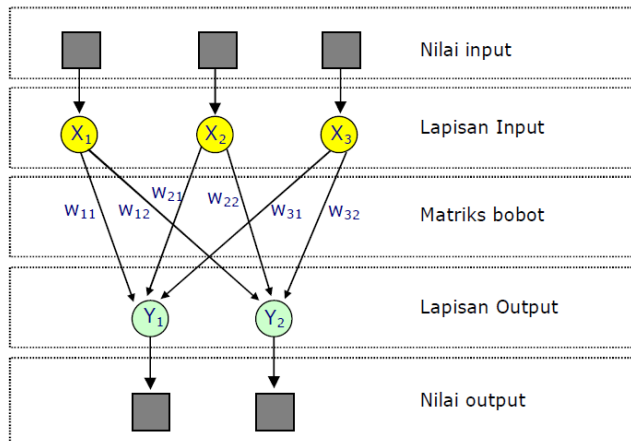
diterapkan pada kasus klasifikasi, asosiasi, prediksi, optimasi dan *self-organizing*.

JST juga merupakan pemodelan data yang kuat yang mampu menangkap dan mewakili hubungan input-output yang kompleks, karena kemampuannya untuk memecahkan beberapa masalah relatif mudah digunakan, ketahanan untuk menginput data kecepatan untuk eksekusi dan menginisialisasikan system rumit[15].

Konsep JST menurut [16] bisa dilihat dari kerangka kerja JST yaitu dilihat dari jumlah lapisan layer dan jumlah node pada setiap lapisan. Lapisan-lapisan penyusun JST dapat dibagi menjadi tiga yaitu lapisan *input*, lapisan *hidden* dan lapisan *output*. Ada beberapa arsitektur JST di antaranya Jaringan *Single Layer* dan Jaringan *Multi-Layer*.

1) Jaringan Single Layer

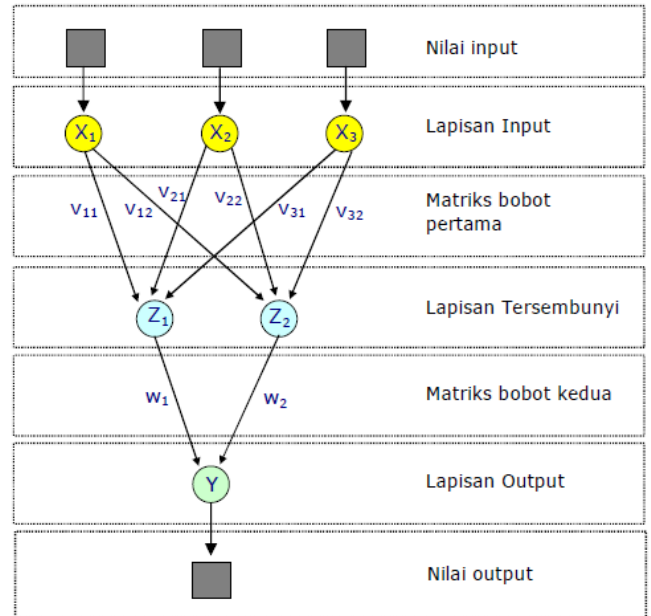
Jaringan yang memiliki arsitektur jenis inihanya memiliki satu bah lapisan bobot koneksi, jaringan ini terdiri dari unit-unit *input* yang menerima sinyal dari luar dan unit-unit *output* dimana kita bisa membaca respon dari JST tersebut. Adapun gambar untuk jaringan *single layer* dapat dilihat pada gambar 2.



Gambar 2. Jaringan Single Layer

2) Jaringan Multi-Layer

Merupakan jaringan dengan satu atau lebih lapisan tersembunyi. *Multi-layer* ini memiliki kemampuan lebih dalam memecahkan masalah bila dibandingkan dengan *single layer*. Namun jaringan *multi-layer* dalam pelatihannya jauh lebih rumit, adapun gambar arsitektur jaringan *multi-layer* dapat dilihat pada gambar 3.



Gambar 3. Jaringan Multi-Layer

C. Fungsi Aktivasi

Fungsi aktivasi merupakan bagian penting dalam tahapan perhitungan keluaran dari suatu algoritma. Beberapa fungsi aktivasi yang digunakan dalam JST adalah :

1. Fungsi identitas  $f(x)=x$ , untuk semua  $x$  (1)
2. Fungsi undak biner (dengan batas ambang)

$$f(x) = \begin{cases} 1 & \text{untuk } x \geq \theta \\ 0 & \text{untuk } x < \theta \end{cases} \dots\dots\dots (2)$$

3. Fungsi sigmoid

$$f(x) = \frac{1}{1 + \exp(-\sigma x)} \dots\dots\dots (3)$$

$$f'(x) = \sigma f(x)[1 - f(x)] \dots\dots\dots (4)$$

4. Fungsi sigmoid bipolar

$$g(x) = 2f(x) - 1 = \frac{2}{1 + \exp(-\sigma x)} \dots\dots\dots (5)$$

$$= \frac{1 - \exp(-\sigma x)}{1 + \exp(-\sigma x)} \dots\dots\dots (6)$$

$$g'(x) = \frac{\sigma}{2} [1 + g(x)][1 - g(x)] \dots\dots\dots (7)$$

D. Daftar Istilah dan Notasi Dalam Jaringan Syaraf Tiruan

Beberapa istilah-istilah jaringan syaraf tiruan yang sering ditemui dapat dilihat pada TABEL I [17].

