

## **STATISTICAL TECHNIQUE DAN PARAMETER OPTIMIZATION PADA NEURAL NETWORK UNTUK FORECASTING HARGA EMAS**

Harminto Mulyo  
Fakultas Sains dan Teknologi, UNISNU Jepara  
minto@unisnu.ac.id

### **ABSTRACT**

*Gold is a precious metal that is valuable in the world that is soft, corrosion-resistant, malleable. The investment experts often advise to invest in gold because gold is a classic hedge against inflation and adds value in conditions of instability of currency exchange rate fluctuations. Gold price history data from year to year tends to rise, it's interesting researchers to examine the data using a variety of forecasting methods, including Statistical Technique and Data Mining Technique. Experiments were performed to look for the smallest error value by using statistical techniques and data mining. From the test results, the best statistical technique that uses a technique Single Moving Average with MSE of 760.55. In the data mining technique using Neural Network Backprogration obtained RMSE of +/- 22 730 6945.*

**Keywords:** *Parameter, Optimization, gold, Neural Network*

### **ABSTRAK**

Emas merupakan salah satu logam mulia yang bernilai di dunia yang bersifat lunak, tahan korosi, mudah ditempa. Para pakar investasi seringkali menganjurkan untuk berinvestasi pada emas karena emas merupakan sarana lindung nilai klasik untuk melawan inflasi dan menambah nilai dalam kondisi ketidakstabilan fluktuasi nilai mata uang. Data riwayat harga emas dari tahun ke tahun cenderung naik, hal tersebut menarik peneliti untuk menguji data menggunakan berbagai metode peramalan, diantaranya *Statistical Technique* dan *Data Mining Technique*. Eksperimen dilakukan untuk mencari nilai *error* terkecil dengan menggunakan teknik statistik dan data mining. Dari hasil pengujian, teknik statistik terbaik yaitu menggunakan teknik *Single Moving Average* dengan *MSE* sebesar 760.55. Pada teknik data mining dengan menggunakan metode *Neural Network Backprogration* didapat *RMSE* sebesar 22.730 +/- 6.945.

**Kata Kunci:** *Parameter, Optimization, emas, Neural Network*

### **Pendahuluan**

Investasi pada hakikatnya merupakan komitmen terhadap sejumlah sumber daya pada saat ini dengan tujuan untuk mendapatkan keuntungan di masa depan (Halim 2005). Umumnya investasi dibedakan menjadi dua, yaitu investasi pada aset-aset finansial (*financial assets*) dan investasi pada aset-aset riil (*real assets*). Investasi pada aset-aset finansial dilakukan di pasar uang, misalnya berupa sertifikat deposito, commercial paper, surat berharga pasar uang, dan lain-lain. Investasi juga dapat dilakukan di

pasar modal, misalnya berupa saham, obligasi, waran, opsi, dan lain-lain. Sedangkan investasi pada aset-aset riil dapat berbentuk pembelian aset produktif, pendirian pabrik, pembukaan pertambangan, pembukaan perkebunan dan lainnya.

Emas merupakan salah satu logam mulia yang bernilai di dunia yang bersifat lunak, tahan korosi, mudah ditempa. Para pakar investasi seringkali menganjurkan untuk berinvestasi pada emas karena emas merupakan sarana lindung nilai klasik untuk melawan inflasi dan menambah nilai dalam

kondisi ketidakstabilan fluktuasi nilai mata uang (Aprianti 2012). Investasi dalam emas dibedakan menjadi dua jenis yaitu, investasi pada saham emas dan investasi pada emas batangan. Harga saham open emas dari tahun 2000 sampai dengan 2013 nyaris tidak pernah turun tajam. Data yang diperoleh dari data.okfn.org yaitu pada bulan Januari 2000 harga emas US\$ 284.59 per troy ounce dan pada Desember 2013 harga emas mencapai US\$ 1221.588 per troy ounce. Begitu juga dengan harga emas batangan yang menyesuaikan harga saham emas. Harga emas yang nyaris tidak pernah turun tersebut melatarbelakangi minat beli investor terhadap emas.

Selain itu, faktor yang melatarbelakangi minat beli investor terhadap emas adalah pembelian emas yang melonjak tajam oleh Negara China dan India. Ancaman inflasi masa depan akibat kebijakan cetak uang oleh bank sentral negara-negara maju (kebijakan ini berakibat pada kehancuran nilai tukar mata uang global) termasuk salah satu faktor yang mempengaruhi kenaikan atau penurunan harga emas. Faktor yang mempengaruhi kenaikan atau penurunan harga emas yang lain adalah (1) krisis finansial, (2) naiknya permintaan emas di pasaran, (3) kurs dollar, (4) harga minyak, dan (6) situasi politik dunia.

