

## Penerapan Metode *Elman Recurrent Neural Network (ERNN)* Untuk Peramalan Penjualan

**Eka Pandu Cynthia<sup>1</sup>, Novi Yanti<sup>2</sup>, Yusra<sup>3</sup>, Yelvi Fitriani<sup>4</sup>, Muhammad Yusuf<sup>5</sup>**  
<sup>1,2,3,4,5</sup>Program Studi Teknik Informatika Fakultas Sains dan Teknologi, Universitas Islam  
 Negeri Sultan Syarif Kasim Riau  
<sup>1,2,3,4,5</sup>Jl. HR. Soebrantas KM 15 Simpang Baru, Panam – Pekanbaru, Riau  
 e-mail: <sup>1</sup>[eka.pandu.cynthia@uin-suska.ac.id](mailto:eka.pandu.cynthia@uin-suska.ac.id), <sup>2</sup>[novi\\_yanti@uin.suska.ac.id](mailto:novi_yanti@uin.suska.ac.id), <sup>3</sup>[yusra@uin-suska.ac.id](mailto:yusra@uin-suska.ac.id), <sup>4</sup>[yelvi.fitriani@uin-suska.ac.id](mailto:yelvi.fitriani@uin-suska.ac.id), <sup>5</sup>[muhhammad.yusuf8@students.uin-suska.ac.id](mailto:muhhammad.yusuf8@students.uin-suska.ac.id)

### Abstrak

*Penjualan adalah suatu aktivitas atau bisnis menjual suatu produk atau jasa. Pada penelitian ini, mengambil studi kasus pada sebuah perusahaan penjualan tempe (PT. HB) di Kota Pekanbaru yang menggunakan metode penjualan melalui agen atau sales. Permasalahan penjualan pada perusahaan ini adalah sering terjadinya returned product karena tidak laku, yang dapat menyebabkan kerugian yang cukup besar. Menggunakan Algoritma Elman Recurrent Neural Network (ERNN), penelitian ini melakukan prediksi penjualan tempe pada PT. HB. Data yang digunakan adalah data penjualan tempe harian periode Juli 2016 hingga September 2018 dengan parameter Jumlah Produksi, Harga, Jumlah Agen dan Jumlah Penjualan. Hasil yang diperoleh melalui percobaan beberapa skenario pelatihan dan pengujian implementasi algoritma pada kasus ini adalah akurasi tertinggi bernilai 96,92% pada arsitektur jaringan 3 input neuron layer, 3 neuron hidden layer, 1 output, pembagian data latih dan uji 70 : 30, nilai learning rate 0,9 dan maksimum epoch 900.*

**Kata kunci:** ERNN, Jaringan Syaraf Tiruan, Penjualan, Prediksi.

### Abstract

*Sales is an activity or business of selling a product or service. In this research, taking a case study at a Tempe sales company (PT. HB) in Pekanbaru City using the sales method by agents. The problem with sales at this company is the frequent occurrence of returned products because they do not sell, which can cause substantial losses. Using the Elman Recurrent Neural Network (ERNN) algorithm, this research predicts Tempe's sales at PT. HB. The data used are daily Tempe's sales data from July 2016 to September 2018 with parameters of Production Amount, Price, Amount of Agents and Amount of Sales. The results obtained through the experiment of several training scenarios and testing the algorithm implementation in this case are the highest accuracy of 96.92% in the network architecture 3 input neuron layer, 3 hidden layer hidden, 1 output, training data sharing and 70: 30 test, the value learning rate 0.9 and maximum epoch 900.*

**Keywords:** Artificial Neural Network, ERNN, Prediction, Sales.

### 1. Pendahuluan

Prediksi penjualan barang merupakan salah satu cara untuk menjaga stabilitas penjualan barang. Hasil prediksi yang diperoleh dapat dijadikan sebagai pertimbangan untuk mengambil keputusan dalam perencanaan manajemen bisnis. PT. HB sebagai salah satu perusahaan penjualan tempe yang cukup besar di Pekanbaru tentunya membutuhkan analisa-analisa dari hasil penjualan produksinya terutama dalam menjaga stabilitas produksi dan menghindari

kerugian. Hal ini disebabkan produk tempe merupakan produk yang memiliki durasi ketahanan kualitas yang cukup singkat, sehingga jika produk tempe tersebut tidak segera habis terjual maka akan terjadi pembusukan yang mengakibatkan kerugian bagi perusahaan dikarenakan *returned product* oleh agen penjualan.

Seiring berkembangnya teknologi informasi, salah satu cara dalam melakukan prediksi penjualan produk adalah menggunakan Jaringan Syaraf Tiruan. Merupakan sebuah metode kecerdasan buatan, konsep dari jaringan syaraf tiruan ini adalah bagaimana struktur dan kinerja otak manusia yang kemudian diimplementasikan dalam program komputer. Jaringan syaraf tiruan menerima informasi masukan lalu melakukan proses pembelajaran melalui bobot yang diterimanya. Jaringan Syaraf Tiruan juga mampu melakukan pengenalan terhadap basis data masa lalu dengan pengalaman. Data masa lalu yang dipelajari oleh Jaringan Syaraf Tiruan akan memiliki kemampuan dalam memberikan keputusan terhadap data yang belum pernah dipelajari. Selama proses pembelajaran tersebut pola masukan disajikan bersama-sama dengan pola keluaran yang diinginkan. Jaringan akan menyesuaikan nilai bobot sebagai tanggapan atas pola masukan dan saran yang disajikan tersebut (Hermawan, 2006).