TABEL I  
DAFTAR ISTILAH JST

Isitilah	Keterangan
Neuron, node, unit	Sel saraf tiruan yang merupakan elemen pengolahan jaringan syaraf tiruan. Setiap neuron menerima data input, memproses input tersebut (melakukan sejumlah perkalian dengan

Isitilah	Keterangan
	melibatkan fungsi aktivasi), dan mengirimkan hasilnya berupa sebuah <i>output</i> .
Jaringan	Kumpulan <i>neuron</i> yang saling terhubung dan membentuk lapisan.
Input atau masukan	Berhubungan dengan sebuah atribut tunggal dari sebuah pola atau data lain dari dunia luar. Sinyal-sinyal <i>input</i> ini kemudian diteruskan ke lapisan selanjutnya.
Lapisan tersembunyi ( <i>Hidden Layer</i> )	Lapisan yang tidak secara langsung berinteraksi dengan dunia luar. Lapisan ini memperluas kemampuan jaringan syaraf tiruan dalam menghadapi masalah-masalah kompleks.
Bobot	Bobot dalam jaringan syaraf tiruan merupakan nilai matematis dari koneksi, yang mentransfer jaringan dari satu ke lapisan lainnya. Bobot ini digunakan untuk mengatur jaringan sehingga jaringan syaraf tiruan bisa menghasilkan <i>output</i> yang diinginkan sekaligus bertujuan membuat jaringan tersebut belajar.
Fungsi aktivasi, fungsi transfer	Fungsi yang menggambarkan hubungan antara tingkat aktivasi yang berbentuk <i>linear</i> atau <i>nonlinear</i> . Beberapa fungsi aktivasi jaringan syaraf tiruan yaitu <i>Hard limit</i> , <i>Purelin</i> dan <i>Sigmoid</i> . Yang populer digunakan saat ini adalah fungsi <i>sigmoid</i> yang memiliki beberapa varian yakni <i>sigmoid</i> logaritma, <i>sigmoid biner</i> , <i>sigmoid bipolar</i> dan <i>sigmoid tangent</i> .
Aturan kerja	Secara umum aturan kerja dari teknik algoritma jaringan syaraf tiruan ada 4, yaitu aturan <i>error correcting</i> , aturan <i>Boltzman</i> , aturan <i>Hebbian</i> dan aturan kompetitif

Notasi yang sering dipakai dalam Jaringan Syaraf Tiruan menggunakan beberapa notasi yang berguna dalam pembuatan algoritma pembelajaran *backpropagation* antara lain dapat dilihat pada TABEL II berikut ini:

TABEL II  
DAFTAR NOTASI DALAM JST

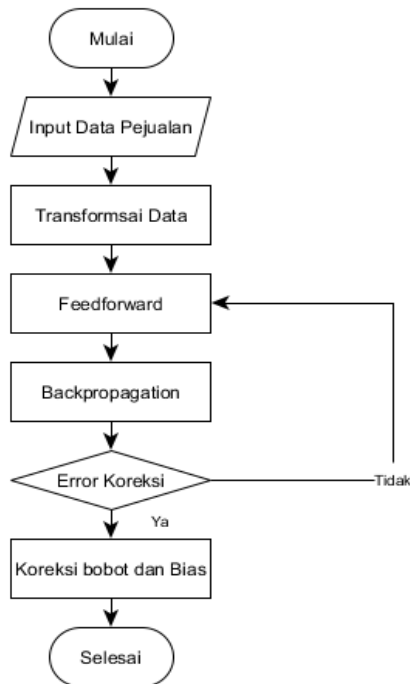
Notasi	Keterangan
$x_i, y_j$	Aktivitas dari unit $x_i, y_j$ untuk unit input $x_i, x_i =$ sinyal input, sedangkan unit $y_j, y_j = f(y_{in_j})$
$W_{ij}$	Bobot koneksi dari unit $x_i$ ke unit $y_i$
$b_j$	Biasa unit $y_j$

Notasi	Keterangan
$Y_{in_j}$	Input jaringan ke unit $y_j, y_{in_j} = b_j + \sum_i x_i w_{ij}$
$W$	Bobot matriks $W = \{w_{ij}\}$
$W_{ij}$	Vector bobot $w_{ij} = (w_{1j}, w_{2j}, \dots, w_{nj})^T$
$\ x\ $	Vector $X$
$\theta_j$	Threshold untuk aktivasi neuron $y_j$ , jika input yang masuk ke jaringan lebih besar dari nilai <i>threshold</i> , maka fungsi aktivasi = 1, sebaliknya fungsi aktivasi = 0
$S$	Pelatihan vector input $S = (S_1, S_2, \dots, S_n)$
$T$	Pelatihan vector output atau target $t = (t_1, t_2, \dots, t_n)$
$X$	Vector input $x = (x_1, x_2, \dots, x_n)$
$\Delta w_{ij}$	Perubahan dalam $w_{ij}$ " $\Delta w_{ij} = w_{ij}$ baru - $w_{ij}$ lama
$\alpha$	Tingkat pembelajaran ( <i>learning rate</i> ). <i>Learning rate</i> akan mengawasi atau mengontrol bobot dalam setiap kali pembelajaran

#### E. Algoritma Backpropagation

*Backpropagation* merupakan algoritma pembelajaran yang terawasi (*supervised learning*) dan biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyi [18]. *Backpropagation* termasuk ke dalam arsitektur jaringan *multi-layer*.

*Backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*feedforward*) harus dikerjakan terlebih dahulu [7]. Untuk memudahkan pemahaman lihat gambar 4.



