

# Pengaruh Fungsi Pembelajaran Terhadap Kinerja Pelatihan Jaringan Syaraf Tiruan *Backpropagation*

## Studi Kasus : Indeks Harga Saham Gabungan di Bursa Efek Indonesia

Anis Fagustina  
Jurusan Informatika  
Universitas Sebelas Maret  
Jl. Ir. Sutami 36A Ketingan  
Surakarta 57126  
anis.fagustina@gmail.com

YS. Palgunadi  
Jurusan Informatika  
Universitas Sebelas Maret  
Jl. Ir. Sutami 36A Ketingan  
Surakarta 57126  
palgunadi@uns.ac.id

Wiharto  
Jurusan Informatika  
Universitas Sebelas Maret  
Jl. Ir. Sutami 36A Ketingan  
Surakarta 57126  
wi\_harto@yahoo.com

### ABSTRAK

*Ketidaklengkapan data dan ketepatan penggunaan fungsi pada jaringan syaraf tiruan backpropagation, yang digunakan dapat mempengaruhi cepat atau lambatnya kinerja proses pelatihan. Namun penggunaan data tidak lengkap (missing data) juga dapat menimbulkan error pada hasil pelatihan tersebut sehingga perlu dilakukan perbaikan missing data. Perbaikan missing data dapat dilakukan dengan beberapa cara, salah satunya adalah dengan menggunakan metode Autoregressive Integrated Moving Average (ARIMA). Penelitian ini mempelajari fungsi pembelajaran jaringan syaraf tiruan backpropagation yang paling efektif dengan perbaikan missing data input. Data latih yang digunakan adalah data Indeks Harga Saham Gabungan (IHSG) dan faktor pengaruh IHSG berupa indeks global yang diwakilkan oleh Hang Seng, KLSE, KOSPI, Nikkei, STI dan Taiex. Proses pelatihan dilakukan sampai pencapaian target nilai error.*

*Dari Penelitian ini didapat kesimpulan yaitu :*

- Fungsi pembelajaran bobot jaringan syaraf tiruan backpropagation terbaik adalah Resilient Backpropagation (trainrp). Namun beberapa pelatihan ketika digunakan data input kurang dari delapan data, fungsi backpropagation Levenberg-Marquardt (trainlm) memberikan hasil terbaik.*
- Penggunaan data dengan range waktu pengambilan semakin besar akan menghasilkan hasil latih yang semakin lebih baik.*
- Hasil pelatihan menggunakan IHSG sebagai data input lebih baik dibandingkan menggunakan kombinasi IHSG dan faktor pengaruh.*

**Kata Kunci:** Pelatihan Jaringan Syaraf Tiruan *Backpropagation*, *Missing Data*, IHSG, Indeks Global

### 1. LATAR BELAKANG

Seiring bertambahnya waktu, pasar modal dengan produk utama berupa saham turut serta menjadi salah satu penggerak perekonomian di suatu negara. Indeks Harga Saham Gabungan (IHSG) merupakan nilai yang digunakan untuk mengukur kinerja gabungan seluruh saham (perusahaan/emiten) tercatat di Bursa Efek Indonesia (BEI). Fenomena kecenderungan penurunan IHSG sering bersamaan dengan

perubahan sikap investor di Indonesia yang dipengaruhi oleh perubahan bursa global. Indeks global mempunyai pengaruh positif bagi pertumbuhan IHSG [1]. Pertumbuhan IHSG tersebut dapat dianalisa dengan menggunakan beberapa metode. Salah satu diantaranya adalah jaringan syaraf tiruan.

Jaringan syaraf tiruan adalah model komputasi dari otak manusia. Jaringan syaraf tiruan berasumsi perhitungan yang didistribusikan melalui beberapa unit sederhana yang disebut *neuron*, yang saling berhubungan dan beroperasi secara paralel sehingga dikenal sebagai sistem pemrosesan paralel terdistribusi atau sistem koneksi [2]. *Backpropagation* melatih jaringan untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama pelatihan serta kemampuan jaringan untuk memberikan respon yang benar terhadap pola masukan yang serupa dengan pola yang dipakai selama pelatihan [3].