*Elman Recurrent Neural Network* (ERNN) merupakan salah satu pengembangan dari algoritma jaringan syaraf tiruan *Backpropagation*. Perbedaan antara kedua algoritma ini yaitu pada ERNN mempunyai *feedback* di *hidden*. Dari hasil *feedback* tersebut menghasilkan tambahan *layer* yang disebut *context layer*. dengan adanya *context layer* dapat membuat iterasi dan kecepatan *update* parameter lebih baik. Hal ini memungkinkan untuk melakukan perhitungan berdasarkan nilai dari perhitungan sebelumnya, sehingga membuat jaringan syaraf elman lebih sesuai dalam melakukan peramalan atau peramalan (Talahatu, dkk 2015).

Penelitian terkait algoritma ERNN yang telah dilakukan sebelumnya antara lain “Penerapan *Elman Recurrent Neural Network* Pada Peramalan Konsumsi Listrik Jangka Pendek di PT. PLN APP Malang” (Sani, 2014) dan “Penggunaan *Elman Recurrent Neural Network* dalam Peramalan Suhu Udara sebagai Faktor yang Mempengaruhi Kebakaran Hutan” (Maulida, 2011). Pada penelitian-penelitian tersebut memaparkan kehandalan algoritma ERNN dalam melakukan prediksi atau peramalan dengan menghasilkan nilai *error* pada RMSE dan MAPE yang sangat kecil dan persentasi akurasi yang tinggi.

Berdasarkan pemaparan di atas, penelitian ini akan melakukan peramalan penjualan dengan studi kasus produk tempe pada PT. HB menggunakan algoritma ERNN dengan variabel masukan pelatihan : Jumlah Produksi (Pcs), Harga (Rupiah), Jumlah Agen Penjualan (Orang) dan Jumlah Produk Terjual (Pcs). Jumlah data yang digunakan sebanyak 180 data harian hasil penjualan tempe PT. HB dimulai pada bulan Juli 2017 hingga September 2017. Keluaran dari proses ini adalah peramalan jumlah penjualan tempe untuk satu hari kedepannya.

## 2. Metode Penelitian

Metode Penelitian memberikan penjelasan tentang langkah-langkah, data, lokasi penelitian, metode evaluasi yang digunakan serta penjelasan terstruktur tentang algoritma atau metode dari penelitian yang dibahas.



Gambar 1. Metodologi Penelitian

### 2.1. Studi Kepustakaan

Tahap studi pustaka merupakan tahapan mendapatkan dan mengumpulkan data dan informasi yang dibutuhkan dalam proses penelitian. Tahapan ini dilakukan untuk memperoleh referensi-referensi yang berkaitan dengan penelitian yang dilakukan, berupa jurnal, *ebook*, buku, artikel atau skripsi yang berhubungan dan mendukung proses penelitian.

### 2.2. Pengumpulan Data

Tahap pengumpulan data merupakan tahapan memperoleh data secara *primer* (langsung) dari PT. HB. Data yang digunakan merupakan data perhari dimulai dari Juli 2017 hingga September 2017 sebanyak 180 data. Jumlah variabel *Input* yang digunakan dalam penelitian ini terdiri dari 4 *Input*, yaitu jumlah produksi, harga, jumlah agen penjualan dan jumlah produk terjual.

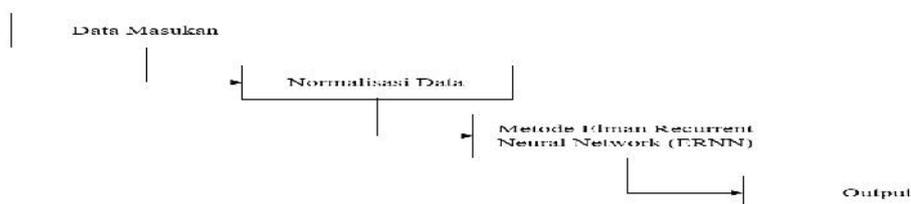
### 2.3. Analisa dan Perancangan

#### 2.3.1. Analisa

Tahap analisa merupakan tahapan menganalisa permasalahan yang berhubungan dengan penelitian. Analisa diperlukan agar keputusan yang diambil sesuai dengan tujuan. Tahapan analisa pada penelitian ini dibagi menjadi tiga bagian, yaitu analisa proses dan analisa dialog.

##### 2.3.1.1. Analisa Proses

Tahapan analisa proses penerapan algoritma ERNN untuk memprediksi penjualan tempo PT. HB dapat dilihat pada gambar berikut :



Gambar 2. Analisa Proses Penerapan Algoritma ERNN untuk Peramalan Penjualan

#### 1. Data Masukan

Data Masukan merupakan langkah awal yang dilakukan pada tahapan analisa proses. Pada tahap ini dilakukan proses penentuan variabel yang akan digunakan pada penelitian ini.

Variabel masukan yang digunakan ada 4 yaitu jumlah produksi (Pcs), harga (Rp), jumlah agen penjualan (Orang), dan jumlah produk terjual (Pcs). Pada tahap ini terjadi proses pembagian data, yaitu data latih dan data uji. Skenario pembagian data latih dan data uji pada penelitian ini yaitu 90:10, 80:20, dan 70:30.