Gambar 4. Flowchart alur *Backpropagation*

Dilihat pada gambar 4 secara garis besar proses peramalan dengan *JST Backpropagation* dilakukan dalam 3 langkah besar yaitu *Feedforward*, *backpropagation* dan update bobot juga bias. Dalam perhitungannya secara manual menggunakan beberapa rumus untuk proses pelatihan. Seperti yang telah dijelaskan sebelumnya, Dalam tahapan pelatihan *backpropagation* ini meliputi 3 fase yaitu :

- 1) Fase pertama *Feedforward*  
Untuk masing-masing pasangan *training* lakukan proses *feedforward* dengan 3 langkah.  
**Langkah 1:** Masing-masing unit *input* ( $X_i, i=1, \dots, n$ ) menerima sinyal *input*  $X_i$  dan menyebarkan ke sinyal unit lapisan atas (*hidden unit*).  
**Langkah 2 :** Masing-masing unit tersembunyi ( $Z_j, j=1, \dots, p$ ) menggunakan persamaan (8) menjumlahkan bobot sinyal *input*  $z_{in_j} = v_{0j} + \sum_i x_i v_{ij}$  .....(8) dan mengaplikasikan fungsi aktifasi untuk menghitung sinyal *output* menggunakan persamaan (9)  $z_j = f(z_{in_j})$ .....(9) dan mengirimkan sinyal tersebut ke semua unit  $p$  ada lapisan atasnya (unit *output*).  
**Langkah 3:** Masing-masing unit *output* ( $Y_k, k=1, \dots, m$ ) menggunakan persamaan (10) menjumlahkan jumlah sinyal *input*  $Y_{in_k} = W_{0k} + \sum_i z_i w_{ji}$  .....(10) dan mengaplikasikan fungsi aktifasi persamaan (11) untuk menghitung sinyal *output*  $Y_k = f(y_{in_k})$  .....(11)
- 2) Fase kedua *Backpropagation*

Untuk masing-masing pasangan *training*, kemudian lakukan *backpropagation*.

**Langkah 1 :** Masing-masing unit *output* ( $Y_k, k=1, \dots, m$ ) menerima sebuah pola target yang bersesuaian dengan pola *training input*, menghitung informasi kesalahan menggunakan persamaan  $\delta_k = (t_k - y_k) f'(y_{in_k})$  .....(12) kemudian menghitung koreksi bobot (ini digunakan untuk memperbaiki  $w_{jk}$ ) dengan persamaan  $\Delta w_{jk} = \alpha \delta_k \cdot z_j$ ....(13) dan akhirnya menghitung korelasi bias (ini digunakan untuk memperbaiki  $w_{ok}$ ) dengan persamaan  $\Delta w_{0k} = \alpha \delta_k$  .....(14) setelah itu mengirimkan  $\delta_k$  ke unit dalam lapisan paling atas.  
**Langkah 2 :** Masing-masing unit yang tersembunyi ( $Z_j, j=1, \dots, p$ ) menjumlahkan input delta (dari unit lapisan atas) dengan menggunakan persamaan

$$\delta_{inj} = (x + \alpha)^n = \sum_{k=1}^m \delta_k w_{jk} \dots\dots\dots (15)$$

Kalikan nilai ini dengan turunan fungsi aktivasi untuk menghitung informasi kesalahan menggunakan persamaan  $\delta_j = (\delta_j) f'(Z_{inj})$  .....(16) kemudian hitunglah koreksi bobot (digunakan untuk memperbaiki  $v_{ij}$ ), dengan menggunakan persamaan yaitu  $\Delta v_{ij} = \alpha \delta_j x$  .....(17) Setelah itu hitung koreksi bias (digunakan untuk memperbaiki  $v_{oj}$ ) gunakan persamaan

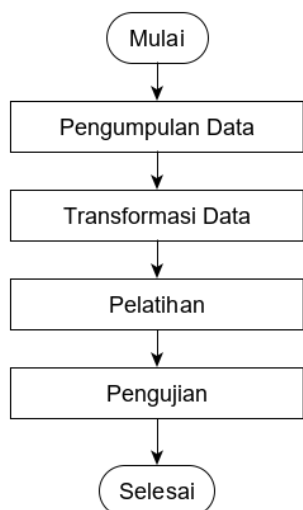
$$\Delta v_{0j} = \alpha \delta_j \dots\dots\dots (18)$$

3) Fase tiga *update* bobot dan bias  
Setelah itu masuk pada fase memperbaiki bobot dan bias atau memperbarui nilai bobot dan bias. Dengan 2 langkah :

- Langkah 1:** Masing-masing unit *output* ( $Y_k, k=1, \dots, m$ ) memperbaiki bobot dan bias ( $Z_j, j=1, \dots, p$ ) dengan menggunakan persamaan  $W_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}$  .....(19) kemudian masing-masing unit tersembunyi ( $Z_j, j=1, \dots, p$ ) memperbaiki bobot dari bias ( $X_i, i=1, \dots, n$ ) menggunakan persamaan  $v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}$  .....(20)
- Langkah 2:** Tes kondisi berhenti

F. Tahap Membentuk *Backpropagation* Menggunakan *MATLAB*

Tahapan yang harus dilakukan untuk implementasi program *Backpropagation* dengan Matlab adalah dengan 4 tahapan seperti pada gambar 5 yaitu pengumpulan data, transformasi data (*pre-processing*), pelatihan dan pengujian.



Gambar 5. Tahap implementasi JST dengan Matlab

1) Pengumpulan Data

Dalam tahap pengumpulan data ini dilakukan identifikasi masalah, kemudian melakukan studi literatur yaitu mencari studi atau penelitian-penelitian dan referensi buku terkait peramalan khususnya peramalan menggunakan metode JST *Backpropagation* untuk mengetahui lebih lanjut kebutuhan data dalam tahap penelitian selanjutnya. Setelah itu dilakukan pengumpulan data pada perusahaan yang kemudian didapatkan data yang akan digunakan dalam penelitian ini yaitu data penjualan produk AMDK 600ml tahun 2017-2018. Data penjualan dapat dilihat pada gambar 6.

