

# IMPLEMENTASI JARINGAN SYARAF TIRUAN PERAMBATAN BALIK UNTUK MEMPREDIKSI HARGA LOGAM MULIA EMAS MENGGUNAKAN ALGORITMA LEVENBERG MARQUARDT

Reza Najib Hidayat<sup>1</sup>, R. Rizal Isnanto<sup>2</sup>, Oky Dwi Nurhayati<sup>2</sup>

## Abstract

*Gold is one of commodities investment which its value continue to increase by year. The rising price of gold will encourage investors to choose to invest in gold rather than the stock market. With the risks that are relatively low, gold can give better results in accordance with its increasing price. In addition, gold can also be a safe value protector in the future. The Objectives of the research are to predict the price of gold using artificial neural networks backpropagations methods and to analyze best network used in prediction.*

*In the process of training data, it is used some training parameters to decide the best gold prediction architecture. Comparative parameters that is used are the variation of the number of hidden layers, number of neurons in each hidden layer, learning rate, minimum gradients and fault tolerance.*

*The results showed that the best architecture has an accuracy rate of 99,7604% of data training and test data at 98,849% with architecture combinations are have two hidden layer neurons combined 10-30, the error rate 0.00001 and 0.00001 of learning rate.*

**Index Terms :** Prediction, Gold, Artificial Neural Networks, Backpropagation.

## I. Pendahuluan

Kegiatan ekonomi di dunia ini tidak dapat dilepaskan dari kegiatan investasi karena dengan adanya investasi akan mendorong pertumbuhan ekonomi suatu negara. Logam mulia emas merupakan salah satu komoditas investasi yang nilainya terus mengalami kenaikan dari tahun ke tahun. Kenaikan harga emas ini akan mendorong investor untuk memilih berinvestasi emas daripada pasar modal. Investasi dalam emas memberikan hasil yang lebih baik untuk jangka panjang dan dengan daya beli (*purchasing power*) yang lebih baik, sehingga investasi emas merupakan solusi efektif mengingat nilai uang yang setiap tahun tergerus oleh inflasi. Inflasi merupakan suatu gejala yang terjadi karena adanya ketidakstabilan ekonomi sehingga harga mengalami kenaikan secara kontinu. Keadaan seperti ketidakstabilan ekonomi inilah yang mendorong banyak orang, organisasi dan perusahaan untuk berinvestasi logam mulia emas.

Berdasarkan permasalahan di atas, kemampuan untuk dapat memprediksi harga logam mulia emas merupakan keuntungan tersendiri untuk melindungi diri dari dampak inflasi, sehingga timbul ide untuk merancang sebuah aplikasi yang dapat digunakan untuk memprediksi harga logam mulia emas menggunakan metode Jaringan Syaraf Tiruan.

Tujuan dari penelitian ini diantaranya sebagai berikut.

1. Memprediksi harga logam mulia emas untuk satu hari kedepan.
2. Memberikan pilihan solusi untuk investor dalam melakukan investasi logam mulia emas.
3. Menganalisis tingkat akurasi prediksi harga logam mulia emas menggunakan JST perambatan balik dengan algoritma Levenberg Marquardt.

## II. Dasar Teori

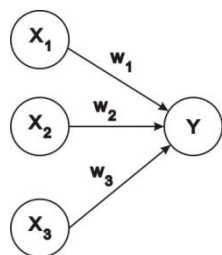
### 2.1 Jaringan Syaraf Tiruan (JST)

Jaringan syaraf tiruan (JST), adalah sistem komputasi di mana arsitektur dan operasi diilhami dari pengetahuan tentang sel syaraf biologis di dalam otak, yang merupakan salah satu representasi buatan dari otak manusia yang selalu mencoba menstimulasi proses pembelajaran pada otak manusia tersebut. JST dapat digambarkan sebagai model matematis dari model syaraf biologi. Model syaraf ditunjukkan dengan kemampuannya dalam emulasi, analisis,

---

1. Mahasiswa Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro. Jl. Prof. Soedarto Tembalang, Semarang-Indonesia. [rezanajib88@gmail.com](mailto:rezanajib88@gmail.com)  
2. Dosen Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro. Jl. Prof. Soedarto Tembalang, Semarang-Indonesia.

prediksi dan asosiasi. JST ditentukan oleh tiga hal, yaitu arsitektur jaringan, metode pelatihan dan fungsi aktivasi. Berikut merupakan contoh arsitektur jaringan sederhana yang ditunjukkan pada Gambar 1.



Gambar 1 Contoh Arsitektur Jaringan Sederhana

Keterangan :

Neuron masukan :  $X_1, X_2, X_3$

Bobot :  $W_1, W_2, W_3$

Neuron Keluaran :  $Y$

$Y$  menerima masukan dari neuron  $X_1, X_2, X_3$  dengan bobot hubungan masing-masing adalah  $W_1, W_2, W_3$ . Ketiga impuls neuron yang ada dijumlahkan ( $net = X_1W_1 + X_2W_2 + X_3W_3$ ). Besarnya impuls yang diterima oleh  $Y$  mengikuti fungsi aktivasi,  $y = f(net)$ . Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk mengubah bobot.