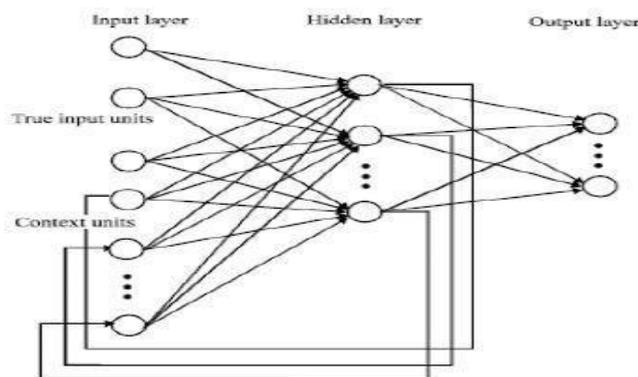
2. Normalisasi Data

Normalisasi data dilakukan untuk memperoleh data dalam ukuran yang lebih sedikit dibandingkan dengan data asli tanpa menghilangkan nilai dari data asli menggunakan persamaan berikut :

$$x^* = \frac{x - \min_x}{\max_x - \min_x} \tag{1}$$

3. Metode *Elman Recurrent Neural Network* (ERNN)

*Elman Recurrent Neural Network* (ERNN) merupakan salah satu jenis jaringan berulang yang mempunyai lapisan konteks yang akan direferensikan. Dalam penerapannya, ERNN akan dilakukan pelatihan yang diawasi dengan menggunakan algoritma *backpropagation* berdasarkan *Input* dan *target* yang dimasukkan. Terdapat beberapa parameter yang berpengaruh dalam proses pelatihan, yaitu inisialisasi bobot, jenis *Inputan*, jumlah *neuron* yang tersembunyi, tingkat pembelajaran, dan faktor momentum. Jika terjadi kesalahan dalam menentukan parameter, maka akan membutuhkan waktu yang lama untuk proses tersebut (Sundaram dan Ramesh, 2015). Gambar 3. merupakan gambar arsitektur *Elman Recurrent Neural Network*.



Gambar 3. Arsitektur Jaringan *Elman Recurrent Neural Network*

ERNN mempunyai *loop* yang tertutup di dalam topologi jaringan. Unit-unit dari kelompok tertentu yang memperoleh sinyal umpan balik dari langkah yang sebelumnya disebut dengan Unit Kompleks. Arsitektur jaringan ERNN umumnya adalah jaringan yang memiliki dua *layer* dengan proses umpan balik dari *output layer* pertama, dan *Input layer* pertama. Koneksi berulang yang terdapat pada jaringan Elman menjadikan Elman lebih baik dalam melakukan pendeteksian dan menghasilkan pola waktu yang bervariasi. Jaringan Elman memiliki *neuron tansig* yang terdapat pada lapisan tersembunyi dan *neuron purelin* yang terdapat pada lapisan *output*. Peningkatan kompleksitas dapat dilakukan dengan cukupnya *neuron* yang terdapat pada lapisan tersembunyi. Ketika diberikan nilai bobot dan bias yang sama, akan tetapi diberikan sebuah *Inputan* yang identik pada waktu tertentu, *output* yang dihasilkan bisa berbeda karena umpan balik yang berbeda untuk setiap *statenya* (Swathi dkk, 2015). Pada arsitektur ERNN terdapat sebuah *layer* yang disebut dengan *context layer*. *Context layer* berfungsi untuk membuat proses iterasi menjadi lebih cepat sehingga dapat memudahkan dalam memperoleh proses keluaran (Permana dkk, 2014).

**Algoritma ERNN**

Langkah-langkah algoritma ERNN adalah sebagai berikut :

1. Memberikan nilai inisialisasi bobot antara *Input-hidden layer* dan *hidden- output layer*, *learning rate*, toleransi *error*, dan maksimal *epoch*.
2. Setiap unit *Input*  $x_i$  akan menerima sinyal *Input* dan kemudian sinyal *Input* tersebut akan dikirimkan pada seluruh unit yang terdapat pada *hidden layer*.
3. Setiap unit *hidden layer*  $net_j(t)$  akan ditambahkan dengan nilai *Input*  $x_i$  yang akan dikalikan dengan  $v_{ji}$  dan dikombinasikan dengan *context layer*  $y_h(t-1)$  yang dikalikan bobot  $u_{jh}$  dijumlahkan dengan bias  $q$  dengan persamaan (2).

$$net_j = \left( \sum_i^n x_i(t)v_{ji} + \sum_h^n y_h(t-1)u_{jh} + \theta_j \right) \quad (2)$$

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*

Untuk fungsi pengaktif *neuron* yang digunakan adalah *sigmoid biner* dengan persamaan (3) dan (4)

$$y_j(t) = f(net_j(t)) \quad (3)$$

$$f(net_j) = \frac{1}{1+e^{-net_j}} \quad (4)$$

4. Setiap unit yang terdapat pada  $y_k$  akan ditambahkan dengan nilai keluaran pada *hidden layer*  $y_j$  yang dikalikan dengan bobot  $w_{kj}$  dan dijumlahkan dengan bias bagian *hidden layer* agar mendapatkan keluaran, maka  $net_k$  akan dilakukan perhitungan dalam fungsi pengaktif menjadi  $y_k$  dengan persamaan (5) dan (6).