	TANGGAL	INVOICE	PRODUCCCT	VARIANT	ORDER TYPE	SKU	SOLD QTY	CURRENCI
1	5-Jan-17	9050001	Robong Holo 600ml	Dus	Distributor		200	IDR
2	5-Jan-17	9050001	Robong Holo 330ml	Dus	Distributor		100	IDR
3	5-Jan-17	9050001	Robong Holo 240ml	Dus	Distributor		350	IDR
4	5-Jan-17	9050002	Robong Holo 600ml	Dus	Distributor	030101	200	IDR
5	5-Jan-17	9050002	Robong Holo 240ml	Dus	Distributor	030106	250	IDR
6	5-Jan-17	9050003	Robong Holo 600ml	Dus	Distributor	030108	111	IDR
7	6-Jan-17	9050004	Robong Holo 240ml	Dus	Distributor	030113	90	IDR
8	6-Jan-17	9050005	Robong Holo 240ml	Dus	Distributor		189	IDR
9	8-Jan-17	9050006	Robong Holo 600ml	Dus	Distributor	040101	200	IDR
10	8-Jan-17	9050007	Robong Holo 330ml	Dus	Distributor	040102	890	IDR
11	9-Jan-17	9050008	Robong Holo 330ml	Dus	Distributor	040103	175	IDR
12	10-Jan-17	9050009	Robong Holo 240ml	Dus	Distributor	040104	200	IDR
13	11-Jan-17	9050010	Robong Holo 600ml	Dus	Distributor	040105	135	IDR
14	11-Jan-17	9050011	Robong Holo 600ml	Dus	Distributor	040106	200	IDR
15	11-Jan-17	9050012	Robong Holo 600ml	Dus	Distributor	040108	250	IDR

Gambar 6. Data Penjualan Produk

Dari data penjualan harian pada gambar 6. Kemudian di persiapkan untuk proses normalisasi pada tahap transformasi data.

2) Transformasi Data (Pre-processing)

Transformasi data dilakukan dengan bantuan software MATLAB R2018a. Data penjualan yang telah didapatkan pada tahapan sebelumnya diolah menjadi data set. Pengolahan data penjualan menjadi data set ini sesuai dengan tahapan yang ada pada JST *Backpropagation*. Sebelum melakukan proses peramalan, maka data yang

sudah ada akan di bagi menjadi dua bagian, yaitu data *training* dan data *testing*. Data *training* merupakan data yang akan digunakan dalam proses pelatihan yaitu data penjualan tahu 2017 bulan Januari-Desember. Adapun data *testing* adalah data yang digunakan dalam proses peramalan yaitu data penjualan bulan Januari-Desember 2018.

Dalam tahap transformasi data akan dilakukan 2 aktivitas yaitu normalisasi dan perancangan arsitektur jaringan.

a) Normalisasi

Data penjualan yang telah dikumpulkan terlebih dahulu ditransformasikan ke dalam data antara 0 – 1 sebelum dilakukan pelatihan dan pengujian dengan JST *Backpropagation*. Normalisasi bertujuan agar jaringan tidak mengalami kegagalan ketika melakukan pelatihan dan pengujian. Data dinormalisasikan dengan persamaan rumus (21) dalam interval [0-1]. Rumus transformasi data actual untuk di normalisasikan menggunakan rumus:

$$X' = \frac{0,8(X-a)}{(b-a)} + 0,1 \dots\dots\dots (21)$$

Dimana:

- X' = Data Normalisasi
- X = Data Asli
- a = Nilai data asli minimum
- b = Nilai data asli maximum

Sehingga data normalisasi didapatkan seperti pada TABEL III.

TABEL III  
NORMALISASI DATA PELATIHAN

Bulan/Produk	Produk AMDK 600ml
Januari	0.1209
Februari	0.1000
Maret	0.1203
April	0.1405
Mei	0.1379
Juni	0.1439
Juli	0.1415
Agustus	0.2216
September	0.4282
Oktober	0.3286
November	0.7904
Desember	0.9000

Hasil normalisasi dalam excel akan dipanggil dan di baca dengan matlab untuk selanjutnya di lakukan pembuatan arsitektur JST menggunakan perintah:

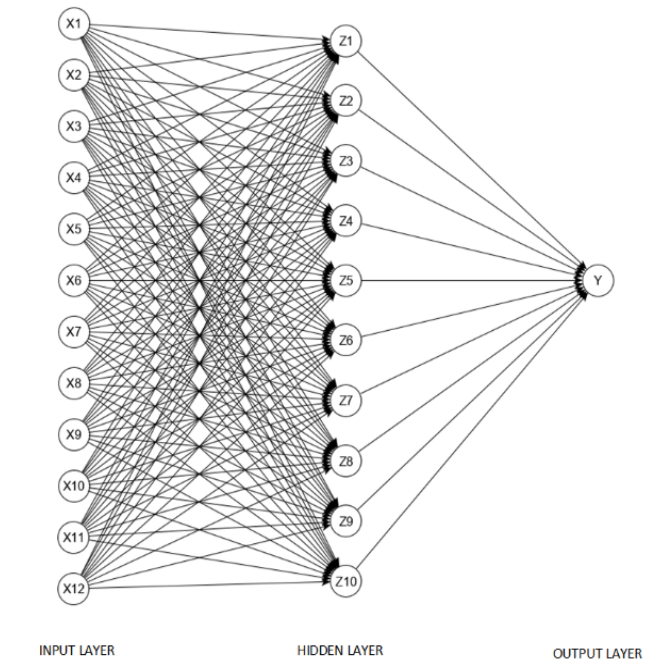
```

Filename = 'Book1.xlsx';
Sheet = 2;
xlRange = 'D6:P17';
Data = xlsread(filename, sheet, xlRange);
data_latih = Data(:,1:12)';
  
```

```
target_latih = Data(:,13)';
[m,n] = size(data_latih);
```

b) Perancangan Arsitektur Jaringan

Tahap *pre-processing* selanjutnya adalah perancangan arsitektur jaringan dan menentukan parameter. Jaringan yang digunakan untuk meramal penjualan AMDK memiliki 3 layers yaitu 12 *input layer*, 11 *hidden layer* dan 1 *output layer*. Jumlah *hidden layers* dan parameter ditentukan dengan cara *trial and error*. Berdasarkan jumlah neuron *input*, *hidden* dan *output* yang telah ditentukan maka gambar arsitektur *JST Backpropagation* yang dibentuk akan terlihat seperti pada gambar 7.