Agar tujuan investasi tercapai, maka sebelum memasuki dunia investasi diperlukan pengetahuan keuntungan dan risiko yang didapat ketika terjun di bidang investasi. Harapan keuntungan dalam dunia investasi sering juga disebut sebagai *return*. Risiko investasi bisa diartikan sebagai kemungkinan terjadinya perbedaan antara *return* aktual dengan *return* yang diharapkan. Dua konsep ini, risiko maupun *return*, bagaikan dua sisi mata uang yang se lalu berdampingan. Artinya, dalam berinvestasi di samping menghitung *return* yang diharapkan investor juga harus memperhatikan risiko yang ditanggung (Halim 2005). Pengetahuan ini

penting sebagai pegangan ketika memasuki dunia investasi yang penuh risiko dan ketidakpastian. Salah satu pengetahuan penting dalam berinvestasi emas adalah peramalan harganya. Peramalan harga emas diperlukan bagi investor untuk mengetahui kecenderungan harga emas di masa datang. Peramalan adalah proses perkiraan (pengukuran) besarnya atau jumlah sesuatu pada waktu yang akan datang berdasarkan data pada masa lampau (*time series*) yang dianalisis secara ilmiah khususnya menggunakan metode statistika (Sudjana, 2005).

Peramalan harga emas bertujuan untuk mengetahui peluang investasi harga emas di masa yang akan datang sehingga dapat digunakan sebagai pertimbangan oleh investor emas untuk mengetahui perubahan harga emas. Metode peramalan sangat banyak dan seringkali memerlukan asumsi asumsi yang harus dipenuhi, namun terdapat juga model yang tidak memerlukan asumsi-asumsi salah satunya *Neural Network* (NN). Model NN dibedakan menjadi 2, yaitu *Feed Forward Neural Network* (FFNN), dimana proses pelatihan berjalan maju dari lapisan input menuju lapisan output selanjutnya, *Recurrent Neural Network* (RNN) yang proses pembelajarannya paling sedikit ada satu koneksi umpan balik supaya terjadi proses siklis (Computing 2010). Yang termasuk dalam kelas FFNN adalah *Backpropagation Neural Network*, *Radial Basis Function Network*, *General Regression Neural Network*. Algoritma *Backpropagation* dalam FFNN merupakan model yang sederhana jika digunakan untuk menyelesaikan masalah data *time series*. Salah satu data *timeseries* adalah harga emas.

Penggunaan FFNN untuk analisis data *time series* secara luas telah banyak dilakukan, antara lain Edy Supriyanto (2004) menerapkan jaringan syaraf tiruan untuk memprediksi harga saham (Supriyanto 2004),

Oman Somantri (2014) menggunakan NN dengan Bootstrap Aggregating (*Bagging*) untuk Penentuan Prediksi Harga Listrik (Somantri et al. 2014). Oleh karena itu dalam artikel ini akan dikaji pemodelan dengan menggunakan *Statistical Technique* dan *Neural Network* dengan *Algoritma Backpropagation* untuk data *timeseries* dan aplikasinya pada data harga emas.

**Tinjauan Pustaka**

**Emas**

Pada umumnya orang memilih berinvestasi dalam bentuk emas untuk memperoleh keuntungan (Aprianti 2012). Investasi emas dibedakan menjadi 2 yaitu, emas batangan dan saham emas (sertifikat). Keduanya mempunyai peluang investasi dan resiko masing-masing. Berinvestasi dalam emas batangan membutuhkan biaya untuk menyewa *safe deposit box* dan memungkinkan resiko lebih besar daripada berinvestasi emas dalam bentuk saham. Dalam berinvestasi emas berbentuk saham, investor hanya perlu keahlian membaca bursa saham.

**Konsep Dasar Time Series**

*Timeseries* adalah pengamatan pada suatu variabel dari waktu lampau dan dicatat secara berurutan menurut urutan waktu dengan periode yang tetap (Hanke 2005). Pada umumnya pencatatan ini dilakukan dalam periode tertentu misalnya harian, bulanan, tahunan dan sebagainya, sedangkan analisis time series adalah suatu metode kuantitatif untuk menentukan pola data masa lampau yang telah dikumpulkan secara teratur. Jika telah menemukan pola data tersebut, maka dapat digunakan untuk peramalan di masa mendatang. Beberapa konsep dasar dalam analisis time series adalah *Statistical Techniques* dan *Data Mining Forecasting Techniques* (Mishra & Jain 2014).

**Statistical Techniques Trend Linier**

Trend dinyatakan sebagai fungsi sederhana suatu garis lurus di sepanjang deret waktu yang diobservasi sehingga secara sistematis bentuk persamaannya adalah sebagai berikut:

$$T_t = a + b.Y_t$$

Dimana  $T_t$  = nilai trend periode t

a = konstanta nilai trend pada periode dasar

b = koefisien garis arah trend setiap periode

$Y_t$  = variabel independen mewakili waktu dan diasumsikan bernilai integer 1,2,3 ... sesuai dengan urutan waktu terkait.