$$net_k(t) = \left( \sum_j^m y_j(t)w_{kj} \right) + \theta_k \quad (5)$$

$$y_k(t) = g(net_k(t)) \quad (6)$$

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  $net_k(t)$

5. Setiap unit *output* akan menerima pola target  $t_k$  sesuai dengan pola masukan pada saat proses pelatihan dan akan dihitung nilai *error*-nya dan dilakukan perbaikan terhadap nilai bobot. Proses perhitungan nilai *error* dalam turunan fungsi pengaktif dengan persamaan (7).

$$\delta_k = g'(net_k)(t_k - y_k) \tag{7}$$

Keterangan:

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

$t_k$  = target

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

Perhitungan perbaikan nilai bobot dengan persamaan (8)

$$w_{kj} = \alpha \delta_k y_j \tag{8}$$

Keterangan:

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

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

Perhitungan perbaikan nilai kolerasi dengan persamaan (9)

$$\theta_k = \alpha \delta_k \tag{9}$$

Keterangan:

$\theta_k$  = hasil perbaikan nilai bias

dan nilai  $\delta_k$  yang diperoleh akan digunakan pada semua unit lapisan sebelumnya.

- Setiap *output* yang menghubungkan antara unit *output* dan unit *hidden layer* akan dikalikan dengan  $\delta_k$  dan dijumlahkan sebagai masukan unit yang selanjutnya dengan persamaan (10).

$$\delta_{-net_j} = \sum \delta_k w_{kj} \tag{10}$$

Kemudian dikalikan dengan turunan fungsi aktivasi untuk memperoleh galat dengan persamaan (11).

$$\delta_j = \delta_{-net_j} f'(net_j) \tag{11}$$

Selanjutnya lakukan perhitungan perbaikan terhadap nilai bobot dengan persamaan (12).

$$v_{kj} = \alpha \delta_j x_i \tag{12}$$

Hitung perbaikan nilai kolerasi dengan persamaan (13).

$$\theta_j = \alpha \delta_j \tag{13}$$

- Setiap unit *output* akan dilakukan perbaikan terhadap nilai bobot dan biasanya dengan persamaan (14).

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) \Delta w_{kj} \tag{14}$$

Tiap unit *hidden layer* juga dilakukan perbaikan terhadap nilai bobot dan biasanya dengan persamaan (15).

$$v_{kj}(\text{baru}) = v_{kj}(\text{lama}) \Delta v_{kj} \tag{15}$$

- Setiap *ouput* akan dibandingkan dengan target  $t_k$  yang diinginkan, agar

memperoleh nilai *error* (E) keseluruhan dengan persamaan (16).

$$E(t) = \frac{1}{2} \sum_{i=1}^k (t_k - y_k)^2 \quad (16)$$

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.

### 2.3.1.2. Analisa Dialog

Analisa dialog merupakan sebuah gambaran kerja dari alur sistem yang dibangun agar dapat menjadi sebuah informasi. Analisa dialog pada penelitian ini berupa *Data Flow Diagram* (DFD), *Entity Relationship Diagram* (ERD).

1. *Data Flow Diagram* (DFD)

Sebagai gambaran sistem Peramalan Penjualan tempe yang memiliki sistem kerja antara satu fungsi dengan fungsi yang lain yang memiliki hubungan dengan aliran data dan data *store*.

2. *Entity Relationship Diagram* (ERD)

Sebagai penggambaran relasi anantara satu entitas dengan entitas yang lainnya.

### 2.3.2. Perancangan

Tahap perancangan merupakan tahapan perancangan terhadap sistem yang akan dibangun, agar user mengerti pada saat melakukan implementasi terhadap sistem tersebut. Tahap perancangan dibagi menjadi tiga bagian, yaitu perancangan *database*, perancangan struktur menu, dan perancangan *interface*.

1. Perancangan Database

Tahap perancangan *database* merupakan tahapan pembuatan *database* yang berisikan tabel-tabel, *field* dan atribut yang akan digunakan dalam pembuatan sistem yang akan dibangun.

2. Perancangan Struktur Menu

Tahap perancangan struktur menu merupakan tahapan yang berisikan tampilan yang terdapat pada menu dan submenu. Hal ini diperlukan agar pada saat digunakan, pengguna mengetahui fungsi dan maksud dari tampilan menu-menu tersebut.

3. Perancangan *Interface*

Tahap perancangan *interface* merupakan tahapan yang digunakan sebagai sarana pengembangan untuk melakukan komunikasi yang mudah dalam proses pengaplikasiannya. Perancangan *interface* lebih menekankan tentang tampilan dan tombol-tombol agar dapat dimengerti oleh pengguna.

## 2.4. Implementasi dan Pengujian

### 2.4.1. Implementasi

Implementasi merupakan tahapan awal dilakukannya testing terhadap sistem yang dibangun. Dalam melakukan penerapan pembuatan aplikasi, dibutuhkan perangkat keras dan perangkat lunak. Perangkat keras yang digunakan yaitu:

*Processor* : Intel® Core(TM) i7-4710HQ CPU @ 2.50GHz

*Memory* : 8.00 GB

Sedangkan untuk perangkat lunaknya yaitu:

*Operation System* : Windows 10 Home

Bahasa Pemrograman : PHP

DBMS : *MySQL*

Browser : Google Chrome

#### 2.4.2. Pengujian

Pengujian merupakan tahapan untuk mengetahui bagaimana kinerja sistem yang telah dibangun. Apakah sistem tersebut telah sesuai dengan keinginan pengguna. Ada beberapa cara yang dilakukan dalam tahap pengujian, yaitu pengujian akurasi, *Mean Square Error* (MSE), dan Galat.