Gambar 7. Arsitektur *JST Backpropagation*

Membuat arsitektur *JST* dan memberikan nilai pada setiap parameter ini sangat berpengaruh selanjutnya dalam proses pelatihan.

3) Pelatihan

Menggunakan data pelatihan yaitu data penjualan bulan Januari-Desember tahun 2017 dengan menggunakan model arsitektur yang sudah terbentuk, kemudian dilakukan inialisasi bobot dan parameter dengan perintah `newff`, kemudian perintah `train` untuk melatih dan melihat bobot akhirnya. Dengan memberikan nilai yang kita inginkan pada parameter-parameter kita dapat memperoleh hasil yang optimal. Dalam MATLAB perintah `net.trainParam.show` dipakai untuk menampilkan frekuensi perubahan mse (defaultnya setiap 25 epoch). Perintah `net.trainParam.epochs` untuk menentukan jumlah epoch atau terasi maksimum pelatihan. Perintah `net.trainParam.goal` untuk menentukan batas nilai mse agar iterasi dihentikan. Iterasi berhenti jika mse < batas

yang ditentukan dalam `net.trainParam.goal` atau jumlah iterasi mencapai batas yang ditentukan dalam `net.trainParam.epochs`. Perintah `net.trainParam.lr` untuk menentukan laju pemahaman (learning rate).

Adapun model arsitektur dan parameter yang digunakan untuk mendapatkan arsitektur terbaik dalam *JST* dapat dilihat pada TABEL IV.

TABEL IV  
MODEL DAN PARAMETER ARSITEKTUR *JST BACKPROPAGATION*

Arsitektur	Parameter				
	lr	Mc	Epoch	Goal	Fungsi Aktivasi
12-10-1	0.1	0.9	1000	0.001	Logsig

4) Pengujian

Pada tahap pengujian data yang menjadi data *training* dimasukkan ke dalam Matlab. Dalam penelitian ini yaitu data penjualan bulan januari- hingga desember tahun 2018. Setelah mendapatkan hasil *JST* dilakukan pengujian menggunakan *MAPE*. *MAPE* atau *Mean Absolute Percentage* adalah dengan persamaan rumus[19]:

$$MAPE = \frac{\sum \left| \frac{x_t - \hat{x}_t}{x_t} \right|}{n} \times 100\% \dots\dots\dots (22)$$

Setelah mendapatkan hasil simulasi selanjutnya untuk mendapatkan hasil peramalan dilakukan denormalisasi dengan persamaan rumus [20]:

$$x = \frac{(x' - 0,1)(b - a)}{0,8} + a \dots\dots\dots (23)$$

III. HASIL DAN PEMBAHASAN

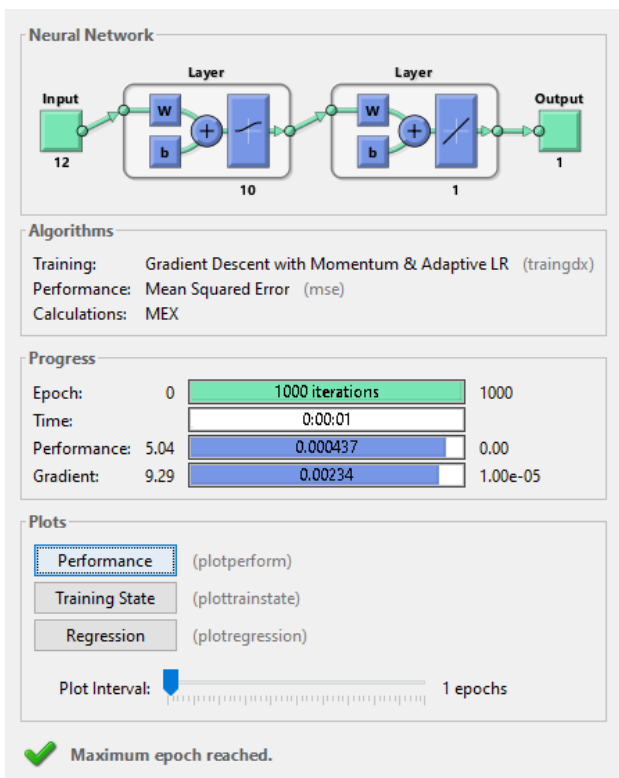
Hasil dari penelitian ini adalah sebuah model arsitektur dengan tingkat akurasi yang terbaik dan hasil peramalan penjualan AMDK.