Metode yang dapat digunakan untuk menentukan persamaan trend linear diantaranya adalah metode kuadrat terkecil (*least square method*). Metode ini memilih nilai-nilai koefisien dalam persamaan trend (a dan b) yang meminimumkan rata-rata kesalahan kuadrat (*mean square error*) yaitu :

$$b = \frac{n \sum Y_t T_t - \sum Y_t \sum T_t}{n \sum Y_t^2 - (\sum Y_t)^2}$$

$$a = \bar{Y}_t - b \bar{T}_t$$

**Trend Quadratic**

Ada kalanya deret berkala mempunyai data kecepatan (bernilai) atau kelambatan kenaikan pada tahap awal dan mengalami kecepatan atau kenaikan yang lebih besar pada tahap deret berkala berikutnya atau bisa dikatakan nilai data naik turun secara tidak sistematis. Dalam kasus seperti ini, trend nonlinear lebih baik digunakan dari pada trend linear. Ada beberapa jenis trend nonlinear, diantaranya:

Ekspensial

$$T_t = ab^y$$

Kuadratic

$$T_t = a + bY_t + cY_t^2$$

Kubik

$$T_t = a + bY_t + cY_t^2 + dY_t^3$$

Untuk menentukan trend yang paling tepat maka harus dipilih trend yang mempunyai derajat kesalahan paling kecil yaitu yang mempunyai selisih antara data asli (*actual*) dengan hasil estimasi (*trend*) yang

paling kecil. Untuk menentukan trend yang paling baik adalah memilih trend yang mempunyai nilai *Standard Error* paling kecil dan R-square yang paling besar.

**Moving Average**

*Moving average* (perataan bergerak) dapat digunakan untuk memuluskan data deret waktu dengan berbagai metode perataan, diantaranya *Single moving average* (rata-rata bergerak tunggal) dan *Double moving average* (rata-rata bergerak ganda). Untuk semua kasus dari metode tersebut, tujuannya adalah memanfaatkan data masa lalu untuk peramalan pada periode mendatang.

**Single Moving Average**

*Moving Average* (rata-rata bergerak tunggal) adalah metode yang merata-ratakan nilai dari t periode terakhir. Dengan kata lain, dengan munculnya data yang baru, maka nilai rata-rata yang baru dapat dihitung dengan menghilangkan data yang tertua dan menambahkan data yang terbaru. Metode ini untuk data yang stasioner dan data tidak mengandung trend dan musiman.

Jika diberikan N titik data dan diputuskan menggunakan T pengamatan pada setiap rata-rata ("rata-rata bergerak berorde T"), dinotasikan MA(T) maka keadaan yang ditemui adalah sebagai berikut:

$Y_1 Y_2 \dots \dots \dots Y_T \quad Y_{T+1} \dots \dots \dots Y_N$   
 Kelompok Inisialisasi      Kelompok Pengujian

| Waktu | Rata-rata Bergerak                               | Ramalan                              |
|-------|--|--------------------------------------|
| T     | $\gamma = \frac{y_1 + y_2 + \dots + y_T}{T}$     | $F_{T+1} = \sum_{i=1}^T y_i / T$     |
| T+1   | $\gamma = \frac{y_2 + y_3 + \dots + y_{T+1}}{T}$ | $F_{T+2} = \sum_{i=2}^{T+1} y_i / T$ |
| T+2   | $\gamma = \frac{y_3 + y_4 + \dots + y_{T+2}}{T}$ | $F_{T+3} = \sum_{i=3}^{T+2} y_i / T$ |
| ..... |  |                                      |

**Double Moving Average**

**Double Moving Average** (rata-rata bergerak ganda) didasarkan pada perhitungan rata-rata bergerak yang kedua. Rata-rata bergerak kedua dihitung dari rata-rata bergerak yang pertama, dinotasikan dengan MA (T x T) yang artinya adalah MA(T) periode dari MA(T)

$$S'_i = \frac{Y_i + \dots + Y_{n-T+1}}{T}$$

periode. Metode ini dapat digunakan untuk meramalkan data dengan komponen trend linear. Prosedur menghitung rata-rata bergerak ganda:

1. Hitung rata-rata bergerak tunggal
2. Hitung nilai penyesuaian yang merupakan selisih antara MA-tunggal dengan MA-ganda, di mana  $(S'_n - S''_n)$ ,  $S''_i = \frac{S'_i + \dots + S'_{n-T+1}}{T}$
3. Sesuaikan trend dari periode n ke n+m, jika anda ingin melakukan peramalan m periode ke depan

Ramalan untuk m periode ke depan adalah  $a_n$  – dimana ini merupakan nilai rata-rata yang disesuaikan untuk periode n- ditambah m kali komponen trend  $b_n$ .

**Exponential Smoothing**

Pada metode *Exponential Smoothing* (Pemulusan Eksponensial), pada dasarnya data masa lalu dimuluskan dengan cara melakukan pembobotan menurun secara eksponensial terhadap nilai pengamatan yang lebih tua.

