

**PERBANDINGAN PREDIKSI
HARGA SAHAM DENGAN
MENGUNAKAN METODE
JARINGAN SYARAF TIRUAN
DAN FUZZY TSUKAMOTO**
*COMPARISON OF SHARE PRICE
PREDICTION USING
ARTIFICIAL NEURAL NETWORK
AND FUZZY TSUKAMOTO*

ABSTRACT

Dengan kemajuan teknologi yang semakin pesat, hampir semua hal bisa di prediksi menggunakan teknologi. Salah satunya adalah memprediksi harga saham. Prediksi harga saham sangat berguna bagi investor-investor saham dalam hal mengambil keputusan dalam pembelian saham di masa depan. Pengambilan keputusan yang tepat dalam pembelian saham merupakan syarat mutlak dalam bisnis investasi saham ini. Untuk itu diperlukan adanya suatu metode yang dapat digunakan untuk memprediksi harga saham di masa depan. Saham terdiri atas 2 jenis, yaitu common stock atau saham biasa dan preferred stock atau saham preferen common stock adalah saham yang paling populer di masyarakat dan bagi emiten, common stock juga di gunakan untuk mencari dana yang besar dari investor Oleh karena itu, penentuan pembelian saham yang tepat adalah hal yang wajib di butuhkan oleh broker atau trader dan investor. Kesalahan dalam melakukan pembelian tau penjualan saham dapat mengakibatkan kerugian yang fatal bagi broker atau trader dan investor.

Muchlas(2007) telah merancang prediksi harga saham berbasis web dengan sistem inferensi Fuzzy Tsukamoto berdasarkan aspek teknis, yaitu : harga penawaran, harga permintaan dan laju inflasi. Untuk mengetahui apakah ada metode yang lebih baik, maka pada penelitian ini dilakukan percobaan dengan menggunakan metode Jaringan Syaraf Tiruan Backpropagation. Dengan menggunakan inputan yang sama, dapat diketahui metode mana yang lebih akurat dalam memprediksi harga saham.

Kata kunci: Fuzzy Tsukamoto, Jaringan Syaraf Tiruan, Backpropagation

1. LATAR BELAKANG MASALAH

Dengan kemajuan teknologi yang semakin pesat, hampir semua hal bisa di prediksi menggunakan teknologi. Salah satunya adalah memprediksi harga saham. Prediksi harga saham sangat berguna bagi investor-investor saham dalam hal mengambil keputusan dalam pembelian saham di masa depan. Pengambilan keputusan yang tepat dalam pembelian saham merupakan syarat mutlak dalam bisnis investasi saham ini.

Emiten atau perusahaan penerbit surat berharga juga mendapat keuntungan dari perdagangan saham ini dengan menjual sebagian atau bahkan seluruh saham mereka pada investor. Dana yang besar merupakan keuntungan yang di diharapkan dapat

membantu perusahaan mereka menjadi lebih berkembang. Bila perusahaan sudah dalam keadaan hampir bangkrut, menjual seluruh saham mereka adalah pilihan terbaik agar perusahaan bisa tetap berjalan dan karyawan di perusahaan tersebut tidak di PHK.

Saham terdiri atas 2 jenis, yaitu common stock atau saham biasa dan preferred stock atau saham preferen common stock adalah saham yang paling populer di masyarakat dan bagi emiten, common stock juga di gunakan untuk mencari dana yang besar dari investor.

Oleh karena itu, penentuan pembelian saham yang tepat adalah hal yang wajib di butuhkan oleh broker atau trader dan investor. Kesalahan dalam melakukan pembelian tau penjualan saham dapat mengakibatkan kerugian yang fatal bagi broker atau trader dan investor.

Berdasarkan masalah di atas, dibutuhkan aplikasi yang memungkinkan para investor saham untuk memprediksi harga saham dengan akurat. Muchlas(2007) telah merancang prediksi harga saham berbasis web dengan sistem inferensi Fuzzy Tsukamoto berdasarkan aspek teknis,

yaitu : harga penawaran, harga permintaan dan laju inflasi.

Pada penelitian ini, dengan menggunakan matlab, akan merancang prediksi harga saham namun dengan menggunakan metode Jaringan Syaraf Tiruan dengan aspek yang sama yaitu : harga penawaran, harga permintaan dan laju inflasi. Di harapkan dengan metode ini, didapatkan hasil prediksi saham yang lebih akurat dari metode yang telah ada sebelumnya.