Pada penelitian Rohim digunakan metode jaringan syaraf tiruan *backpropagation* untuk melakukan prediksi saham. Kesimpulan dari penelitian menyatakan bahwa metode jaringan syaraf tiruan *backpropagation* lebih baik (mendekati nilai data aktual) daripada data hasil prediksi metode *steepest descent* pada data harga penutupan saham PT. Indofood Sukses Makmur, Tbk dan PT. Gudang Garam, Tbk [4]. Sedangkan penelitian guna mendapatkan jumlah lapisan tersembunyi (*hidden layer*) metode jaringan syaraf tiruan *backpropagation* terbaik juga telah dilakukan penelitian sebelumnya, termasuk oleh Freisleben.

Data lengkap sangatlah dibutuhkan dalam perhitungan metode jaringan syaraf tiruan. Namun, pengumpulan data yang digunakan pada suatu penelitian tidak jarang diketemukannya *missing data* yang dapat mengakibatkan munculnya masalah serius seperti terhambatnya proses pelatihan serta kurang akuratnya hasil yang didapat ketika melakukan prediksi. *Missing data* dapat diatasi dengan menggunakan metode penggantian data [5]. Salah satu metode prediksi data *time series*, ARIMA dapat dimanfaatkan untuk menganalisis data *time series* mengatasi *missing data* guna mendapatkan data lengkap.

Berdasarkan masalah *missing data* yang ditemukan pada data, perlu diteliti fungsi pembelajaran yang seperti apa yang lebih baik digunakan ketika hasil penelitian Freisleben ikut dimanfaatkan pada proses penentuan jumlah *hidden layer* yang digunakan untuk pelatihan jaringan syaraf tiruan *backpropagation*. Fungsi pembelajaran yang efektif dapat dijadikan bahan pertimbangan pada penggunaan penelitian selanjutnya.

## 2. LANDASAN TEORI

### 2.1 Pasar Modal

Pasar modal menurut UU No.8 tahun 1995 adalah kegiatan yang bersangkutan dengan penawaran umum dan perdagangan efek, perusahaan yang berkaitan dengan efek yang diterbitkannya, serta lembaga dan profesi yang berkaitan dengan efek.

### 2.2 IHSG

Indeks harga saham adalah indikator atau cerminan pergerakan harga saham. Indeks merupakan salah satu pedoman bagi investor untuk melakukan investasi di pasar modal, khususnya saham [6]. Saham dapat didefinisikan sebagai surat berharga bukti penyertaan atau pemilikan individu maupun institusi dalam suatu perusahaan [7]. IHSG merupakan salah satu dari 11 jenis indeks harga saham yang dimiliki Indonesia. IHSG pertama kali diperkenalkan pada tanggal 1 April 1983. IHSG merepresentasikan rata-rata dari seluruh saham di BEI. Fungsi dari IHSG yaitu sebagai penanda arah pasar, pengukur tingkat keuntungan, dan tolok ukur kinerja portofolio.

### 2.3 Hubungan IHSG dengan Indeks Global

Indeks global seperti Dow Jones, KOSPI, Hang Seng dan KLSE berpengaruh lebih dominan terhadap pergerakan IHSG dibanding inflasi dan tingkat suku bunga SBI [8]. Walaupun peranan investor domestik makin meningkat akan tetapi terdapat kebiasaan dari investor domestik untuk melakukan strategi mengekor pada investor asing atau setidaknya investor domestik menggunakan perilaku investor asing sebagai acuan [9]. Apabila terjadi *shock* dari bursa lain secara cepat *shock* tersebut akan ditransmisikan ke hampir semua bursa saham di Asia Pasifik termasuk BEI [10].

### 2.4 Missing Data

Ketidaklengkapan data (*missing data process*) merupakan kejadian sistematis eksternal pada responden atau perilaku pada sebagian responden yang menyebabkan *missing data*. Terjadinya *missing data* pada data tersebut disebabkan oleh berbagai faktor, seperti data yang sudah tidak bisa terbaca lagi, data tidak relevan ketika dilakukan penelitian, peralatan yang tidak berfungsi dengan baik, penolakan dari responden untuk menjawab pertanyaan dan lain sebagainya.

### 2.5 Pola dan Mekanisme Missing Data

Terdapat dua pola data hilang (*missing data*), yaitu pola data hilang monoton dan pola data hilang nonmonoton [11]. Pola kehilangan data dikatakan monoton jika kejadian hilangnya peubah dari unit tertentu mengakibatkan *subsequent* peubah hilang untuk individu unit tersebut. Ilustrasi dari pola tersebut bisa dilihat pada tabel 1.