## 2.2 Jaringan Syaraf Tiruan Perambatan Balik

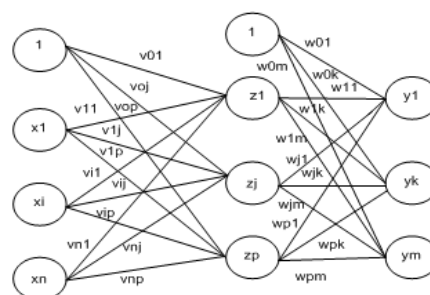
Perambatan balik merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perseptron dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Jaringan perambatan galat mundur merupakan salah satu algoritma yang sering digunakan dalam menyelesaikan masalah-masalah yang rumit. Hal ini dimungkinkan karena jaringan dengan algoritma ini dilatih dengan menggunakan metode belajar terbimbing. Pada jaringan diberikan sepasang pola yang terdiri atas pola masukan dan pola yang diinginkan. Ketika suatu pola diberikan kepada jaringan, bobot-bobot diubah untuk memperkecil perbedaan pola keluaran dan pola yang diinginkan. Latihan ini dilakukan berulang-ulang sehingga semua pola yang dikeluarkan jaringan dapat memenuhi pola yang diinginkan.

Algoritma pelatihan jaringan syaraf perambatan galat mundur terdiri atas dua langkah, yaitu perambatan maju dan perambatan mundur. Langkah perambatan maju dan perambatan mundur ini dilakukan pada jaringan untuk setiap pola yang diberikan selama jaringan mengalami pelatihan

## 2.3 Arsitektur Perambatan Balik

Perambatan balik memiliki beberapa unit yang ada dalam satu atau lebih lapisan tersembunyi. Pada

Gambar 2 menunjukkan arsitektur perambatan balik dengan  $n$  buah masukan (ditambah sebuah bias), sebuah lapisan tersembunyi yang terdiri dari  $p$  unit (ditambah sebuah bias), serta  $m$  unit keluaran.



Gambar 2 Arsitektur Jaringan Perambatan Balik

$V_{ji}$  merupakan bobot garis dari unit masukan  $X_i$  ke unit lapisan tersembunyi  $Z_j$  ( $V_{j0}$  merupakan bobot garis yang menghubungkan bias di unit masukan ke unit lapisan tersembunyi  $Z_j$ ).  $W_{kj}$  merupakan bobot dari unit lapisan tersembunyi  $Z_j$  ke unit keluaran  $Y_k$  ( $W_{k0}$  merupakan bobot dari bias di lapisan tersembunyi ke unit keluaran  $Z_k$ ).

## 2.4 Algoritma Pembelajaran Levenberg Marquardt

Algoritma levenberg marquardt merupakan pengembangan algoritma perambatan balik standar, pada algoritma standar proses perubahan bobot dan bias menggunakan negative gradient descent secara langsung sedangkan algoritma levenberg marquardt menggunakan pendekatan matrik hesian ( $H$ ) pada perubahan bobot dan biasnya yang dapat dihitung dengan

$$H = J^T e \dots\dots\dots (2.1)$$

Sedangkan gradient dapat dihitung dengan

$$g = J^T J \dots\dots\dots (2.2)$$

Dalam hal ini  $J$  merupakan matrik jacobian yang berisikan turunan pertama dari error jaringan terhadap bobot dan bias jaringan. Perubahan bobot dapat dihitung dengan

$$\Delta X = [J^T J + \mu I]^{-1} J^T e \dots\dots\dots (2.3)$$

Sehingga perbaikan bobot dapat ditentukan dengan

$$X = X + \Delta X \dots\dots\dots (2.4)$$

$$X = \Delta X + [J^T J + \mu I]^{-1} J^T e \dots\dots\dots (2.5)$$

$X$  = fungsi bobot-bobot jaringan dan bias

$$X = [V_{11}, V_{12}, \dots, V_{1j}; V_{11}, V_{01}, V_{02}, \dots, V_{0j}; W_{11}, W_{12}, \dots, W_{kj}; W_{01}, W_{02}, \dots, W_{0k}]$$

$e$  adalah vektor yang menyatakan semua error pada output jaringan

$$e = [t_1 - Y_1, t_2 - Y_2, \dots, t_p - Y_p]^T \dots (2.6)$$

$\mu$  = laju pembelajaran ;  $I$  = matrik identitas

## III. Metodologi Penelitian

Aplikasi jaringan syaraf tiruan untuk memprediksi harga logam mulia ini merupakan sistem yang dapat digunakan untuk meramalkan harga logam mulia untuk satu hari ke depan. Jaringan syaraf

digunakan untuk mengenali pola data pergerakan harga logam mulia.

Data diperoleh dari <http://antamgold.com>. Kemudian data dibagi menjadi 60% data latih dan 40% data uji. Data latih merupakan data yang digunakan dalam proses pelatihan dan data uji merupakan data yang digunakan dalam proses pengujian (tidak digunakan dalam proses pelatihan).

Setiap basis data kemudian diolah terlebih dahulu dengan melakukan proses normalisasi. Proses selanjutnya adalah melatih basis data tersebut ke dalam jaringan syaraf tiruan. Data masukan diumpangkan ke dalam jaringan untuk diolah, kemudian hasil pengolahan tersebut akan dibandingkan dengan target yang sebelumnya telah ditentukan. Proses pelatihan ini dibatasi oleh parameter-parameter pelatihan yang diinginkan. Setelah dilatih, sistem akan menyimpan jaringan tersebut sesuai dengan parameter pelatihan yang digunakan. Jaringan yang telah dilatih dan disimpan tersebut kemudian dapat diketahui kemampuannya dalam mengenali data latih. Kemampuan jaringan dalam mengenali data latih ini dapat menjadi acuan untuk kelayakan jaringan dalam mengenali data, terutama dalam proses peramalan.