2. SISTEM INFERENSI FUZI TSUKAMOTO

Pada metode Tsukamoto, setiap konsekuen pada aturan yang berbentuk IF-Then harus direpresentasikan dengan suatu himpunan Fuzi dengan fungsi keanggotaan yang monoton. Sebagai hasilnya, output hasil inferensi dari tiap-tiap aturan diberikan dengan tegas (crisp) berdasarkan α -predikat (fire strength). Hasil akhirnya diperoleh dengan menggunakan rata-rata terbobot.

Misal ada 2 variabel input, var-1(x) dan var-2(y) serta 1 variabel output var-3(z), dimana var-1 terbagi atas 2 himpunan yaitu A1 dan A2 dan var-2 terbagi atas himpunan B1 dan B2. Sedangkan var-3 juga terbagi atas 2

himpunan yaitu C1 dan C2. Ada dua aturan yang digunakan yaitu: [R1] IF (x is A1) and (y is B2) THEN (z is C1) [R2] IF (x is A2) and (y is B1) THEN (z is C2).

3. JARINGAN SYARAF TIRUAN FUZZY TSUKAMOTO

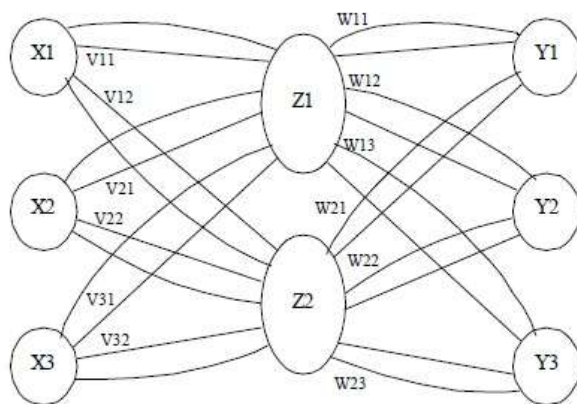
Propagasi balik merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma propagasi balik menggunakan error output untuk mengubah nilai bobot-bobotnya dalam arah mundur (backward). Untuk mendapatkan error ini, tahap perambatan maju (forward propagation) harus dikerjakan terlebih dahulu. Pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi sigmoid, yaitu :

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

Jaringan propagasi balik dikembangkan oleh Paul Werbos (Valurru B. Rao and Hayagriva V

Rao, 1993;87) dan hampir 80% dari seluruh jaringan syaraf tiruan yang ada dalam perkembangannya menggunakan jaringan ini karena mudah dalam proses belajarnya. Jaringan syaraf tiruan propagasi balik memiliki kemampuan untuk menentukan hubungan antara sekelompok pola masukan dengan sekelompok pola keluaran yang diberikan dan menggunakan hubungan ini pada saat diberikan pola masukan baru. Suatu fungsi aktivasi propagasi balik memiliki ciri utama yaitu berkesinambungan dan adanya peningkatan/perbaikan yang berulangulang. Aplikasi propagasi balik yaitu memetakan masukan (input) terhadap target keluaran (output). Tujuan pemetaan adalah untuk melatih jaringan mencapai suatu keseimbangan antara kemampuan merespon pola masukan yang digunakan dalam pelatihan dengan kemampuan untuk memberi respon masukan yang disesuaikan.

Konfigurasi jaringan propagasi balik bisa dilihat pada gambar 2.8



Gambar 2.8 Konfigurasi Jaringan Syaraf Tiruan Propagasi balik.

Prosedur belajar propagasi balik menggunakan metode gradien descent dengan paradigma belajar supervised, sehingga pasangan data input-output (set data) mutlak harus tersedia. Data ini akan digunakan sebagai pembimbing dalam mengenali polanya. Untuk dapat mengubah bobot sinapsis, maka proses komputasi dilakukan dengan cara arah maju (forward) dan arah mundur (backward). Dengan mengacu pada gambar 2.8, prosedur pelatihan jaringan beserta penjelasannya : pertama sebuah vektor masukan, $X = X_{p1}, X_{p1}, \dots, X_{pN}$ diberikan pada

lapisan masukan jaringan dengan p menyatakan pasangan vektor ke- p . Sel-sel masukan mendistribusikan nilai-nilai masukan tersebut ke sel pada lapisan dalam. Jumlah masukan keseluruhan (net input) untuk sel lapisan dalam ke- j dihitung sebagai penjumlahan dari keluaran sel-sel lapisan masukan yang dikalikan dengan bobot interkoneksinya ditambah dengan suatu besaran yang disebut bobot bias. Bias satuan ini memberikan nilai masukan fiktif 1 pada bobot bias dan pada bobot lainnya. Bobot bias diperlakukan sama dan berpartisipasi dalam proses belajar seperti halnya bobot-bobot lainnya. Masukan net pada sel tersembunyi ke- j adalah. Proses belajar dari model jaringan propagasi balik adalah sebagai berikut :

- Inisialisasi bobot (ambil bobot awal dengan nilai random yang cukup kecil).
- Tetapkan : Maksimum Epoch, Target Error, dan *Learning Rate* (α).