Pada metode ini terdapat beberapa kategori, diantaranya:

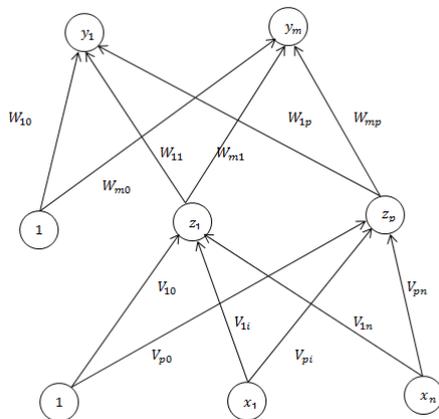
- 1). Pemulusan eksponensial tunggal (*single exponential smoothing*);
- 2). Pemulusan eksponensial ganda (*double exponential smoothing*) dengan metode Brown;
- 3). Pemulusan eksponensial ganda (*double exponential smoothing*) dengan metode Holt; dan
- 4). Pemulusan eksponensial tripel (*triple exponential smoothing*) dengan metode Winter.

**Data Mining Forecasting Techniques**

*Backpropagation* merupakan salah satu algoritma pembelajaran dalam jaringan saraf tiruan. Jaringan syaraf tiruan sendiri merupakan sebuah model yang mengadopsi cara kerja *neuron* secara biologi dengan fokus pada cara kerja saraf otak. Algoritma

*backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*) (Kusrini 2009). Untuk mendapatkan *error* ini, tahapan propagasi maju (*forward propagation*) harus dikerjakan terlebih dahulu. Jaringan Backpropagation umumnya terdiri dari banyak lapisan yakni:

1. Lapisan *input* (1 buah) yang terdiri dari 1 hingga n *input*.
2. Lapisan tersembunyi (minimal 1 buah) yang terdiri dari 1 hingga p *unit* tersembunyi.
3. Lapisan *output* (1 buah) yang terdiri 1 hingga m *unit output*.



Gambar 2.1. Arsitektur Algoritma BPNN

Algoritma pelatihan *Backpropagation* pada dasarnya terdiri dari 3 tahapan (Fausset 1994) yaitu :

1. Tahap propagasi maju (*feedforward*)
2. Tahap propagasi mundur (*backward*)
3. Tahap perubahan bobot dan bias

Selama propagasi maju (*feedforward*), sinyal masukan ( $x_i$ ) akan menerima sinyal *input* dan akan menyebarkan sinyal tersebut pada tiap *hidden unit* ( $z_j$ ), setiap *hidden unit* kemudian akan menghitung aktivasinya dan mengirim sinyal ( $z_j$ ) ke tiap unit *output*. Kemudian setiap unit *output* ( $y_k$ ) juga akan menghitung aktivasinya ( $y_k$ ) untuk menghasilkan respons terhadap *input* yang diberikan jaringan.

Saat proses pelatihan (*training*), setiap unit *output* membandingkan aktivasinya

( $y_k$ ) dengan nilai target ( $t_k$ ) untuk menentukan besarnya *error*. Berdasarkan *error* ini dihitung faktor  $\delta_k$ , di mana faktor ini digunakan untuk mendistribusikan *error* dari *output* ke *layer* sebelumnya. Dengan cara yang sama, faktor  $\delta_j$  juga dihitung pada *hidden unit*  $z_j$ , di mana faktor ini digunakan untuk memperbaharui bobot antara *hidden layer* dan *input layer*. Setelah semua faktor  $\delta$  ditentukan, bobot untuk semua *layer* diperbaharui.

Notasi yang digunakan dalam algoritma pelatihan:

$x$  = Data *training input*  $x = (x_1, \dots, x_i, \dots, x_n)$

$t$  = Data *training untuk target output*  $t = (t_1, \dots, t_k, \dots, t_m)$

$\alpha$  = *Learning rate* yaitu parameter untuk mengontrol perubahan bobot selama pelatihan.

$X_i$  = Unit *input* ke-i

$Z_j$  = *Hidden unit* ke-j

$Y_k$  = Unit *output* ke-k

$V_{j0}$  = Bias untuk *hidden unit* ke-j

$V_{ji}$  = Bobot antara unit *input* ke-i dengan *hidden unit* ke-j

$W_{k0}$  = Bias untuk unit *output* ke-k

$W_{kj}$  = Bobot antara *hidden unit* ke-j dengan unit *output* ke-k

$\delta_k$  = Faktor koreksi error untuk bobot  $w_{jk}$

$\delta_j$  = Faktor koreksi error untuk bobot  $v_{ij}$

$m$  = Momentum

Tahap-tahap pelatihan:

1. Tahap 0: Inisialisasi bobot dan bias. Baik bobot maupun bias dapat diset dengan sembarang angka (acak) dan biasanya angka di sekitar 0 dan 1 atau -1 (bias positif atau negatif).
2. Tahap 1: Jika *stopping condition* masih belum terpenuhi, jalankan tahap 2-9
3. Tahap 2: Untuk setiap data *training*, lakukan tahap 3-8
4. Tahap 3: Setiap unit *input* ( $X_i$ ,  $i=1, \dots, n$ ) menerima sinyal *input*  $X_i$  dan menyebarkan

sinyal tersebut pada seluruh unit pada *hidden layer*.