Data diperoleh dari <http://antamgold.com>, merupakan harga logam mulia harian terhitung mulai tanggal 1 Januari 2012 sampai 31 Oktober 2012 (dalam rupiah). Data yang telah diperoleh kemudian dibagi menjadi dua yaitu sebagai data latih dan sebagai data uji. Pembagian data latih sebesar 60% dan data uji sebesar 40%, sehingga memiliki data uji sebanyak 178 pasang data dan data uji sebanyak 108 pasang data.

Pembuatan basis data ini berlaku untuk data latih dan data uji, namun data yang dipakai dalam proses pelatihan adalah data latih, sedangkan data uji digunakan sebagai masukan dan peramalan harga untuk kemudian hasil peramalan dicocokkan dengan target yang ada.

Data dibuat menjadi 10 masukan dan 1 buat target untuk satu data. Harga masukan merupakan harga 10 hari sebelum data yang akan diramal. Sedangkan target merupakan harga yang akan diramal pada 1 hari kedepan. Jika harga logam mulia dimulai dengan hari ke-1 sampai ke-10, maka target dan hari yang akan diramal adalah hari ke-11. Jika masukan mulai dengan harga hari ke-2 sampai hari ke-11, maka target dan hari yang akan diramal adalah hari ke-12, begitu seterusnya lalu disimpan dalam format \*.mat

Untuk merancang jaringan sebagai pola pelatihan, ditentukan terlebih dahulu parameter-parameter masukan sehingga peramalan dapat dilakukan.

Algoritma yang digunakan adalah perambatan balik dengan pembelajaran terbimbing sehingga membutuhkan data masukan dan data target. Berikut merupakan rancangan program yang terdapat dalam penelitian ini adalah

1. Lapisan Masukan : terdiri atas satu lapisan dengan 10 neuron yang merupakan harga dari logam mulia 10 hari sebelumnya.
2. Lapisan tersembunyi : terdiri atas dua pilihan lapisan tersembunyi yaitu satu lapisan tersembunyi dan dua lapisan tersembunyi dengan neuron yang bervariasi sesuai dengan masukan dari pengguna.
3. Lapisan keluaran : terdiri atas satu lapis dengan satu neuron yang merupakan target atau keluaran dari peramalan.

Selanjutnya kita lakukan pelatihan jaringan yang bertujuan untuk mendapatkan nilai bobot terbaik yang digunakan sebagai acuan bagi masukan data pada peramalan. Pola masukan dirambatkan melalui lapisan masukan sampai lapisan keluaran, lalu akan dicocokkan dengan keluaran target awal. Jika terdapat perbedaan nilai diantara keduanya, ini yang disebut dengan galat dan galat tersebut dapat kita atur nilainya, sehingga sebelum nilai galat sesuai dengan yang diinginkan maka pelatihan akan terus dilakukan.

Dalam pelatihan jaringan ini, terdapat beberapa parameter untuk mendapatkan hasil yang terbaik. Parameter-parameter tersebut, antara lain :

1. `net.trainParam.epochs`  
Merupakan parameter yang berfungsi untuk menentukan jumlah maksimum iterasi yang diinginkan. Besar nilai menentukan hasil yang dicapai. Jika nilai iterasi ditentukan sebesar 1000, maka pelatihan akan dihentikan ketika nilai iterasi telah mencapai 1000 atau nilai parameter lainnya telah tercapai.
2. `net.trainParam.goal`  
Merupakan parameter yang berfungsi untuk menentukan target galat pada pelatihan jaringan. Jika menentukan galat sebesar 0,00001, maka pelatihan akan berhenti ketika target galat telah mencapai 0,00001 atau nilai dari parameter lain telah tercapai.
3. `net.trainParam.lr`  
Merupakan parameter yang berfungsi untuk menentukan besar laju pembelajaran pada jaringan yang memiliki besaran nilai antara 0-1.
4. `net.trainParam.min_grad`  
Merupakan parameter yang menentukan batas gradient minimum yang dicapai pada saat pelatihan jaringan.

Parameter-parameter tersebut saling berkaitan satu sama lain sehingga yang pelatihan akan dihentikan jika salah satu nilai telah terpenuhi, dan pelatihan di sini menggunakan cara *trial and error* sehingga kita harus mengkombinasikan parameter-parameter tersebut untuk mencapai nilai target.

## IV. Hasil Penelitian dan Pembahasan

### 4.1 Pengujian Data Latih

Dalam pengujian data latih digunakan beberapa kombinasi antara lain :

1. Menggunakan nilai parameter *default* yaitu dengan nilai *target/toleransi* kesalahan sebesar 0, laju pembelajaran bernilai 0,01 dan gradien minimum sebesar  $1,00 \times 10^{-10}$ .
2. Dari nilai *default*, menambahkan nilai toleransi menjadi 0,0001, memperbesar nilai laju pembelajaran menjadi 0,3.
3. Lalu merubah nilai toleransi menjadi 0,00001 untuk lebih memperkecil galat yang akan memberikan hasil tingkat akurasi yang lebih tinggi, menggunakan variasi nilai laju pembelajaran sebesar 0,001 ; 0,0001 dan 0,00001 untuk mengetahui apakah nilai laju pembelajaran mempengaruhi hasil tingkat akurasi dan merubah nilai gradien minimum menjadi sebesar  $1,00 \times 10^{-7}$ .