Tabel 1. Ilustrasi pola data hilang monoton

Group	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>
1	X	X	X
2	X	X	.
3	X	.	.

Sedangkan pola nonmonoton tidak ada keteraturan letak data hilang. Ilustrasi dari pola tersebut bisa dilihat pada tabel 2.

Tabel 2. Ilustrasi pola data hilang nonmonoton

Group	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>
1	X	X	X
2	X	.	X
3	.	X	.
4	.	.	X

Mekanisme *missing value* ada 3, yakni :

1. *Missingness is Non-Ignorable*, terjadi *missing data* pada suatu variabel berkaitan dengan variabel itu sendiri hingga tidak bisa diprediksi dari variabel lain.
2. *Missing at Random (MAR)*, terjadi *missing data* hanya berkaitan dengan variabel respon/pengamatan.
3. *Missing Completely at Random (MCAR)*, terjadinya *missing data* tidak berkaitan dengan nilai semua variabel, apakah itu variabel dengan *missing values* atau dengan variabel pengamatan.

### 2.6 Metode Penggantian Data

*Missing data* yang dialami dalam sebuah penelitian dapat mengakibatkan masalah terhadap hasil penelitian yang dilakukan. Kurang akuratnya data yang dihasilkan bisa menjadi salah satu masalah serius yang muncul dari adanya *missing data* tersebut. Oleh karena itu, diperlukan penanganan untuk meminimalkan resiko kurang akuratnya data yang didapat dalam penelitian. Penanganan data bisa dilakukan dengan cara melakukan penggantian data. Penggantian data dilakukan dengan cara melakukan analisa data *time series* adalah analisa yang menerangkan dan mengukur berbagai perubahan atau perkembangan data selama periode tertentu.

#### 2.6.1 ARIMA (1,0,1)

Salah satu model yang diyakini dapat digunakan untuk menganalisa perdagangan saham harian adalah *Auto Resressive Integrated Moving average (ARIMA)* yang dikembangkan oleh George Box dan Gwilym Jenkins (1976). Metodologi Box-Jenkins dapat digunakan untuk meramal tingkat *employment*, menganalisis pengaruh promosi terhadap penjualan barang-barang konsumsi, menganalisis persaingan antara jalur kereta api dengan jalur pesawat terbang, mengestimasi perubahan struktur harga suatu industri. ARIMA merupakan suatu metode yang menghasilkan ramalan-ramalan berdasarkan sintesis dari pola data secara historis [12]. Variabel ARIMA yang digunakan adalah nilai-nilai terdahulu bersama nilai kesalahannya.

Model umum dari ARIMA yaitu ARIMA (p,d,q), dengan p adalah order untuk bagian persamaan *Autoregressive (AR)*, sedangkan q adalah order untuk bagian persamaan *Moving Average (MA)*. Yang dimaksud dengan d adalah *differencing*. MA adalah model untuk memprediksi Y<sub>t</sub> sebagai fungsi dari kesalahan prediksi di masa lalu (*past forecast error*) dalam memprediksi Y<sub>t</sub>. AR adalah model untuk memprediksi Y<sub>t</sub> sebagai fungsi dari data di masa yang lalu, yakni t<sub>-1</sub>, t<sub>-2</sub>, ..., t<sub>-n</sub>. Proses *differencing* yang dilakukan pada metode Box-Jenkins (ARIMA) dapat menerapkan, menjelaskan, atau mewakili series yang stasioner atau telah dijadikan stasioner.

ARIMA (1,0,1) merupakan model ARIMA(p,d,q) yang berarti nilai AR sebesar 1, nilai MA sebesar 1 dan nilai d adalah 0. Model ARIMA (1,0,1) merupakan bentuk model campuran dari metode Box-Jenkins (ARIMA) dengan proses berupa proses ARMA. Model ini merupakan campuran dari proses AR (1) murni dan MA (1) murni.

## 2.7 Jaringan Syaraf Tiruan Backpropagation

Jaringan syaraf tiruan didefinisikan dengan sebuah *neural network* adalah prosesor yang terdistribusi paralel, terbuat dari unit-unit yang sederhana, dan memiliki kemampuan untuk menyimpan pengetahuan yang diperoleh secara eksperimental dan siap pakai untuk berbagai tujuan [13]. *Backpropagation* merupakan algoritma pembelajaran yang terawasi (*supervised learning*) yang diperkenalkan oleh David Rumelhart dan James McClelland pada tahun 1986 [14].