5. Tahap 4: Setiap *hidden unit* ( $Z_j, j=1, \dots, p$ ) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot, termasuk biasnya:

$$Z\_net_j = V_{0j} + \sum_{i=1}^n X_i V_{ij} \quad (1)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari *hidden unit* yang bersangkutan:

$$z_j = f(z\_net_j) = \frac{1}{1+e^{-z\_net_j}} \quad (2)$$

lalu mengirim sinyal *output* ini ke seluruh *unit* pada *unit output*

6. Tahap 5: Setiap *unit output* ( $y_k, k=1, \dots, m$ ) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot termasuk biasnya:

$$y\_net_k = W_{0k} + \sum_{j=1}^p Z_j W_{jk} \quad (3)$$

dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari *unit output* yang bersangkutan:

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

7. Tahap 6: Propagasi balik *error* (*backpropagation of error*).

Setiap *unit output* ( $y_k, k=1, \dots, m$ ) menerima suatu *target* (*output* yang diharapkan) yang akan dibandingkan dengan *output* yang dihasilkan:

$$\delta_k = (t_k - y_k) f'(y\_net_k) = (t_k - y_k) y_k (1 - y_k) \quad (5)$$

Faktor  $\delta_k$  ini digunakan untuk menghitung koreksi *error* ( $\Delta W_{jk}$ ) yang nantinya akan dipakai untuk memperbaharui  $W_{jk}$ , di mana:

$$\Delta W_{jk} = \alpha \delta_k z_j \quad (6)$$

Selain itu juga dihitung koreksi bias  $\Delta W_{0k}$  yang nantinya akan dipakai untuk memperbaharui  $W_{0k}$ , di mana:

$$\Delta W_{0k} = \alpha \delta_k \quad (7)$$

Faktor  $\delta_k$  ini kemudian dikirimkan ke *layer* di depannya.

8. Tahap 7: Setiap *hidden unit* ( $Z_j, j=1, \dots, p$ ) menjumlah *input delta* (yang dikirim dari *layer* pada tahap 6) yang sudah berbobot:

$$\delta\_net_j = \sum_{k=1}^m \delta_k W_{jk} \quad (8)$$

Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi *error*  $\delta_j$ , di mana:

$$\delta_j = \delta\_net_j f'(z\_net_j) = \delta\_net_j z_j (1 - z_j) \quad (9)$$

Faktor  $\delta_j$  ini digunakan untuk menghitung koreksi *error* ( $\Delta V_{ij}$ ) yang nantinya akan dipakai untuk memperbaharui  $V_{ij}$ , di mana:

$$\Delta V_{ij} = \alpha \delta_j X_i \quad (10)$$

Selain itu juga dihitung koreksi bias  $\Delta V_{0j}$  yang nantinya akan dipakai untuk memperbaharui  $V_{0j}$ , di mana:

$$\Delta V_{0j} = \alpha \delta_j \quad (11)$$

9. Tahap 8: Pembaharuan bobot dan bias: Setiap *unit output* ( $Y_k, k=1, \dots, m$ ) akan memperbaharui bias dan bobotnya dengan setiap *hidden unit*:

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (12)$$

Demikian pula untuk setiap *hidden unit* akan memperbaharui bias dan bobotnya dengan setiap *unit input*:

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad (13)$$

Pada standar *backpropagation*, perubahan bobot didasarkan atas gradien yang terjadi untuk untuk pola yang dimasukkan saat itu. Modifikasi yang dapat dilakukan adalah melakukan perubahan bobot yang didasarkan atas arah gradien pola terakhir dan pola sebelumnya (disebut momentum) yang dimasukkan. Penambahan momentum dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lain. Dengan penambahan momentum, bobot baru pada waktu ke (t+1) didasarkan atas bobot pada waktu t dan (t-1). Di sini harus ditambahkan 2 variabel baru yang mencatat besarnya momentum untuk 2 iterasi terakhir.

Adapun persamaan perubahan bobot dengan momentum adalah sebagai berikut:

$$w_{jk}(t + 1) = w_{jk}(t) + \alpha \delta_k z_j + \mu (w_{jk}(t) - w_{jk}(t - 1)) \tag{14}$$

dan

$$v_{ij}(t + 1) = v_{ij}(t) + \alpha \delta_k x_i + \mu (v_{ij}(t) - v_{ij}(t - 1)) \tag{15}$$

10. Tahap 9: Memeriksa *stopping condition*

Jika *stop condition* telah terpenuhi, maka pelatihan jaringan dapat dihentikan. Untuk menentukan *stopping condition* terdapat dua cara yang biasa dipakai, yaitu:

- Membatasi iterasi yang ingin dilakukan  
Misalnya jaringan akan dilatih sampai iterasi yang ke-500. Yang dimaksud dengan satu iterasi adalah perulangan tahap 3 sampai tahap 8 untuk semua *training data* yang ada.
- Membatasi *error*  
Misalnya menentukan besar Mean Square Error antara output yang dikehendaki dan output yang dihasilkan oleh jaringan.