##### 1. Akurasi

Akurasi merupakan hasil rata-rata dari pengukuran nilai yang sebenarnya. Perhitungan akurasi dilakukan untuk menguji berapa persen ketepatan terhadap peramalan (Permana dkk, 2014). Untuk menghitung nilai akurasi menggunakan persamaan :

$$akurasi = 1 - \frac{output - target}{target} \times 100\% \quad (17)$$

##### 2. Mean Square Error (MSE)

*Mean Square Error* (MSE) merupakan pengujian yang dilakukan sebagai tolak ukur analisis kuantitatif dalam menentukan kualitas sebuah *output* serta keunggulan dari metode yang digunakan. Perhitungan MSE menggunakan persamaan berikut :

$$MSE = \sum \frac{E_t^2}{n} \quad (18)$$

Untuk memperoleh nilai galat digunakan persamaan :

$$E_t = x_t - f_t \quad (19)$$

Keterangan:

- $E_t$  = nilai galat
- $x_t$  = data aktual pada periode ke t
- $f_t$  = data ramalan pada periode ke t
- n = jumlah data

##### 3. Galat (*error*)

Galat (*error*) merupakan suatu alat ukur yang digunakan untuk mengukur ketepatan jaringan syaraf tiruan terhadap data target pembelajaran. Galat pada keluaran jaringan syaraf tiruan ini merupakan selisih antar keluaran sebenarnya (*current output*) dengan keluaran yang diinginkan (*desired output*) dari masukkan data tertentu.. Semakin kecil tingkat *error*, maka semakin tinggi nilai tingkat akurasiya. Rumus tingkat *error* dapat dilihat pada persamaan berikut:

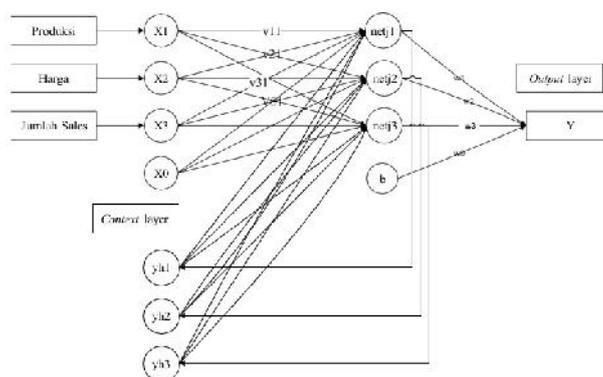
$$Error = \frac{output - target}{target} * 100\% \quad (20)$$

### 3. Hasil dan Pembahasan

Bagian Hasil dan Pembahasan memuat hasil-hasil dari penelitian serta pembahasan menyeluruh dari masing-masing hasil yang didapatkan dari penelitian yang dibahas. Pada penelitian ini akan menunjukkan hasil pembangunan sistem peramalan penjualan produk tempe PT. HB menggunakan metode ERNN, yang terdiri atas : Pembangunan Arsitektur Jaringan Syaraf Tiruan, Hasil Pengujian Akurasi, Hasil Pengujian MSE, Hasil Pengujian Galat, Hasil Pengujian Implementasi Sistem serta Antarmuka Implementasi Sistem yang Dibangun.

#### 3.1. Arsitektur Jaringan Syaraf Tiruan ERNN Yang Dibangun

Berikut dapat dilihat arsitektur yang dibangun.



Gambar 4. Arsitektur Jaringan Syaraf Tiruan ERNN

1. Data masukan merupakan data yang diperoleh dari informasi penjualan tempo. Jumlah *Inputan* yang digunakan adalah tiga, yaitu produksi dengan inisialisasi  $X_1$ , harga dengan inisialisasi  $X_2$ , dan jumlah agen penjualan (*sales*) dengan inisialisasi  $X_3$ ,  $X_0$  merupakan inisialisasi untuk nilai *bias* yang akan digunakan dalam proses perhitungan.
2. Nilai *Input* akan dilakukan normalisasi terlebih dahulu, kemudian akan ditransfer dari *Input layer* menuju *hidden layer* menggunakan fungsi aktivasi *sigmoid biner*.
3. Pada *hidden layer* terdapat tujuh *neuron* yang setiap *neuronnya* akan diteruskan menuju *context layer*. Jumlah setiap *neuron* yang terdapat pada *hidden layer* sama jumlahnya dengan jumlah *context layer*.
4. Proses perhitungan dapat dilakukan setelah dilakukan pemberian nilai terhadap parameter awal, diantaranya yaitu nilai bobot  $v$ , nilai bobot  $w$  dan nilai *bias*.
5. Bobot keluaran *hidden layer* akan *dicopy* menuju *context layer* dan akan dikembalikan ke *hidden layer* kemudian akan diteruskan menuju *output layer* yang memiliki sebuah *output*. *Neuron* yang terdapat pada *output layer* disimbolkan dengan  $Y$ .

### 3.2. Hasil Pengujian Akurasi

Pengujian akurasi dilakukan dengan membuat beberapa skenario uji yaitu dengan merubah nilai-nilai parameter *learning rate* (dimulai dari 0,1 hingga 0,9) dan pembagian data latih dan data uji. Nilai maksimum *epoch* pada skenario uji ini tetap yaitu 500, begitu juga nilai *minimum error* sebesar 0,0001. Hasil yang diperoleh adalah nilai akurasi terbaik dicapai pada set parameter *learning rate* bernilai 0,9 untuk seluruh skenario perbandingan pembagian data latih dan data uji.