A. Hasil Pelatihan *JST Backpropagation*

Pelatihan data dilakukan beberapa kali *trial and error* untuk mendapatkan jaringan terbaik dengan menentukan jumlah neuron. Perintah untuk inialisasi bobot dan parameter pada MATLAB menggunakan instruksi :

```
net=newff(minmax(data_latih), [10,1], {'logsig', 'purelin'}, 'traingdx');
net.performFcn='mse';net.trainParam.goal =0,001;
net.trainParam.show=25;
net.trainParam.epochs=1000;
net.trainParam.mc= 0.9;
net.trainParam.lr = 0.1;
```

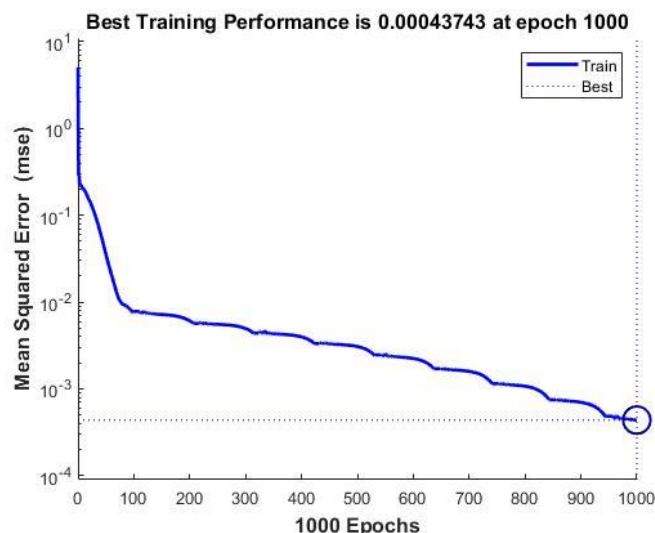
Hasilnya dapat dilihat pada gambar 8 yang merupakan hasil dari arsitektur terbaik.



Gambar 8. Model Arsitektur JST Pada Matlab

Pada pengolahan data yang telah didapati berdasarkan pelatihan terbaik berikut beberapa parameter pelatihan yang diatur agar hasil pelatihan optimal.

- *Learning rate* (*lr*) bertujuan untuk mempercepat laju iterasinya. Hasil pembahasan untuk *lr* ini dimana jika nilai *lr* terlalu besar maka algoritma menjadi kurang stabil dan mencapai titik minimum local. Jadi *error*nya adalah nol (0).
- *Momentum constant* (*mc*) bertujuan untuk menurunkan gradient dengan momentum dengan memberikan nilai 0-1. Untuk mendapatkan *error* terkecil *lr* dan *mc* harus terkoordinasi dengan baik. Tidak ada ketentuan parameter yang pasti untuk mendapatkan *error* yang kecil. Jadi harus dilakukan beberapa kali percobaan. Dalam penelitian ini menggunakan *lr* 0,1 dan *mc* 0,9.
- *Epoch* (iterasi) bertujuan untuk menunjukkan jumlah iterasi maksimum pada pelatihan. Tidak ada ketentuan berapa nilai iterasi yang harus digunakan. Namun jika terlalu banyak nilai iterasi yang digunakan akan memerlukan waktu yang lama, sebaliknya jika iterasi terlalu sedikit hasil pelatihan akan kurang baik. Dalam penelitian ini menggunakan 1000 tercapai dan mendapatkan hasil yang baik.
- *Goal* untuk menentukan batas nilai MSE agar iterasi berhenti. Dalam penelitian ini menggunakan *goal* sebesar 0,001. Jika *goal* tercapai maka pada grafik *performance* MSE harus kurang atau sama dengan 0,001. Pada penelitian ini mendapatkan MSE 0,00043743, dapat dilihat pada gambar 9.



Gambar 9. Performance Model Arsitektur 12-10-1

- Fungsi Aktivasi bertujuan untuk menghubungkan *hidden layer* ke *output layer*. Dalam penelitian ini menggunakan fungsi aktivasi fungsi sigmoid biner (*logsig*) yang umum digunakan dengan range antara 0-1.

#### B. Hasil Pengujian JST Backpropagation

Hasil pelatihan terbaik diuji dengan data uji yang telah disiapkan menggunakan instruksi:

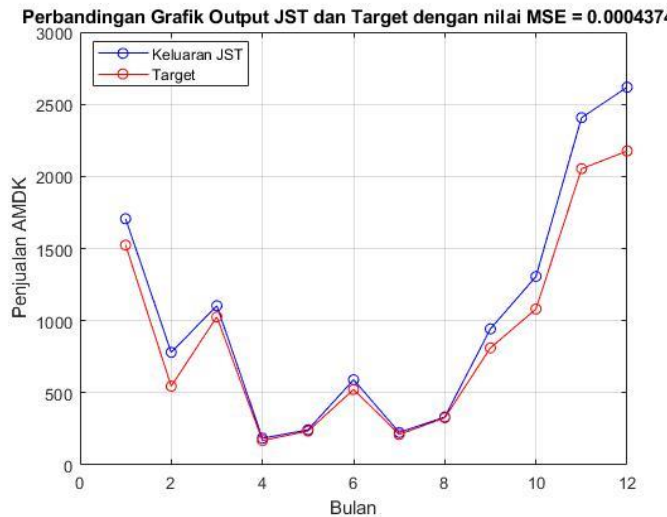
```
filename='Book1.xlsx';
sheet=2;
xlRange='D24:P35';
Data=xlsread(filename, sheet, xlRange);
data_uji=Data(:,1:12)';
target_uji=Data(:,13)';
[m,n] = size(data_uji);
```

Kemudian pencarian hasil uji JST dengan intruksi :

```
hasil_uji= sim(net_keluaran,data_uji);
nilai_error=hasil_uji-target_uji;
max_data=2590;
min_data = 169;
hasil_uji=((hasil_uji-0.1)*(max_data-
min_data)/0.8)+min_data;
```

Hasil pengujian seperti pada gambar 10.