- Inisialisasi : Epoch = 0, MSE=1.
- Kerjakan langkah-langkah berikut selama (Epoch < Maksimum Epoch) dan (MSE > Target error).

1. Epoch = Epoch+1
2. Untuk tiap-tiap elemen input yang akan dilakukan pembelajaran kerjakan:

Feedforward :

- a. Tiap-tiap unit input (x_i , $i=1,2,3,\dots,n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan yang ada di atasnya (lapisan tersembunyi).
- b. Tiap-tiap unit pada suatu lapisan tersembunyi (z_j , $j=1,2,3,\dots,p$)

menjumlahkan sinyal-sinyal terbobot :

$$y_in_k \bullet b2_k \bullet \sum_{j=1}^p z_j \cdot W_{jk}$$

gunakan fungsi aktivasi untuk menghitung sinyal outputnya :

$$y_k \bullet f(y_in_k)$$

dan kirimkan sinyal tersebut ke unit di lapisan atasnya.

- c. Tiap-tiap unit output (y_k , $k=1,2,3,\dots,m$) menjumlahkan sinyal-sinyal terbobot :

$$y_in_k \bullet b2_k \bullet \sum_{j=1}^p z_j \cdot W_{jk}$$

gunakan fungsi aktivasi untuk menghitung

sinyal outputnya
:

$$y_k = f(y_{in_k})$$

Backpropogatio

n :

- d. Tiap-tiap unit output (y_k , $k=1,2,3,\dots,m$) menerima target pola yang berhubungan dengan pola input pembelajaran, hitung informasi error :

$$d_k = (t_k - y_k)$$

$$z_j = \sum_k w_{jk} \cdot d_k$$

$$b_j = z_j$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai w_{jk}) :

$$\Delta w_{jk} = \eta \cdot d_k \cdot z_j$$

Hitung koreksi bias :

$$\Delta b_k = d_k$$

Langkah (d) ini dilakukan sebanyak jumlah lapisan tersembunyi, yaitu menghitung informasi error dari suatu lapisan tersembunyi ke lapisan tersembunyi sebelumnya.

- e. Tiap-tiap unit tersembunyi ($z_j, j=1,2,3,\dots,p$) menjumlahkan delta input :

$$d_{in_j} = \sum_k d_k \cdot w_{jk}$$

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

$$\delta_j = \sum_{i=0}^n \delta_{ij} \cdot f'(z_j)$$

$$\delta_{ij} = \delta_j \cdot x_j$$

$$\delta_j = \delta_j$$

kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk

memperbaiki nilai v_{ij}):

$$\Delta v_{ij} = \delta_j \cdot \delta_{ij}$$

hitung juga koreksi bias (yang nantinya akan digunakan untuk

memperbaiki nilai b_{1j}):

- f. Tiap-tiap unit output ($y_k, k=1,2,3,\dots, m$) memperbaiki bias dan bobotnya ($j=0,1,2,\dots,p$):

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

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

tiap-tiap unit tersembunyi ($z_j, j=1,2,3,\dots,p$) memperbaiki bias dan bobotnya ($i=0,1,2,\dots,n$):

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

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

3. Hirung MSE

Obyektif dari prosedur ini adalah untuk mendapatkan bentuk persamaan dan nilai koefisien dalam formula dengan meminimalkan jumlah Sum Square Error melalui model yang dikembangkan (training set). Langkah – langkah yang digunakan adalah :

1. Dimulai dengan lapisan masukan, hitung luaran

dari setiap elemen pemroses melalui lapisan luaran.

2. Hitung kesalahan pada lapisan luaran yang merupakan selisih antara data aktual dan target.

3. Transformasikan kesalahan tersebut pada kesalahan yang sesuai di sisi masukan elemen pemroses.

4. Propagasi balik kesalahan-kesalahan tersebut ke lapisan

tersembunyi.

Transformasikan

kesalahan ini pada luaran

setiap elemen pemroses ke

kesalahan yang terdapat

pada masukan.

5. Ubah seluruh bobot dengan

menggunakan kesalahan

pada sisi masukan

elemen pemroses dan

luaran elemen

pemroses yang

terhubung.