Untuk semua nilai parameter tersebut menggunakan variasi jumlah lapisan tersembunyi dan kombinasi jumlah neuron untuk mengetahui arsitektur seperti apakah yang paling baik untuk pelatihan data tersebut.

#### 4.2 Pengujian Data Latih Satu Lapisan Tersembunyi

Pengujian data latih satu lapisan tersembunyi merupakan pengujian yang dilakukan untuk melihat hasil akurasi yang dihasilkan setelah data melalui proses pelatihan. Tabel 1 berikut merupakan hasil pengujian data latih dengan satu lapisan tersembunyi.

Tabel 1. Hasil Pengujian Data Latih Satu Lapisan Tersembunyi

No.	Parameter Pelatihan				Hasil Akurasi (%)
	Neuron	Galat	Laju Pembelajaran	Gradien Minimum	
1	10	0	0,01	$1,00 \times 10^{-10}$	99,900
2	10	0	0,3	$1,00 \times 10^{-10}$	99,899
3	10	0,0001	0,01	$1,00 \times 10^{-10}$	99,404
4	10	0,0001	0,3	$1,00 \times 10^{-10}$	99,330
5	10	0,00001	0,00001	$1,00 \times 10^{-7}$	99,771
6	10	0,00001	0,0001	$1,00 \times 10^{-7}$	99,756
7	10	0,00001	0,001	$1,00 \times 10^{-7}$	99,766
8	20	0	0,01	$1,00 \times 10^{-10}$	100
9	20	0	0,3	$1,00 \times 10^{-10}$	100
10	20	0,0001	0,01	$1,00 \times 10^{-10}$	99,326
11	20	0,0001	0,3	$1,00 \times 10^{-10}$	99,361
12	30	0,00001	0,00001	$1,00 \times 10^{-7}$	99,757
13	30	0,00001	0,0001	$1,00 \times 10^{-7}$	99,765
14	30	0,00001	0,001	$1,00 \times 10^{-7}$	99,762
15	50	0,00001	0,00001	$1,00 \times 10^{-7}$	99,755
16	50	0,00001	0,0001	$1,00 \times 10^{-7}$	99,763
17	50	0,00001	0,001	$1,00 \times 10^{-7}$	99,751
18	70	0,00001	0,00001	$1,00 \times 10^{-7}$	99,756
19	70	0,00001	0,0001	$1,00 \times 10^{-7}$	99,763
20	70	0,00001	0,001	$1,00 \times 10^{-7}$	99,763
21	90	0,00001	0,00001	$1,00 \times 10^{-7}$	99,763
22	90	0,00001	0,0001	$1,00 \times 10^{-7}$	99,767
23	90	0,00001	0,001	$1,00 \times 10^{-7}$	99,777

Dari Tabel 1 bisa kita lihat bahwa hasil tingkat akurasi terbaik mencapai 100% yaitu pada kombinasi satu lapisan tersembunyi dengan jumlah neuron 20, galat 0 dan laju pembelajaran masing-masing bernilai 0,01 dan 0,3. Sedangkan hasil tingkat akurasi terendah sebesar 99,241% dengan kombinasi jumlah neuron 20, galat 0,0001 dan laju pembelajaran bernilai 0,3. Pada kasus tersebut, hasil akurasi terbaik maupun hasil akurasi terendah memiliki jumlah neuron yang sama yaitu berjumlah 20, hal tersebut menunjukkan bahwa jumlah neuron tidak mempengaruhi tingkat akurasi data.

Pada hasil akurasi terbaik, memiliki galat 0 dan pada hasil akurasi terendah memiliki galat bernilai 0,0001, hal tersebut menunjukkan bahwa nilai parameter galat mempengaruhi hasil tingkat akurasi data. Lalu untuk nilai parameter laju pembelajaran, pada hasil akurasi terbaik memiliki nilai sebesar 0,01 dan 0,3, begitu juga dengan nilai parameter pada hasil akurasi terendah yang memiliki nilai laju pembelajaran sebesar 0,3, hal tersebut menunjukkan bahwa laju pembelajaran tidak mempengaruhi tingkat akurasi data. Untuk parameter nilai gradien minimum, kedua hasil tersebut memiliki nilai yang sama yaitu sebesar  $1,00 \times 10^{-10}$  yang berarti nilai gradien minimum tidak mempengaruhi hasil tingkat akurasi data pelatihan

#### 4.3 Pengujian Data Latih Dua Lapisan Tersembunyi

Pengujian data latih dua lapisan tersembunyi merupakan pengujian untuk melihat hasil akurasi data latih. Jika sebelumnya memiliki satu lapisan tersembunyi, kali ini jaringan memiliki dua lapisan tersembunyi. Hal ini dilakukan untuk mengetahui apakah penambahan lapisan tersembunyi berpengaruh terhadap akurasi data atau tidak yang ditunjukkan pada Tabel 2 berikut.