Setelah pelatihan selesai, *backpropagation network (BPNN)* dianggap telah pintar sehingga apabila jaringan diberi *input* tertentu, jaringan akan menghasilkan *output* seperti yang diharapkan. Cara mendapatkan *output* tersebut adalah dengan mengimplementasikan metode *backpropagation* yang sama seperti proses pelatihan, tetapi hanya pada bagian umpan majunya saja

**Metodologi**

**Data Harga Emas Bulanan**

Sumber data untuk eksperimen diambil dari <http://data.okfn.org/data/core/gold-prices/>. Imana dalam dataset ini mendefinisikan harga emas bulanan sejak tahun 1950 dalam USD

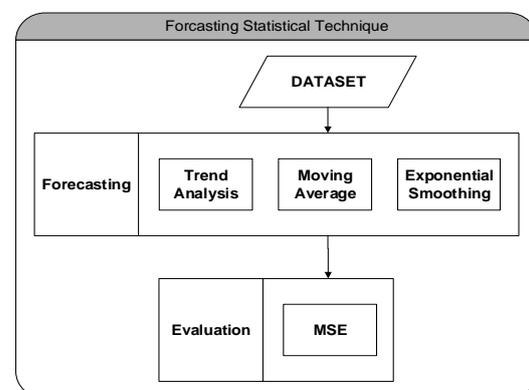
(pasar London). Data bersumber dari *Bundesbank*, dalam dataset ini terdiri dari 2 *attribute* yaitu tanggal (*date*) dan harga (*price*) dengan 768 *tuple (record)*. Harga tersebut merupakan harga emas murni dalam satuan ons, dan jika diubah kesatuan gram yaitu 31.1034768 gram. Adapun contoh data tersebut yaitu:

Tabel 3.1. Data Harga Emas Bulanan

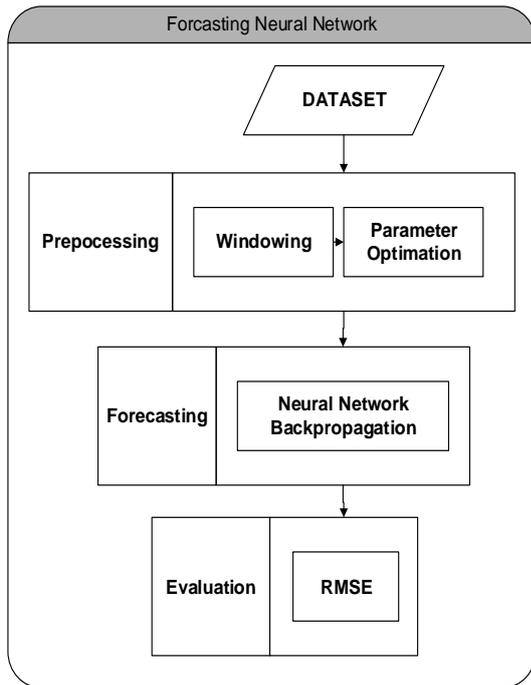
| Date       | Price    |
|------------|----------|
| 01/11/1950 | 34.73    |
| 01/12/1950 | 34.72    |
| 01/01/1951 | 34.72    |
| 01/02/1951 | 34.73    |
| 01/03/1951 | 34.73    |
| 01/04/1951 | 34.73    |
| 01/05/1951 | 34.73    |
| 01/06/1951 | 34.73    |
| .....      | .....    |
| .....      | .....    |
| 01/10/2013 | 1314.402 |
| 01/11/2013 | 1277.417 |
| 01/12/2013 | 1221.588 |

**Tahapan Eksperimen**

Pada artikel ini, beberapa tahapan digunakan dalam eksperimen untuk mendapatkan sebuah model terbaik menggunakan *Statistical Technique* dan *Neural Network Backpropagation* dengan mengoptimasi parameter.



Gambar 3.1. *Proposed Method for Statistical Technique*



Gambar 3.2. *Proposed Method for Neural Network Backpropagation*

Seperti yang terlihat pada Gambar 3.2, metode yang diusulkan adalah optimasi parameter pada *Neural Network*. Dataset yang berasal dari data.okfn.org kemudian dilakukan *windowing* untuk menjadikan dataset menjadi *multivariate* yang nantinya akan dimasukkan sebagai data testing dan data training pada model. Validasi data dilakukan dengan menggunakan 10 *Fold Cross Validation* untuk mengetahui nilai performance prediction yaitu nilai *Root Mean Squared Error* (RMSE) dari model yang kita buat berdasarkan dataset yang diolah. *Optimasi Parameter* dilakukan pada data *training* sebelum dimasukkan kedalam *Neural Network* untuk mendapatkan nilai akurasi yang lebih baik.