Tabel 1. Hasil Pengujian Akurasi

| Learning Rate | Perbandingan Data Latih dan Data Uji = 70% : 30% |                 | Perbandingan Data Latih dan Data Uji = 80% : 20% |                 | Perbandingan Data Latih dan Data Uji = 90% : 10% |                 |
|---------------|--------------------------------------------------|-----------------|--------------------------------------------------|-----------------|--------------------------------------------------|-----------------|
|               | Min. Error                                       | Akurasi (%)     | Min. Error                                       | Akurasi (%)     | Min. Error                                       | Akurasi (%)     |
| 0.1           | 0.0001                                           | 94.534487472393 | 0.0001                                           | 94.20256156479  | 0.0001                                           | 93.550716498645 |
| 0.2           | 0.0001                                           | 94.621943087868 | 0.0001                                           | 94.469211026886 | 0.0001                                           | 93.865265757291 |
| 0.3           | 0.0001                                           | 95.3250003006   | 0.0001                                           | 94.25908173374  | 0.0001                                           | 94.169235356952 |
| 0.4           | 0.0001                                           | 96.061444152424 | 0.0001                                           | 94.459010710909 | 0.0001                                           | 94.509201288096 |
| 0.5           | 0.0001                                           | 96.359179722911 | 0.0001                                           | 94.58150960317  | 0.0001                                           | 94.856090360808 |
| 0.6           | 0.0001                                           | 96.335686943326 | 0.0001                                           | 94.571238078538 | 0.0001                                           | 95.269193447196 |
| 0.7           | 0.0001                                           | 96.367467658235 | 0.0001                                           | 94.565997462091 | 0.0001                                           | 95.770458996234 |
| 0.8           | 0.0001                                           | 96.816096186943 | 0.0001                                           | 94.650104557377 | 0.0001                                           | 96.121881909195 |
| 0.9           | 0.0001                                           | 96.921950710643 | 0.0001                                           | 94.855301520547 | 0.0001                                           | 96.260101727486 |

### 3.3. Hasil Pengujian Mean Square Error (MSE)

Berdasarkan hasil pengujian menggunakan MSE pada ketiga skenario pengujian, dapat dilihat bahwa nilai MSE semakin menurun dari epoch ke-1 hingga epoch ke-500. Ini membuktikan bahwa penggunaan algoritma ERNN sudah baik dan tepat dalam menyelesaikan persoalan memprediksi penjualan.

Tabel 2. Hasil Pengujian MSE

| Perbandingan Data Latih dan Data Uji = 70% : 30% |                    | Perbandingan Data Latih dan Data Uji = 80% : 20% |                    | Perbandingan Data Latih dan Data Uji = 90% : 10% |                    |
|--------------------------------------------------|--------------------|--------------------------------------------------|--------------------|--------------------------------------------------|--------------------|
| Epoch                                            | MSE                | Epoch                                            | MSE                | Epoch                                            | MSE                |
| 1                                                | 0.017078831523522  | 1                                                | 0.018038851881605  | 1                                                | 0.017619149302808  |
| 2                                                | 0.020052459387753  | 2                                                | 0.017593936071012  | 2                                                | 0.016773104026176  |
| 3                                                | 0.019994397223786  | 3                                                | 0.017518248779335  | 3                                                | 0.016701879681139  |
| 4                                                | 0.01993836138542   | 4                                                | 0.017436946142251  | 4                                                | 0.016621495724487  |
| 5                                                | 0.019882927261868  | 5                                                | 0.017346175489821  | 5                                                | 0.016524353025352  |
| ...                                              | .....              | ....                                             | .....              | ...                                              | .....              |
| 496                                              | 0.0014787668709159 | 496                                              | 0.001582532709527  | 496                                              | 0.0018942523423718 |
| 497                                              | 0.0014784343435747 | 497                                              | 0.0015820208112851 | 497                                              | 0.0018938670252034 |
| 498                                              | 0.0014781040189736 | 498                                              | 0.0015815082515339 | 498                                              | 0.0018934732521849 |
| 499                                              | 0.0014777758831766 | 499                                              | 0.0015809950259954 | 499                                              | 0.0018930709340165 |
| 500                                              | 0.001477449922319  | 500                                              | 0.0015804811304322 | 500                                              | 0.0018926599835789 |

### 3.3. Hasil Pengujian Galat (Error)

Hasil pengujian error yang dilakukan untuk 10 data uji pada perbandingan data 90:10, 80:20, 70:30 dengan epoch 500, learning rate 0,9, dan minimal error 0,0001 menunjukkan skenario terbaik pengujian pada pembagian data 70:30 yang memiliki rata-rata nilai error terkecil yakni sebesar 1,31%.