Tabel 2 Hasil Pengujian Data Latih Dua Lapisan Tersembunyi

No.	Parameter Pelatihan					Hasil Akurasi (%)
	Lapisan Tersembunyi		Galat	Laju Pembelajaran	Gradien Minimum	
	I	II				
1	10	10	0	0,01	$1,00 \times 10^{-10}$	100
2	10	10	0	0,3	$1,00 \times 10^{-10}$	100
3	10	10	0,0001	0,01	$1,00 \times 10^{-10}$	99,179
4	10	10	0,0001	0,3	$1,00 \times 10^{-10}$	99,512
5	10	10	0,00001	0,00001	$1,00 \times 10^{-7}$	99,784
6	10	10	0,00001	0,0001	$1,00 \times 10^{-7}$	99,779
7	10	10	0,00001	0,001	$1,00 \times 10^{-7}$	99,780
8	10	20	0	0,01	$1,00 \times 10^{-10}$	100
9	10	20	0	0,3	$1,00 \times 10^{-10}$	100
10	10	20	0,0001	0,01	$1,00 \times 10^{-10}$	99,284
11	10	20	0,0001	0,3	$1,00 \times 10^{-10}$	99,248
12	10	30	0,00001	0,00001	$1,00 \times 10^{-7}$	99,760
13	10	30	0,00001	0,0001	$1,00 \times 10^{-7}$	99,762
14	10	30	0,00001	0,001	$1,00 \times 10^{-7}$	99,766
15	10	50	0,00001	0,00001	$1,00 \times 10^{-7}$	99,771

Tabel 2 Hasil Pengujian Data Latih Dua Lapisan Tersembunyi (Lanjutan)

No.	Parameter Pelatihan				Hasil Akurasi (%)	
	Lapisan Tersembunyi		Galat	Laju Pembelajaran		Gradien Minimum
	I	II				
16	10	50	0,00001	0,0001	$1,00 \times 10^{-7}$	99,770
17	10	50	0,00001	0,001	$1,00 \times 10^{-7}$	99,771
18	20	10	0	0,01	$1,00 \times 10^{-10}$	100
19	20	10	0	0,3	$1,00 \times 10^{-10}$	100
20	20	10	0,0001	0,01	$1,00 \times 10^{-10}$	99,210
21	20	10	0,0001	0,3	$1,00 \times 10^{-10}$	99,335
22	20	20	0	0,01	$1,00 \times 10^{-10}$	100
23	20	20	0	0,3	$1,00 \times 10^{-10}$	100
24	20	20	0,0001	0,01	$1,00 \times 10^{-10}$	99,399
25	20	20	0,0001	0,3	$1,00 \times 10^{-10}$	99,207
26	30	10	0,00001	0,00001	$1,00 \times 10^{-7}$	99,763
27	30	10	0,00001	0,0001	$1,00 \times 10^{-7}$	99,763
28	30	10	0,00001	0,001	$1,00 \times 10^{-7}$	99,762
29	30	30	0,00001	0,00001	$1,00 \times 10^{-7}$	99,772
30	30	30	0,00001	0,0001	$1,00 \times 10^{-7}$	99,767
31	30	30	0,00001	0,001	$1,00 \times 10^{-7}$	99,774
32	30	50	0,00001	0,00001	$1,00 \times 10^{-7}$	99,773
33	30	50	0,00001	0,0001	$1,00 \times 10^{-7}$	99,774
34	30	50	0,00001	0,001	$1,00 \times 10^{-7}$	99,785
35	50	10	0,00001	0,00001	$1,00 \times 10^{-7}$	99,767
36	50	10	0,00001	0,0001	$1,00 \times 10^{-7}$	99,760
37	50	10	0,00001	0,001	$1,00 \times 10^{-7}$	99,767
38	50	30	0,00001	0,00001	$1,00 \times 10^{-7}$	99,758
39	50	30	0,00001	0,0001	$1,00 \times 10^{-7}$	99,759
40	50	30	0,00001	0,001	$1,00 \times 10^{-7}$	99,776
41	50	50	0,00001	0,00001	$1,00 \times 10^{-7}$	99,776
42	50	50	0,00001	0,0001	$1,00 \times 10^{-7}$	99,845
43	50	50	0,00001	0,001	$1,00 \times 10^{-7}$	99,768
44	70	10	0,00001	0,00001	$1,00 \times 10^{-7}$	99,764
45	70	10	0,00001	0,0001	$1,00 \times 10^{-7}$	99,765
46	70	10	0,00001	0,001	$1,00 \times 10^{-7}$	99,770
47	70	30	0,00001	0,00001	$1,00 \times 10^{-7}$	99,771
48	70	30	0,00001	0,0001	$1,00 \times 10^{-7}$	99,780
49	70	30	0,00001	0,001	$1,00 \times 10^{-7}$	99,781
50	70	50	0,00001	0,00001	$1,00 \times 10^{-7}$	99,786
51	70	50	0,00001	0,0001	$1,00 \times 10^{-7}$	99,768
52	70	50	0,00001	0,001	$1,00 \times 10^{-7}$	99,766
53	90	10	0,00001	0,00001	$1,00 \times 10^{-7}$	99,789
54	90	10	0,00001	0,0001	$1,00 \times 10^{-7}$	99,775
55	90	10	0,00001	0,001	$1,00 \times 10^{-7}$	99,771
56	90	30	0,00001	0,00001	$1,00 \times 10^{-7}$	99,783
57	90	30	0,00001	0,0001	$1,00 \times 10^{-7}$	99,795
58	90	30	0,00001	0,001	$1,00 \times 10^{-7}$	99,762
59	90	50	0,00001	0,00001	$1,00 \times 10^{-7}$	99,766
60	90	50	0,00001	0,0001	$1,00 \times 10^{-7}$	99,783
61	90	50	0,00001	0,001	$1,00 \times 10^{-7}$	99,763

Pada satu lapisan tersembunyi, hasil akurasi terendah mencapai 99,241% sedangkan pada jaringan dua lapisan tersembunyi mencapai 99,179%. Hal tersebut menunjukkan bahwa penambahan jumlah lapisan tersembunyi tidak berbanding lurus terhadap peningkatan akurasi data.