**Hasil Dan Pembahasan**

**Statistical Technique Trend Linier**

Dari pengujian data harga emas bulanan dengan menggunakan *software zaitun time series* (Penggunaan n.d.) menggunakan teknik *Trend Linier* (TR), *Trend Quadratic* (TQ), *Single Moving Average* (SMA), *Double*

*Moving Average* (DMA), *Exponential Smoothing* (ES) didapat nilai sebagai berikut:

Tabel 4.1. Hasil Eksperimen Menggunakan Teknik Statistik

|                             | TR       | TQ        | SMA     | DMA     | ES      |
|-----------------------------|----------|-----------|---------|---------|---------|
| Mean Squared Error (MSE)    | 54388.99 | 39225.455 | 760.55  | 864.26  | 2182.84 |
| Harga dibulan Desember 2015 | 849.43   | 1177.0033 | 1249.50 | 1249.50 | 1350.98 |

Pengujian terbaik diperoleh *Single Moving Average* dengan *Mean Squared Error* sebesar 760.55, untuk peramalan *multi step ahead* terlihat peramalan pada bulan desember 2015 yaitu sebesar US\$ 1249.50 /ons. Harga peramalan ini adalah harga emas dunia, namun harga tersebut dapat ditransformasikan ke dalam harga emas Indonesia dengan persamaan sebagai berikut :

$$\begin{aligned}
 & \text{harga emas Indonesia(gram)} \\
 &= \frac{\text{harga emas dunia}}{31.1} \times \text{kurs dolar}
 \end{aligned}$$

dengan demikian peramalan harga emas di Indonesia pada bulan Desember 2015 adalah:

$$\begin{aligned}
 & \text{harga emas Indonesia(gram)} = \\
 & \frac{1249.50}{31.1} \times 13645 = 548213.1
 \end{aligned}$$

dari hasil tersebut harga emas bulan Desember 2015 adalah Rp. 548.213 per gram.

**Windowing data Harga Emas Bulanan**

*Windowing* adalah proses untuk memecah data menjadi beberapa attribut. Data yang terbentuk diubah menjadi "*cross-sectional*" format. Berikut data hasil *windowing* menggunakan *window size* 8, *step size* 1 dan horizon 1.

| label  | date      | price-7 | price-6 | price-5 | price-4 | price-3 | price-2 | price-1 | price-0 |
|--------|-----------|---------|---------|---------|---------|---------|---------|---------|---------|
| 34.730 | 01 Agu 50 | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  |
| 34.730 | 01 Sep 50 | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  |
| 34.730 | 01 Okt 50 | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  |
| 34.720 | 01 Nov 50 | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  |
| 34.720 | 01 Des 50 | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.720  |
| 34.730 | 01 Jan 51 | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.720  |
| 34.730 | 01 Feb 51 | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.720  | 34.720  | 34.730  |
| 34.730 | 01 Mar 51 | 34.730  | 34.730  | 34.730  | 34.730  | 34.720  | 34.720  | 34.730  | 34.730  |
| 34.730 | 01 Apr 51 | 34.730  | 34.730  | 34.730  | 34.720  | 34.720  | 34.730  | 34.730  | 34.730  |
| 34.730 | 01 Mei 51 | 34.730  | 34.730  | 34.720  | 34.720  | 34.730  | 34.730  | 34.730  | 34.730  |
| 34.720 | 01 Jun 51 | 34.730  | 34.720  | 34.720  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  |
| 34.710 | 01 Jul 51 | 34.720  | 34.720  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.720  |
| 34.710 | 01 Agu 51 | 34.720  | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.720  | 34.710  |
| 34.710 | 01 Sep 51 | 34.730  | 34.730  | 34.730  | 34.730  | 34.730  | 34.720  | 34.710  | 34.710  |
| 34.730 | 01 Okt 51 | 34.730  | 34.730  | 34.730  | 34.730  | 34.720  | 34.710  | 34.710  | 34.710  |
| 34.660 | 01 Nov 51 | 34.730  | 34.730  | 34.730  | 34.730  | 34.720  | 34.710  | 34.710  | 34.730  |
| 34.490 | 01 Des 51 | 34.730  | 34.730  | 34.720  | 34.710  | 34.710  | 34.710  | 34.730  | 34.660  |
| 34.490 | 01 Jan 52 | 34.730  | 34.720  | 34.710  | 34.710  | 34.710  | 34.730  | 34.660  | 34.490  |

Gambar 4.1. *Windowing* Data Harga Emas Bulanan

**Parameter Optimization**

*Parameter Optimization* digunakan untuk mencari nilai learning rate dan momentum paling optimal yang akan digunakan untuk menguji data *Neural Network*. Dari hasil eksperimen didapat nilai learning rate dan momentum terbaik adalah 0.1. Berikut hasil parameter optimization yang didapat dari pengujian rapidminer.

```

ParameterSet

Parameter set:

Performance:
PerformanceVector [
----root_mean_squared_error: 22.730 +/- 6.945 (mikro: 23.768 +/- 0.000)
----squared_error: 564.899 +/- 331.882 (mikro: 564.899 +/- 2538.588)
]
Neural Net.learning_rate      = 0.1
Neural Net.momentum          = 0.1
    
```

Gambar 4.2. Parameter Optimization

**Pengujian Model dengan Neural Network**

Pada tahapan eksperimen pertama adalah dengan mencari jumlah *windowing* terbaik pada Neural Network, diperlihatkan pada Tabel 4.1.