Gambar 10. Perbandingan Output JST dengan JST Pelatihan Data

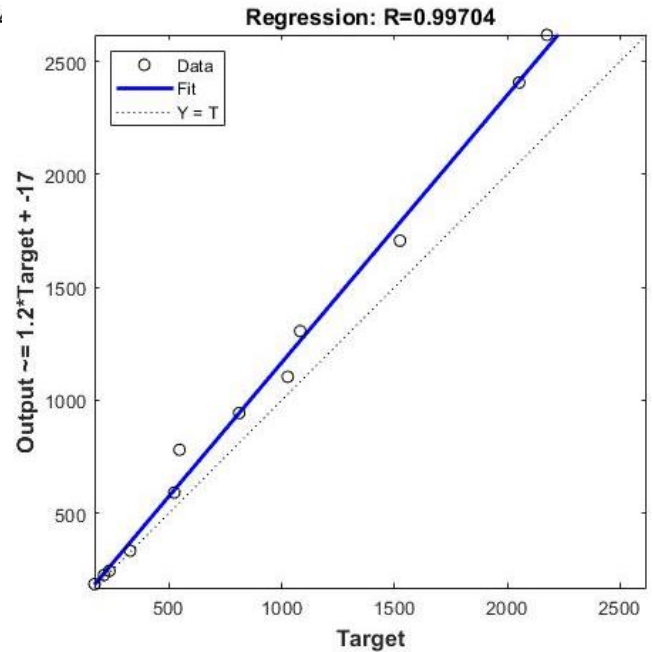
Hasil pengujian terhadap data latih yaitu dengan menggunakan *MSE* untuk membandingkan data aktual dengan hasil peramalan JST. Pada Gambar 10 terlihat model hasil JST tidak berbeda jauh dengan data aktual (target)nya. Maka rancangan JST ini dapat dijadikan model untuk tahap ke pengujian data.

Berikut adalah tabel perbandingan output JST dengan pelatihan data (lihat TABEL V).

TABEL V  
HASIL PERAMALAN AMDK TAHUN 2019

No.	Normalisasi		Denormalisasi		%
	Target	Z	Target	Z	
1	0,7721	0,7941	-0,0220	-0,0285	-2,85
2	0,7342	0,7102	0,0240	0,0327	3,27
3	0,4736	0,4908	-0,0172	-0,0363	-3,63
4	0,3502	0,3443	0,0059	0,0168	1,68
5	0,2007	0,1930	0,0077	0,0383	3,83
6	0,1649	0,1472	0,0177	0,1071	10,71
7	0,2615	0,2852	-0,0237	-0,0904	-9,04
8	0,1726	0,1624	0,0102	0,0590	5,90
9	0,1453	0,1613	-0,0160	-0,1101	-11,01
10	0,4170	0,3960	0,0210	0,0504	5,04
11	0,2686	0,2937	-0,0251	-0,0932	-9,32
12	0,5707	0,5005	0,0702	0,1231	12,31
MAPE					6,88

Pengujian data menggunakan *MAPE* untuk mengukur validasi jaringan pada keseluruhan data didapat sebesar 6,88%. maka tingkat akurasi termasuk sangat baik karena nilainya dibawah 10%.



Gambar 11. Hasil Regrsai Data Uji

Pada gambar 11 menunjukkan regresi sebesar 0,9847 yang berarti antara variabel-variabel aktual dengan JST pada pengujian mempunyai korelasi yang baik. Ukuran korelasi sebesar 0,9974 menunjukkan adanya derajat asosiasi yang tinggi. Hasil regresi pada Gambar 11 juga menunjukkan verifikasi data antara data aktual dengan JST ada perbedaan atau selisih namun tidak begitu signifikan.

### C. Hasil Peramalan Penjualan AMDK

Setelah didapatkan model arsitektur yang terbaik dari proses pelatihan dan pengujian, model tersebut dapat digunakan untuk melakukan peramalan jumlah penjualan AMDK. Hasil peramalan JST didenormalisasikan untuk mengetahui nilai aslinya menggunakan rumus (16). Sehingga peramalan penjualan AMDK tahun 2019 adalah pada TABEL VI. Data aktual pada bulan Januari hingga Juni ditampilkan guna melihat dan mengkonfirmasi ketepatan hasil peramalan.

TABEL VI  
HASIL PERAMALAN AMDK TAHUN 2019

Periode 2019	Data Aktual	Hasil JST	Hasil Peramalan (pcs)
Januari	2267	0,7941	2271
Februari	2008	0,7102	2019
Maret	1346	0,4908	1358
April	915	0,3443	917
Mei	462	0,1930	462
Juni	324	0,1472	324
Juli	-	0,2852	739

Periode 2019	Data Aktual	Hasil JST	Hasil Peramalan (pcs)
Agustus	-	0,1624	370
September	-	0,1613	367
Oktober	-	0,3960	1073
November	-	0,2937	765
Desember	-	0,5005	1388

Dalam perhitungan hasil peramalan dengan data aktual terdapat beberapa selisih, hal ini normal karena dalam analisis kegiatan peramalan hasil peramalan tidak selamanya sempurna dengan bilangan bulat/real sesuai data aktual dimana hasil peramalan bersifat pendekatan artinya dapat berbeda dengan data aktual. Perhitungan persentasi selisih kesalahan dapat diketahui dengan persamaan rumus persentasi kesalahan (24):

$$Et = \frac{\text{nilai peramalan} - \text{nilai aktual}}{\text{nilai aktual}} \times 100\% \dots\dots\dots (24)$$

Pada penelitian ini ada perbedaan atau selisih namun tidak begitu signifikan yaitu 2% dari semua jumlah data aktual yang ada.