#### 4.4 Pengujian Data Uji Satu Lapisan Tersembunyi

Pengujian data uji dilakukan untuk mengetahui kemampuan jaringan dalam memprediksi harga logam mulia yang sebenarnya dengan menggunakan data baru yang disebut sebagai data uji. Tabel 3

menunjukkan hasil pengujian data uji dengan satu lapis tersembunyi.

Tabel 3 Hasil Pengujian Data Uji Satu Lapis Tersembunyi

No.	Parameter Pelatihan				Hasil Akurasi (%)
	Neuron	Galat	Laju Pembelajaran	Gradien Minimum	
1	10	0	0,01	$1,00 \times 10^{-10}$	53,886
2	10	0	0,3	$1,00 \times 10^{-10}$	82,736
3	10	0,0001	0,01	$1,00 \times 10^{-10}$	97,533
4	10	0,0001	0,3	$1,00 \times 10^{-10}$	95,817
5	10	0,00001	0,00001	$1,00 \times 10^{-7}$	92,849
6	10	0,00001	0,0001	$1,00 \times 10^{-7}$	93,931
7	10	0,00001	0,001	$1,00 \times 10^{-7}$	94,787
8	20	0	0,01	$1,00 \times 10^{-10}$	59,892
9	20	0	0,3	$1,00 \times 10^{-10}$	84,082
10	20	0,0001	0,01	$1,00 \times 10^{-10}$	98,096
11	20	0,0001	0,3	$1,00 \times 10^{-10}$	96,434
12	30	0,00001	0,00001	$1,00 \times 10^{-7}$	92,857
13	30	0,00001	0,0001	$1,00 \times 10^{-7}$	93,229
14	30	0,00001	0,001	$1,00 \times 10^{-7}$	92,270
15	50	0,00001	0,00001	$1,00 \times 10^{-7}$	94,467
16	50	0,00001	0,0001	$1,00 \times 10^{-7}$	93,159
17	50	0,00001	0,001	$1,00 \times 10^{-7}$	92,880
18	70	0,00001	0,00001	$1,00 \times 10^{-7}$	96,277
19	70	0,00001	0,0001	$1,00 \times 10^{-7}$	96,380
20	70	0,00001	0,001	$1,00 \times 10^{-7}$	97,355
21	90	0,00001	0,00001	$1,00 \times 10^{-7}$	95,502
22	90	0,00001	0,0001	$1,00 \times 10^{-7}$	98,826
23	90	0,00001	0,001	$1,00 \times 10^{-7}$	91,614

Pada pengujian data uji, hasil tingkat akurasi tidak sebesar hasil uji pada data latih, hal ini wajar terjadi karena data latih merupakan data yang telah melalui proses pelatihan dan pembobotan yang dibimbing untuk mendekati nilai target, sedangkan data uji tidak melalui proses pelatihan. Tabel 3 menunjukkan bahwa hasil tingkat akurasi terbaik dihasilkan pada kombinasi nilai parameter dengan jumlah neuron sebanyak 90, galat 0,00001, laju pembelajaran 0,0001 dengan gradien minimum  $1,00 \times 10^{-7}$  yaitu sebesar 98,826%. Sedangkan hasil akurasi terendah dihasilkan pada kombinasi nilai parameter dengan jumlah neuron 10, galat 0, laju pembelajaran 0,01 dengan gradien minimum  $1,00 \times 10^{-10}$  yaitu sebesar 53,886%. Beberapa faktor tentu sangat mempengaruhi hasil akurasi data, jaringan dengan nilai parameter galat 0 memerlukan iterasi yang lebih banyak dibandingkan dengan galat bernilai 0,0001. Hal tersebut dikarenakan jaringan akan terus dilatih sampai target benar-benar ditemukan dan memiliki galat 0.

#### 4.5 Pengujian Data Uji Dua Lapisan Tersembunyi

Pengujian data uji dua lapisan tersembunyi dilakukan untuk mengetahui apakah penambahan lapisan tersembunyi mempengaruhi tingkat akurasi prediksi untuk data uji atau tidak. Pada Tabel 4 ditunjukkan hasilnya sebagai berikut.