Tabel 4.2. Hasil Eksperimen Pencarian *Windowing* Terbaik

| Windowing | Train cycle | Learning rate | Momentum | RMSE             |
|-----------|-------------|---------------|----------|------------------|
| 2         | 500         | 0.1           | 0.1      | 24.107 +/- 7.086 |
| 3         | 500         | 0.1           | 0.1      | 23.646 +/- 5.432 |
| 4         | 500         | 0.1           | 0.1      | 23.366 +/- 5.946 |
| 5         | 500         | 0.1           | 0.1      | 24.263 +/- 5.606 |
| 6         | 500         | 0.1           | 0.1      | 25.977 +/- 6.252 |

|    |     |     |     |                  |
|----|-----|-----|-----|------------------|
| 7  | 500 | 0.1 | 0.1 | 23.431 +/- 7.106 |
| 8  | 500 | 0.1 | 0.1 | 22.730 +/- 6.945 |
| 9  | 500 | 0.1 | 0.1 | 24.629 +/- 5.274 |
| 10 | 500 | 0.1 | 0.1 | 27.170 +/- 5.054 |
| 11 | 500 | 0.1 | 0.1 | 23.763 +/- 5.918 |

Berdasarkan hasil eksperimen pencarian jumlah *windowing* terbaik, maka didapatkan hasil *windowing* terbaik adalah 8 dengan hasil RMSE sebesar 22.730 +/- 6.945 seperti pada Tabel 4.1. Tahapan selanjutnya adalah mencari jumlah *neuronhidden* terbaik yang diterapkan pada model, adapun hasilnya adalah seperti pada Tabel 4.2.

Tabel 4.3. Hasil Eksperimen *neuron hidden* Terbaik

| Hidden | Training cycles | Learning rate | Momentum | RMSE              |
|--------|-----------------|---------------|----------|-------------------|
| 2      | 500             | 0.1           | 0.1      | 24.357 +/- 6.990  |
| 3      | 500             | 0.1           | 0.1      | 27.113 +/- 7.287  |
| 4      | 500             | 0.1           | 0.1      | 24.638 +/- 7.684  |
| 5      | 500             | 0.1           | 0.1      | 25.050 +/- 8.294  |
| 6      | 500             | 0.1           | 0.1      | 22.730 +/- 6.945  |
| 7      | 500             | 0.1           | 0.1      | 24.500 +/- 7.682  |
| 8      | 500             | 0.1           | 0.1      | 23.347 +/- 7.055  |
| 9      | 500             | 0.1           | 0.1      | 28.534 +/- 9.143  |
| 10     | 500             | 0.1           | 0.1      | 27.169 +/- 8.476  |
| 10     | 500             | 0.1           | 0.1      | 28.811 +/- 13.379 |

Pada Tabel 4.2 diperlihatkan hasil RMSE dengan jumlah neuron hidden terbaik pada Neural Network yaitu 6 neuron hidden dan mendapatkan RMSE sebesar 22.730 +/- 6.945.

**Validasi dan Evaluasi Data**

Setelah didapatkan nilai RMSE yang terbaik yang didapatkan pada Neural Network, kemudian pengujian dilanjutkan untuk mencari harga pada waktu yang akan datang (*step ahead*). Dari pengujian tersebut peramalan harga untuk bulan Desember 2015 didapat nilai 664.34 sehingga jika ditransformasikan ke Rupiah sebesar Rp. 291.476,-.

**Simpulan**

Eksperimen dilakukan untuk mencari nilai *error* terkecil dengan menggunakan teknik statistik dan data mining. Dari hasil pengujian, teknik statistik terbaik yaitu menggunakan teknik *Single Moving Average* dengan *MSE* sebesar 760.55. Pada teknik data mining dengan menggunakan metode *Neural Network Backprogration* didapat *RMSE* sebesar 22.730 +/- 6.945.

**Daftar Pustaka**

- Aprianti, M., 2012. *Anti Rugi Dengan Berinvestasi Emas*, Yogyakarta: Pustaka Baru Press.
- Computing, S., 2010. *Fundamentals of Neural Networks Soft Computing Fundamentals of Neural Networks Soft Computing*.
- Fausset, L., 1994. *Fundamentals of Neural Network. Architectures, Algorithms, and Applications*.
- Halim, A., 2005. *Analisis investasi* 2nd ed. R. Untung, ed., Jakarta: Salemba Empat.
- Hanke, J.E., 2005. *Business Forecasting 8th Edition* 8th ed., United States of Amerika: Pearson Education, Inc.
- Kusrini, E.T.L., 2009. *Algoritma Data Mining*. In *Data Mining*. Yogyakarta: CV. ANDI OFFSET, pp. 1–209.
- Mishra, N. & Jain, E.A., 2014. *Time Series Data Analysis for Forecasting – A Literature Review*. , 4(1972), pp.1–5.
- Penggunaan, P., Zaitun.
- Somantri, O., Sasmito, G.W. & Sungkar, M.S., 2014. *Optimalisasi Neural Network dengan Bootstrap Aggregating ( Bagging ) untuk Penentuan Prediksi Harga Listrik*. , 1(2), pp.185–192.
- Sudjana, 2005. *Metoda Statistika* 6th ed., Bandung: PT. Tarsito.
- Suprianto, E.D.Y., 2004. *Penerapan jaringan syaraf tiruan untuk memprediksi harga saham*.