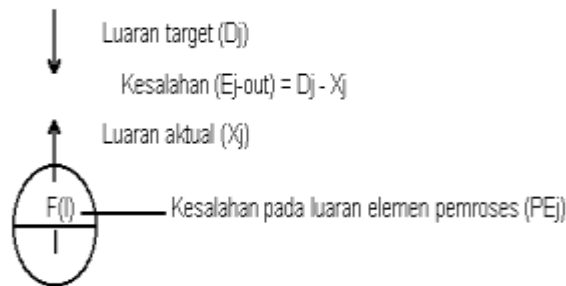
Tujuan dari perubahan bobot

untuk setiap lapisan, bukan merupakan

hal yang sangat penting. Perhitungan

kesalahan

merupakan pengukuran bagaimana jaringan dapat belajar dengan baik. Kesalahan pada luaran dari jaringan merupakan selisih antara keluaran aktual (current output) dan keluaran target (*desired output*).



Gambar 2.9 Perhitungan kesalahan pada luaran

Sum Square Error (SSE) dihitung sebagai berikut :

1. Hitung luaran prediksi atau luaran model untuk

masukan pertama

2. Hitung selisih antara nilai luaran prediksi dan nilai target atau sinyal latihan untuk setiap luaran

3. Kuadratkan setiap luaran kemudian hitung seluruhnya.

Ini merupakan kuadrat kesalahan untuk contoh latihan.

$$SSE = \sum_{j=1}^n \sum_{k=1}^m (D_{jk} - \hat{Y}_{jk})^2$$

Root Mean Square Error (RMS Error) dihitung sebagai berikut :

1. Hitung SSE

2. Hasilnya dibagi dengan perkalian antara banyaknya data pada latihan dan banyaknya luaran, kemudian diakarkan.

$$RMSE = \sqrt{\frac{SS}{N}}$$

Root Mean Square Error

Square Error

Banyaknya data pada latihan

RMSE =

SSE = Sum

N =

K =

Banyaknya

output

Langkah-langkah penelitian ini menggunakan media Matlab dengan menggunakan metode JST backpropagation.. untuk mengetahui apakah metode ini lebih baik dibanding dengan metode Fuzzy Tsukamoto, maka untuk data inputan menggunakan data yang sama dengan penggunaan metode Fuzzy Tsukamoto.

4. LANGKAH-LANGKAH

PENELITIAN

4.1 Data Input

Data Input

harga penawaran	harga permintaan	harga aktual
Rp86,500	Rp169,500	Rp10,850
Rp14,000	Rp32,500	Rp10,850
Rp11,500	Rp5,000	Rp10,800
Rp158,000	Rp3,000	Rp10,800
Rp50,500	Rp31,500	Rp10,750
Rp1,500	Rp10,500	Rp10,450
Rp500	Rp49,000	Rp10,550
Rp1,500	Rp50,000	Rp10,550
Rp44,500	Rp61,000	Rp10,700
Rp91,500	Rp42,500	Rp10,650
Rp5,000	Rp116,000	Rp10,700
Rp152,500	Rp2,000	Rp11,150
Rp162,500	Rp72,500	Rp10,950
Rp34,000	Rp24,500	Rp10,750

4.2 Pemrosesan Data

Dari data inputan di atas akan di proses menggunakan algoritma backpropagation dengan tahapan sebagai berikut:

1. Input data.

Masukan data inputan dalam bentuk matrik. Karena terdapat 3 jenis data dengan panjang 14, maka matrik yang terbentuk adalah matrik 3 x 14 dan disimpan

dengan variabel p. Untuk lebih jelasnya dapat di lihat di bawah ini:

Data inputan
PT.Gudang Garam, Tbk

p= [86500 169500
0,2;
14000 32500
0,2;
11500 5000
0,2;
158000 3000
0,2;
50500 31500
0,2;
1500 10500
0,2;
500 49500
0,2;
1500 50000
0,2;
44500 61000
0,2;
91500 42500
0,2;
5000 11600
0,2;
152500 2000
0,2;

162500 72500
 0,2;
 34000 24500
 0,2];

Data inputan PT.
 Indosat, Tbk.

p= [180500 62500
 0,2;
 283000
 937500 0,2;
 1036000
 2107500 0,2;
 1884500
 782500 0,2;
 1756000
 3096500 0,2;
 3626500
 1008500 0,2;
 1735000
 327000 0,2;
 2834000
 361000 0,2;
 425500
 657000 0,2;
 1117000
 4108500 0,2;
 692000
 706500 0,2;
 285500
 1352000 0,2;

329500
 990000 0,2;
 376000
 349500 0,2
];

Data matrik di atas
 harus di transpose
 agar bisa di masukan
 kedalam jaringan
 pelatihan.