Tabel 4 Hasil Pengujian Data Uji Dua Lapisan Tersembunyi

No.	Parameter Pelatihan					Hasil Akurasi (%)
	Lapisan Tersembunyi		Galat	Laju Pembelajaran	Gradien Minimum	
	I	II				
1	10	10	0	0,01	$1,00 \times 10^{-20}$	60,873
2	10	10	0	0,3	$1,00 \times 10^{-20}$	80,100
3	10	10	0,0001	0,01	$1,00 \times 10^{-20}$	93,283
4	10	10	0,0001	0,3	$1,00 \times 10^{-20}$	97,863
5	10	10	0,00001	0,00001	$1,00 \times 10^{-7}$	96,147
6	10	10	0,00001	0,0001	$1,00 \times 10^{-7}$	92,774
7	10	10	0,00001	0,001	$1,00 \times 10^{-7}$	97,120
8	10	20	0	0,01	$1,00 \times 10^{-20}$	91,966
9	10	20	0	0,3	$1,00 \times 10^{-20}$	81,824
10	10	20	0,0001	0,01	$1,00 \times 10^{-20}$	96,598
11	10	20	0,0001	0,3	$1,00 \times 10^{-20}$	97,385
12	10	30	0,00001	0,00001	$1,00 \times 10^{-7}$	98,849
13	10	30	0,00001	0,0001	$1,00 \times 10^{-7}$	91,571
14	10	30	0,00001	0,001	$1,00 \times 10^{-7}$	97,760
15	10	50	0,00001	0,00001	$1,00 \times 10^{-7}$	97,661
16	10	50	0,00001	0,0001	$1,00 \times 10^{-7}$	96,692
17	10	50	0,00001	0,001	$1,00 \times 10^{-7}$	93,449
18	20	10	0	0,01	$1,00 \times 10^{-20}$	79,635
19	20	10	0	0,3	$1,00 \times 10^{-20}$	89,259
20	20	10	0,0001	0,01	$1,00 \times 10^{-20}$	97,567
21	20	10	0,0001	0,3	$1,00 \times 10^{-20}$	96,963
22	20	20	0	0,01	$1,00 \times 10^{-20}$	67,037
23	20	20	0	0,3	$1,00 \times 10^{-20}$	93,483
24	20	20	0,0001	0,01	$1,00 \times 10^{-20}$	98,605
25	20	20	0,0001	0,3	$1,00 \times 10^{-20}$	94,236
26	30	10	0,00001	0,00001	$1,00 \times 10^{-7}$	98,583
27	30	10	0,00001	0,0001	$1,00 \times 10^{-7}$	96,031
28	30	10	0,00001	0,001	$1,00 \times 10^{-7}$	94,851
29	30	30	0,00001	0,00001	$1,00 \times 10^{-7}$	95,282
30	30	30	0,00001	0,0001	$1,00 \times 10^{-7}$	97,392
31	30	30	0,00001	0,001	$1,00 \times 10^{-7}$	97,645
32	30	50	0,00001	0,00001	$1,00 \times 10^{-7}$	97,697
33	30	50	0,00001	0,0001	$1,00 \times 10^{-7}$	96,425
34	30	50	0,00001	0,001	$1,00 \times 10^{-7}$	98,448
35	50	10	0,00001	0,00001	$1,00 \times 10^{-7}$	93,405
36	50	10	0,00001	0,0001	$1,00 \times 10^{-7}$	93,153
37	50	10	0,00001	0,001	$1,00 \times 10^{-7}$	91,350
38	50	30	0,00001	0,00001	$1,00 \times 10^{-7}$	96,730
39	50	30	0,00001	0,0001	$1,00 \times 10^{-7}$	97,560
40	50	30	0,00001	0,001	$1,00 \times 10^{-7}$	96,595
41	50	50	0,00001	0,00001	$1,00 \times 10^{-7}$	95,947
42	50	50	0,00001	0,0001	$1,00 \times 10^{-7}$	94,055
43	50	50	0,00001	0,001	$1,00 \times 10^{-7}$	96,249
44	70	10	0,00001	0,00001	$1,00 \times 10^{-7}$	97,415
45	70	10	0,00001	0,0001	$1,00 \times 10^{-7}$	98,798
46	70	10	0,00001	0,001	$1,00 \times 10^{-7}$	94,222
47	70	30	0,00001	0,00001	$1,00 \times 10^{-7}$	96,745
48	70	30	0,00001	0,0001	$1,00 \times 10^{-7}$	97,313
49	70	30	0,00001	0,001	$1,00 \times 10^{-7}$	96,833
50	70	50	0,00001	0,00001	$1,00 \times 10^{-7}$	96,839
51	70	50	0,00001	0,0001	$1,00 \times 10^{-7}$	97,514
52	70	50	0,00001	0,001	$1,00 \times 10^{-7}$	95,061
53	90	10	0,00001	0,00001	$1,00 \times 10^{-7}$	95,593
54	90	10	0,00001	0,0001	$1,00 \times 10^{-7}$	96,255
55	90	10	0,00001	0,001	$1,00 \times 10^{-7}$	97,689
56	90	30	0,00001	0,00001	$1,00 \times 10^{-7}$	95,601
57	90	30	0,00001	0,0001	$1,00 \times 10^{-7}$	97,517
58	90	30	0,00001	0,001	$1,00 \times 10^{-7}$	97,941
59	90	50	0,00001	0,00001	$1,00 \times 10^{-7}$	96,597
60	90	50	0,00001	0,0001	$1,00 \times 10^{-7}$	94,176
61	90	50	0,00001	0,001	$1,00 \times 10^{-7}$	95,518

Tabel 4 menunjukkan bahwa hasil tingkat akurasi terbaik untuk data uji adalah pada kombinasi neuron 10-30 dengan galat 0,00001 dan laju pembelajaran sebesar 0,00001 yaitu dengan hasil akurasi sebesar 98,849%. Sedangkan untuk hasil tingkat akurasi terendah terdapat pada kombinasi neuron 10-10 dengan galat 0 dan laju pembelajaran sebesar 0,01 yaitu dengan hasil akurasi sebesar 60,873%.