Jaringan syaraf tiruan *backpropagation* tersusun atas sejumlah *neuron* yang saling terhubung yang terbagi ke dalam beberapa lapisan yaitu lapisan *input*, *hidden*, dan *output*. Pada lapisan *input*, suatu JST terdiri atas beberapa *neuron*, hal ini merepresentasikan parameter dari suatu kasus yang terhubung ke lapisan *hidden*, dan lapisan *hidden* saling terhubung ke lapisan *output* [16]. Jaringan dengan satu hidden layer dapat dimodelkan dengan fungsi kontinyu [17]. Pertimbangan kekonvergensi dan generalisasi, struktur jaringan syaraf tiruan 3-32-16-1 akan relatif stabil [18]. Sedangkan penelitian lain mengatakan bahwa hasil terbaik dari struktur jaringan syaraf tiruan juga dapat dicapai dengan menggunakan jumlah *node hidden layer* dari hasil kelipatan jumlah inputan dikurangi satu [19].

$$\text{Jumlah hidden node} = (k \times n) - 1 \quad (1)$$

### 2.7.1 Algoritma Backpropagation

Algoritma *backpropagation* melakukan tahap perambatan maju (*forward propagation*) yang dikerjakan terlebih dahulu untuk mendapatkan *error output*. Kemudian *error output* yang didapat tersebut digunakan untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*) [15]. Langkah algoritma pelatihan yang dilakukan sebagai berikut:

- Inisialisasi bobot, maksimum *epoch*, *goal*, dan *learning rate* ( $\alpha$ ), *epoch* = 0, MSE = 1
- Kerjakan langkah-langkah berikut selama (*epoch* < maksimum *epoch*) dan (MSE < *goal*).
  - Epoch* = *epoch* + 1
  - Untuk tiap-tiap pasangan elemen yang akan dilakukan pembelajaran, kerjakan:

#### Feedforward

- Nilai input yang masuk ke *hidden layer* dihitung nilai sinyal input bobot dengan bantuan fungsi aktivasi untuk dikirim ke unit output. Perhitungan sinyal dilakukan sebanyak jumlah *hidden layer*.

$$z_{in_j} = b1_j + \sum_{i=1}^n X_i V_{ij} \quad (2)$$

$$z_j = f(z_{in_j}) \quad (3)$$

Tiap unit output menjumlahkan sinyal output dengan bantuan fungsi aktivasi.

$$y_{in_k} = b2_j + \sum_{i=1}^p z_j w_{jk} \quad (4)$$

$$y_j = f(y_{in_j}) \quad (5)$$

Nilai sinyal yang didapat dari hasil perhitungan dikirim ke lapisan atasnya (lapisan output).

#### Backpropagation

- Unit output menghitung koreksi error, koreksi bobot, dan koreksi bias yang didapat dari pola input pembelajaran yang diterima untuk nantinya digunakan ketika memperbaiki nilai bobot dan nilai bias pada *hidden layer* kembali. Langkah ini dilakukan sebanyak jumlah *hidden layer* sebelumnya. Koreksi error :

$$\delta 2_k = (t_k - y_k) f'(y_{in_k}) \quad (6)$$

$$\varphi 2_{jk} = \delta_k z_j \quad (7)$$

$$\beta 2_k = \delta_k \quad (8)$$

Koreksi bobot :

$$\Delta w_{jk} = \alpha \varphi 2_{jk} \quad (9)$$

Koreksi bias:

$$\Delta b 2_k = \alpha \beta 2_k \quad (10)$$

- Tiap unit *hidden* menjumlahkan *delta* inputnya dan menghitung error serta menghitung nilai koreksi bobot dan koreksi bias.

$$\delta_{in_j} = \sum_{k=1}^m \delta 2_k w_{jk} \quad (11)$$

$$\delta 1_j = \delta_{in_j} f'(z_{in_j}) \quad (12)$$

$$\varphi 1_{ij} = \delta 1_j x_j \quad (13)$$

$$\beta 1_j = \delta 1_j \quad (14)$$

$$\Delta v_{ij} = \alpha \varphi 1_{ij} \quad (15)$$

$$\Delta b 1_j = \alpha \beta 1_j \quad (16)$$

- Tiap unit *output* memperbaiki bias dan bobotnya.