#### IV. KESIMPULAN

Berdasarkan pelatihan JST *backpropagation* model arsitektur JST yang digunakan yaitu jaringan *multi-layer* 12 *input layer*, 10 *hidden layer* dan 1 *output layer* dengan nilai *MSE* sebesar 0,00043743 dan nilai *MAPE* 6,88%. Perhitungan peramalan penjualan AMDK dengan menggunakan JST *Backpropagation* diperoleh hasil peramalan penjualan AMDK 600ml untuk periode 2019 dengan data aktual memiliki selisih. Hasil peramalan penjualan produk AMDK dengan menggunakan metode JST pada penelitian ini terdapat selisih 2% dengan data aktual. Hal ini normal karena dalam kegiatan peramalan hasil peramalan bersifat pendekatan artinya dapat berbeda dengan data aktual namun dengan selisih yang tidak banyak.

#### DAFTAR PUSTAKA

[1] D. Puspitasari, M. Mentari, and W. R. Faldiansyah, "Penerapan Metode *Single Exponential Smoothing* Menggunakan Pendekatan Adaptif Pada Peramalan Jumlah Pelanggan Dan Kebutuhan Air Pada Pdam Kota," *Jurnalti Polinema*, pp. 1–6, 2017.

[2] [2] A. Rohman, "Peramalan Penjualan Air Minum Dalam Kemasan (AMDK) PT. Tirta Investama Subang, Jawa Barat," Institut Pertanian Bogor, 2006.

[3] [3] Y. Y. Silfiani and H. Utomo, "Pengaruh Citra Merek, Kualitas Produk Dan Strategi Promosi Terhadap Keputusan Pembelian Produk Amdk Java (Studi Kasus Pada Konsumen Mahasiswa STIE AMA, IAIN Dan UKSW Salatiga) (Yen Yen Silfiani, Hardi Utomo)," *Jurnal Among Makarti*, vol. 10, no. 20, pp. 37–54, 2017.

[4] [4] A. E. Armi, A. H. Kridalaksana, and Z. Arifin, "Peramalan Angka Inflasi Kota Samarinda Menggunakan Metode *Double Exponential Smoothing* (Studi Kasus : Badan Pusat Statistik Kota Samarinda)," vol. 14, no. 1, 2019.

[5] [5] M. Nafarin, *Penganggaran Perusahaan*, Jakarta: Salemba Empat, 2007.

[6] [6] J. Heizer and B. Render, *Operation Management*, New Jersey: Prentice Hall, 2005.

[7] [7] Marjiyono, B. WA Soedijono, and E. T. Luthfi, "Penggunaan Jaringan Syaraf Tiruan Untuk Meramalkan Permintaan Pada Perusahaan retail," *Seminar Nasional Teknologi Informasi dan Multimedia 2018*, paper 2.12-7, p.2.

[8] [8] F. Pakaja, Agus Naba, and Purwanto, "Peramalan Penjualan Mobil Menggunakan Jaringan Syaraf Tiruan dan *Certainty Factor*," *Jurnal EECCIS*, vol. 6, no. 1, pp. 23–28, 2012.

[9] [9] H. C. Aprilianto, S. Kumalaningsih, and I. Santoso, "Penerapan Jaringan Syaraf Tiruan Untuk Peramalan Penjualan Dalam Mendukung Pengembangan Agroindustri Coklat di Kabupaten Blitar," *Jurnla Habitat*, vol. 29, pp. 129–137, 2018.

[10] [10] A. Revi, Solikhun, and M. Safii, "Jaringan Syaraf Tiruan Dalam Memprediksi Jumlah Produksi Daging Sapi Berdasarkan Provinsi," vol. 2, pp. 297–304, 2018.

[11] [11] N. F. Hasan, K. Kusriani, and H. Al Fatta, "Analisis Arsitektur Jaringan Syaraf Tiruan Untuk Peramalan Penjualan Air Minum Dalam Kemasan," *JURTI*, vol. 3, no. 1, pp. 1–10, 2019.

[12] [12] F. A. Muqtadiroh, A. R. Syofiani, and T. S. Ramadhani, "Analisis Peramalan Penjualan Semen Non-curah (ZAK) PT. Semen Indonesia (PERSERO) Tbk Pada Area Jawa Timur," *Seminar Nasional Teknologi Informasi dan Komunikasi*, pp. 308–310, 2015.

[13] [13] A. Kristianto, *Jaringan Syaraf Tiruan (Konsep Dasar, Algoritma dan Aplikasi)*. Yogyakarta: Gava Media, 2004.

[14] [14] A. Hermawan, *Jaringan Saraf Tiruan Teori dan Aplikasi*, Yogyakarta: Penerbit Andi, 2006.

[15] [15] A. Sudarsono, "Jaringan Syaraf Tiruan Untuk Memprediksi Laju Pertumbuhan Penduduk Menggunakan Metode *Backpropagation* (Studi Kasus di Kota Bengkulu)," *Jurnal Media Infotama*, vol. 12, no. 1, pp. 61–69, 2016.

[16] [16] D. Puspitaningrum, *Pengantar Jaringan Saraf Tiruan*, Yogyakarta: Penerbit Andi, 2006.

[17] [17] Andayani, "Memprediksi Kualitas Lulusan Mahasiswa Menggunakan Algoritma Jaringan Syaraf Tiruan," M. Kom. thesis, Universitas AMIKOM Yogyakarta, 2018.

[18] [18] M. Febrina, F. Arina, and R. Ekawati, "Peramalan Jumlah Permintaan Produksi Menggunakan Metode Jaringan Syaraf Tiruan (JST) *Backpropagation*," vol. 1, no. 2, pp. 174–179, 2013.

[19] [19] S. Tias Safitri, Nurkaromah Dwidayati, "Perbandingan Peramalan Menggunakan Metode *Exponential Smoothing Holt-Winters* dan ARIMA," vol. 6, no. 1, pp. 48–58, 2017.

[20] [20] J. J. Siang, *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*, Yogyakarta: Penerbit Andi, 2005.