Tabel 3. Hasil Pengujian Galat (Error)

| Data ke-               | Perbandingan Data Latih dan Data Uji = 70% : 30% |          |              | Perbandingan Data Latih dan Data Uji = 80% : 20% |          |        | Perbandingan Data Latih dan Data Uji = 90% : 10% |              |       |  |
|------------------------|--------------------------------------------------|----------|--------------|--------------------------------------------------|----------|--------|--------------------------------------------------|--------------|-------|--|
|                        | target                                           | prediksi | error        | target                                           | prediksi | error  | target                                           | prediksi     | error |  |
| 1                      | 1145                                             | 1150     | 0,44%        | 1186                                             | 1213.    | 2,28%  | 1352                                             | 1308         | 3,25% |  |
| 2                      | 1178                                             | 1166     | 1,02%        | 1256                                             | 1228     | 2,23%  | 1360                                             | 1313         | 3,46% |  |
| 3                      | 1138                                             | 1163     | 2,20%        | 1095                                             | 1229     | 12,24% | 1419                                             | 1332         | 6,13% |  |
| 4                      | 1182                                             | 1199     | 1,44%        | 1221                                             | 1220     | 0,08%  | 1401                                             | 1315         | 6,14% |  |
| 5                      | 1187                                             | 1192     | 0,42%        | 1252                                             | 1228     | 1,92%  | 1312                                             | 1299         | 0,99% |  |
| 6                      | 1256                                             | 1230     | 2,07%        | 1219                                             | 1213     | 0,49%  | 1354                                             | 1306         | 3,55% |  |
| 7                      | 1234                                             | 1241     | 0,57%        | 1226                                             | 1213     | 1,06%  | 1376                                             | 1318         | 4,22% |  |
| 8                      | 1207                                             | 1227     | 1,66%        | 1244                                             | 1226     | 1,45%  | 1352                                             | 1308         | 3,25% |  |
| 9                      | 1167                                             | 1205     | 3,26%        | 1286                                             | 1244     | 3,27%  | 1360                                             | 1313         | 3,46% |  |
| 10                     | 1205                                             | 1205     | 0,00%        | 1231                                             | 1231     | 0,00%  | 1419                                             | 1332         | 6,13% |  |
| <b>Rata-rata error</b> |                                                  |          | <b>1,31%</b> |                                                  |          |        | <b>2,5%</b>                                      | <b>4,06%</b> |       |  |

### 3.4. Hasil Pengujian Sistem (White Box)

Pada pembangunan sistem penelitian ini menggunakan pengujian white box, dimana pengujian sistem white box adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur control dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara umum dapat dikatakan bahwa pengujian white box merupakan petunjuk untuk mendapatkan program yang benar secara 100%.

Berikut ditampilkan salah satu proses pengujian *white box* pada proses normalisasi data latih dan data uji.

Tabel 4. Hasil Pengujian Sistem (*White Box*)

| NO. | Coding Program                                                                                                                                                                                                                                                                                                                           | Keterangan                                  | Hasil    |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------|----------|
| 1.  | <pre>//CARI NILAI MIN VARIABEL  \$json = \$db-&gt;query_select_array ("SELECT MIN (x1) AS MIN1, MIN (x2) AS MIN2, MIN (x3) penjualan");  \$dataX = json_decode (\$json, true);  \$dataXmin = \$dataX[0];  //CARI NILAI MAX VARIABEL  \$json = \$db-&gt;query_select_array ("SELECT MAX (x1) AS MAX1, MAX (x2) AS MAX2, MAX (x3) AS</pre> | Proses normalisasi data latih dan data uji. | Berhasil |

### 3.1. Implementasi / Antarmuka Sistem

Pada penelitian ini dibangun sebuah aplikasi yang akan lebih memudahkan pengguna dalam melakukan peramalan penjualan. Terdiri dari beberapa menu yakni : beranda, pengguna, penjualan, proses peramalan menggunakan ERNN dan hasil pengujian peramalan.



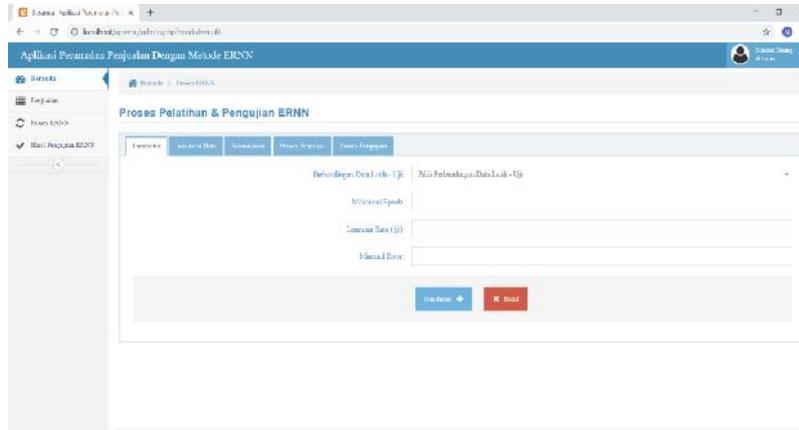
Gambar 5. Halaman Utama Aplikasi Peramalan Penjualan Menggunakan ERNN

Pada menu Data Penjualan, pengguna dapat mengolah data penjualan tempo dengan cara menambahkan data secara manual maupun secara otomatis dengan mengimport dari sebuah *file* dengan ekstensi xls.

| ID | Tanggal    | Penjualan | Harga | Diskon | Status      |
|----|------------|-----------|-------|--------|-------------|
| 1  | 06-03-2017 | 1585      | 4900  | 0      | Diak (Diak) |
| 1  | 06-03-2017 | 1586      | 4900  | 0      | Diak (Diak) |
| 1  | 06-03-2017 | 1589      | 4900  | 0      | Diak (Diak) |
| 4  | 06-03-2017 | 1591      | 4900  | 0      | Diak (Diak) |
| 1  | 06-03-2017 | 1592      | 4900  | 0      | Diak (Diak) |
| 5  | 06-03-2017 | 1599      | 4900  | 0      | Diak (Diak) |
| 7  | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 1  | 06-03-2017 | 1598      | 4900  | 0      | Diak (Diak) |
| 9  | 06-03-2017 | 1599      | 4900  | 0      | Diak (Diak) |
| 10 | 06-03-2017 | 1599      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |
| 11 | 06-03-2017 | 1595      | 4900  | 0      | Diak (Diak) |