Dari percobaan yang telah dilakukan, kita dapat menentukan arsitektur yang terbaik untuk melakukan prediksi harga logam mulia emas yaitu arsitektur dengan rata-rata antara hasil akurasi data latih dan data uji yang paling tinggi.

Dengan demikian, jaringan yang memiliki hasil rata-rata akurasi terbaik untuk data latih dan data uji adalah jaringan dengan kombinasi nilai parameter dengan dua lapisan tersembunyi, neuron 10-30, galat sebesar 0,00001, laju pembelajaran sebesar 0,00001 dan nilai gradien minimum sebesar  $1,00 \times 10^{-7}$  dengan akurasi data latih 99,760% dan akurasi data uji 98,849%.

## V. Kesimpulan

Kesimpulan yang dapat diambil dari hasil pengujian dan pembahasan dari tiap-tiap jaringan adalah sebagai berikut.

1. Jaringan syaraf tiruan melakukan proses peramalan melalui pelatihan dengan nilai-nilai parameter pelatihan jaringan dan mengenali pola dari pergerakan data masa lalu.
2. Parameter-parameter pada JST saling mempengaruhi hasil tingkat akurasi data, kombinasi yang seimbang akan menghasilkan akurasi yang baik.
3. Nilai parameter galat berpengaruh terhadap hasil akurasi data uji karena jika pada saat pelatihan jaringan nilai parameter galat tidak terpenuhi maka hasil pengujian terhadap data uji akan menghasilkan hasil akurasi yang rendah.
4. Arsitektur JST yang paling tepat untuk prediksi harga logam mulia adalah arsitektur dengan hasil akurasi antara data latih dan data uji yang memiliki selisih paling rendah yaitu dengan hasil akurasi data latih sebesar 99,760% dan data uji sebesar 98,849% yang dimiliki oleh jaringan dengan arsitektur dua lapisan tersembunyi kombinasi neuron 10-30, galat sebesar 0,00001 dan laju pembelajaran sebesar 0,00001.
5. Aplikasi jaringan syaraf tiruan perambatan balik untuk memprediksi harga logam mulia emas ini dapat digunakan secara *real time*.

Dari kesimpulan tersebut, penulis memberikan saran sebagai berikut.

1. Perlu dilakukan penelitian lanjutan dengan kombinasi parameter yang lebih baik

sehingga akan menghasilkan akurasi data latih dan data uji yang lebih tinggi.

2. Perlu dilakukan penelitian lanjutan dengan jumlah data yang lebih banyak sehingga lonjakan harga akan lebih bervariasi, misalkan data lima tahunan.

#### Daftar Pustaka

- [1] Sunariyah, "Pengantar Pengetahuan Pasar Modal", UPP APM YKPN, Yogyakarta, 2003.
- [2] Roy Sembel, "Investasi Emas", Kontan, Maret 2008, hal 12.
- [3] Pangestuti, Sri, "*Analisis Return LQ45 Dibandingkan Return Emas dan faktor-Faktor Yang mempengaruhi Return LQ45 dan Return Emas Selama Periode 1995 – 2010*", Thesis-S2, 2010.
- [4] Rahardja, Prathama, "*Uang dan Perbankan*", Rineka Cipta, Jakarta, 1997.
- [5] Baswara, Sastya Yoga, "Analisis Nilai Hasil Investasi Deposito Rupiah, Deposito Dolar Amerika dan Dinar Emas dengan Emas sebagai Alat Ukur", Skripsi-S1, Universitas Diponegoro, Semarang, 2012.
- [6] Hermawan, Arief "Jaringan Syaraf Tiruan-Teori dan Aplikasi", Andi, Yogyakarta, 2006.
- [7] Siang, J. J., "Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan MATLAB", Penerbit Andi, Yogyakarta, 2009.
- [8] Kusumadewi, S. dan S. Hartati, "NEURO-FUZZY Integrasi Sistem Fuzzy dan Jaringan Saraf edisi 2", Graha Ilmu, Yogyakarta, 2010.
- [9] Fausett, Laurene V, "Fundamentals of Neural Networks: Architectures, Algorithms, and Applications", J. N., 1994.
- [10] Rahmat, dkk, "Perbandingan Algoritma Levenberg Marquardt dengan Metoda Backpropagation pada Proses Learning Jaringan Saraf Tiruan untuk Pengenalan Pola Sinyal Elektrokardiograf", Seminar Nasional Aplikasi Teknologi Informasi 2006 (SNATI 2006), Yogyakarta, 17 Juni 2006.
- [11] -- Levenberg Marquardt Backpropagation. <http://www.mathworks.com/help/nnet/ref/trainlm.html>.
- [12] Nugroho, Gathut, "Aplikasi Jaringan Syaraf Tiruan dengan Metode Perambatan Balik untuk Peramalan Harga Dinar dan Dirham", Skripsi-S1, Universitas Diponegoro, Semarang, 2012.

#### Biografi Penulis



Reza Najib Hidayat lahir di Temanggung, 25 Oktober 1989. Memulai pendidikan di SD N Bulu Lor 04, SMP N 7 Semarang, SMA N 6 Semarang. Saat ini penulis sedang menempuh pendidikan S1 di program studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro angkatan 2008.