2. Normalisasi data.

Data input dan data
 target harus di
 normalisasi. Normalisasi
 harus berada pada range
 0 hingga 1. Data target
 di sini adalah data yang
 di harapkan muncul
 pada hasil prediksi,
 dengan kata lain ini
 merupakan data harga
 aktual. Harga aktual ini
 disimpan dengan
 variabel t. Untuk itu
 pada data inputan p dan
 data target t, semua data
 di bagi dengan
 1.000.000 untuk data
 pada PT. Gudang

Garam, Tbk dan
10.000.000 pada PT.
Indosat, Tbk.

3. Membangun jaringan backpropagation.

Jaringan ini terdiri
dari 2 layer, masing-
masing 10 neuron dan 5
neuron dengan 1 output.

```
net =  
newff(minmax(pn), [10 5  
1], {'tansig'  
'logsig'  
'purelin'}, 'traingdm');
```

4. Melihat bobot-bobot awal input, lapisan, dan bias.

```
BobotAwal_Input  
= net.IW{1,1};  
Bias_Awal_Input  
= net.b{1,1};  
BobotAwal_Lapisa  
n1 =  
net.LW{2,1};  
Bias_Awal_Lapisa  
n1 =  
net.b{2,1};
```

```
BobotAwal_Lapisa  
n2 =  
net.LW{3,2};  
Bias_Awal_Lapisa  
n2 =  
net.b{3,1};
```

5. set parameter

parameter yang
digunakan pada jaringan
pelatihan adalah sebagai
berikut:

- Batas epoch
maksimum
: 3000
- Batas maksimum
kesalahan
: 0,00001
- Learning rate
: 0,5
- Momentum
: 0,7
- Show
: 250

```
net.trainParam.e  
pochs =  
3000;
```

```
net.trainParam.goal
                    =
                    1e-5;
```

```
net.trainParam.lr
                    =
                    0.5;
```

```
net.trainParam.mc
                    =
                    0.7;
```

```
net.trainParam.show
                    =
                    250;
```

6. Pelatihan.

Pelatihan ini ditujukan untuk mendapatkan bobot baru, jaringan baru, output jaringan Y yang masih ternormalisasi, dan error.

```
[net, tr, Y, E] =
train(net, pn, tn);
```

Variabel net berisi nilai bobot setelah dilakukan pelatihan atau

disebut juga bobot akhir. Variabel net kemudian disimpan kedalam file bobot_back.mat.

```
nama_file =
'bobot_Back.mat';
save
(nama_file, 'net');
```

7. Melihat bobot-bobot

akhir input, lapisan, dan bias.

```
BobotAkhir_Input
= net.IW{1,1};
Bias_Akhir_Input
= net.b{1,1};
BobotAkhir_Lapis
an1 =
net.LW{2,1};
Bias_Akhir_Lapis
an1 =
net.b{2,1};
BobotAkhir_Lapis
an2 =
net.LW{3,2};
Bias_Akhir_Lapis
an2 =
net.b{3,1};
```

8. Mengembalikan range output Y dan t kedalam bentuk harga saham.
- Untuk data pada PT. Gudang Garam, Tbk.


```
Y = Y*100000;
Y=round(Y)
t=t;
```
 - Untuk data pada PT. Indosat, Tbk.


```
Y = Y*100000;
Y=round(Y);
t=t;
```

```
disp('target T:
'),t
disp('output
jaringan hasil
pelatihan: '), Y
```

9. MAPE(Mean Absolute Percentage Error).
- MAPE digunakan untuk mencari tingkat kesalahan pada hasil prediksi.

```
selisih =
abs(Y-t);
```

```
MAPE
=(sum(selisih/t)/length(t))*100;
disp(['MAPE:
',num2str(MAPE),'%'])
```

Setelah 9 langkah di atas selesai, klik tombol run atau tekan f5 untuk menjalankan pelatihan file.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil percobaan menggunakan metode Jaringan Syaraf Tiruan Backpropagation, didapatkan kesimpulan bahwa:

1. hasil yang di peroleh hampir sama dengan penggunaan metode Fuzzy Tsukamoto.
2. Penggunaan metode Jaringan Syaraf Tiruan Backpropagation lebih cepat karena tidak perlu menggunakan logika fuzzy yang begitu banyak.
3. Penggunaan Jaringan Syaraf Tiruan Backpropagation sangat sesuai untuk memprediksi harga saham

5.2 Saran

Untuk penelitian selanjutnya, sebaiknya penggunaan metode Jaringan Syaraf Tiruan Backpropagation dilakukan dengan menggunakan set parameter yang lebih variatif agar didapatkan hasil prediksi yang lebih akurat.