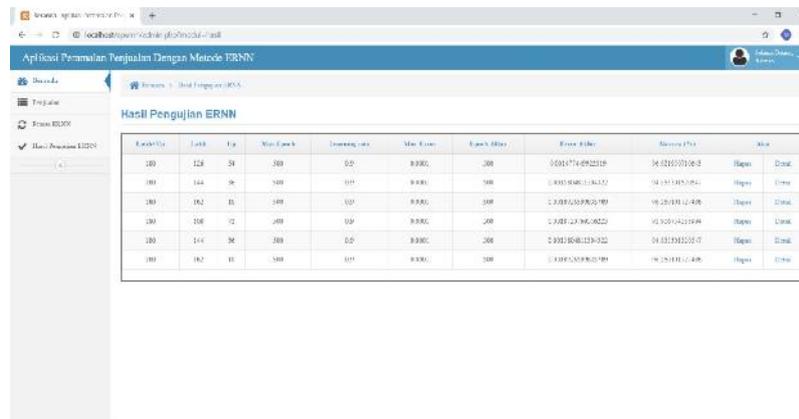
Gambar 6. Antarmuka Sistem Pengolahan Data Penjualan

Pada menu Proses ERNN terdapat sub-sub menu yang menampilkan secara keseluruhan proses peramalan penjualan tempo menggunakan algoritma ERNN, seperti : sub menu Parameter, sub menu Inialisasi Data, sub menu Normalisasi, sub menu Proses Pelatihan dan sub menu Proses Pengujian.



Gambar 7. Antarmuka Sistem Proses Pelatihan dan Pengujian ERNN

Pada menu Hasil Pengujian ERNN ditampilkan secara lengkap dan jelas hasil dari pengujian peramalan penjualan tempo.



Gambar 8. Antarmuka Sistem Hasil Pengujian ERNN

#### 4. Kesimpulan

Kesimpulan dari pengujian penggunaan metode *Elman Recurrent Neural Network* (ERNN) untuk peramalan penjualan tempo yaitu:

1. Penerapan metode *Elman Recurrent Neural Network* (ERNN) memberikan hasil yang baik dalam melakukan peramalan penjualan tempo.
2. Pengujian akurasi dilakukan dengan meng *input* kan 500 *epoch* dan nilai *learning rate*. Nilai *learning rate* yang digunakan yaitu 0,1 hingga 0,9. Pembagian data dilakukan sebanyak tiga kali percobaan, yaitu 70% data latih, 30% data uji, 80% data latih dan 20% data uji, 90% data latih dan 10% data uji. Hasil pengujian akurasi yang tertinggi yaitu 96,92% yang berada pada pembagian data 70% data latih dan 30% data uji dengan nilai *learning rate* 0,9.
3. Proses pengujian MSE memperlihatkan bahwa semakin banyak *epoch* yang digunakan semakin kecil nilai *error*, sehingga membuat akurasi hasil peramalan semakin tinggi.
4. Nilai akurasi akan semakin tinggi jika peng *input* an nilai *learning rate* yang semakin

besar.

### Daftar Pustaka

- [1] Agus Aan Jiwa Permana, W. P. Sistem Evaluasi Kelayakan Mahasiswa Magang Menggunakan Elman Recurrent Neural Network. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*. 2014; Vol. 8(1), 37–48.
- [2] Chasani, M. L. *Aplikasi Peramalan Tagihan Listrik Dengan Jaringan Syaraf Tiruan*. SNATI : Seminar Nasional Aplikasi Teknologi Informasi. Yogyakarta. 2013; 28–31.
- [3] Maulida, Ana. Penggunaan Elman Recurrent Neural Network Dalam Peramalan Suhu Udara Sebagai Faktor Yang Mempengaruhi Kebakaran Hutan. Skripsi. Institiut Pertanian Bogor, Bogor; 2011.
- [4] Mutiara, A. B., Rifky, A., Muslim, A., & Teddy, O. *Testing Implementasi Website Rekam Medis Elektronik Opeltgunasys Dengan Metode Acceptance Testing*. Seminar Ilmiah Nasional Komputer Dan Sistem Intelijen. Jakarta. 2014; 1–7.
- [5] Sani, Dealucky. Penerapan Elman Neural Network Pada Peramalan Konsumsi Listrik Jangka Pendek Di PT . PLN APP Malang. Skripsi. Universitas Brawijaya. Malang, 2014; 441–444.
- [6] Sanny, L., Sarjono, H., Smoothing, E., Trend, W., Smoothing, E., & Moving, W. Peramalan Jumlah Siswa/I Sekolah Menengah Atas Swasta Menggunakan Enam Metode Forecasting. *Forum Ilmiah*. Vol.10 (2), 2013;198–208.
- [7] Sundaram, N. M., & Ramesh, P. N. *Optimization of Training phase of Elman Neural Networks by suitable adjustments on the Network parameters*. International Conference on Systems, Science, Control, Communication, Engineering and Technology. India. 2015;229–235