$$w_{jk} (\text{baru}) = w_{jk} (\text{lama}) + \Delta w_{jk} \quad (17)$$

$$b 2_k (\text{baru}) = b 2_k (\text{lama}) + \Delta b 2_k \quad (18)$$

Tiap unit *hidden layer* memperbaiki bias dan bobotnya.

$$v_{ij} (\text{baru}) = v_{ij} (\text{lama}) + \Delta v_{ij} \quad (19)$$

$$b 1_j (\text{baru}) = b 1_j (\text{lama}) + \Delta b 1_j \quad (20)$$

- Hitung MSE

### 2.7.2 Fungsi Aktivasi

Fungsi aktivasi berfungsi menjembatani perbandingan antara hasil penjumlahan nilai semua bobot yang akan datang dengan nilai input dengan suatu nilai ambang (*threshold*) tertentu pada setiap neuron jaringan syaraf tiruan. 3 diantara fungsi aktivasi diantaranya yang disediakan pada toolbox matlab adalah sebagai berikut [15]:

- Fungsi *linier* (identitas / purelin) ; nilai outputnya sama dengan nilai inputnya.

$$y = x \quad (21)$$

- Fungsi *sigmoid biner* (logsig) ; memiliki nilai output pada range 0 sampai 1.

$$y = f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (22)$$

$$\text{dengan nilai } f'(x) = \sigma f(x)[1 - f(x)] \quad (23)$$

- Fungsi *sigmoid bipolar* (tansig) ; memiliki nilai output pada range -1 sampai 1.

$$y = f(x) = \frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}} \quad (24)$$

$$\text{dengan nilai } f'(x) = \frac{\sigma}{2} [1 + f(x)][1 - f(x)] \quad (25)$$

### 2.7.3 Fungsi Pembelajaran Backpropagation

Dalam rangka untuk mempercepat proses belajar, ada dua parameter dari algoritma *Backpropagation* yang disesuaikan yaitu : laju pembelajaran dan *momentum* [2]. Laju pembelajaran sangat berpengaruh pada intensitas proses pelatihan. Begitu pula terhadap efektivitas dan kecepatan mencapai konvergensi dari pelatihan [20]. *Backpropagation* memiliki bermacam-macam fungsi pembelajaran untuk bobot-bobot yang dapat terdapat di Matlab, beberapa di antaranya adalah seperti berikut.

- Gradient Descent* dengan *Momentum* (GDM / *traingdm*)  
Metode *traingdm* sama dengan *gradient descent*, hanya saja ditambahkan momentum yang memungkinkan ANN merespon tidak hanya kepada *local gradient error*-nya, tetapi juga trend yang baru saja terjadi pada perubahan *error*-nya [21].

Pengalihan dengan momentum dilakukan pada perubahan bobot dan bias untuk error lebih dari 1.

$$\Delta w_{jk} = mc * \Delta w_{jk}(\text{epoch sebelumnya}) + (1 - mc) * \alpha \phi 2_{jk} \quad (26)$$

$$\Delta b_{2k} = mc * \Delta b_{2k}(\text{epoch sebelumnya}) + (1 - mc) * \alpha \beta 2_k \quad (27)$$

$$\Delta v_{ij} = mc * \Delta v_{ij}(\text{epoch sebelumnya}) + (1 - mc) * \alpha \phi 1_{ij} \quad (28)$$

$$\Delta b_{1j} = mc * \Delta b_{1j}(\text{epoch sebelumnya}) + (1 - mc) * \alpha \beta 1_j \quad (29)$$

Akan tetapi, ketika nilai *epoch*=1 nilai perubahan bobotnya :

$$\Delta v_{ij} = \Delta v_{ij} + \alpha \phi 1_{ij} \quad (30)$$

b. *Gradient Descent* dengan *Adaptive Learning Rate* (GDA / *traingda*)

Algoritma *traingda* melakukan *update* pada *weight* sesuai dengan metode *gradient descent* dengan *adaptive learning rate* [21]. Setelah didapat nilai bobot dari persamaan 17, 18, 19 dan 20 dihitung nilai *learning rate* dengan mempertimbangkan nilai *error* baru dengan nilai *error* sebelumnya.

$$\alpha = \begin{cases} \alpha * lr_{dec} ; & \text{jika } \frac{\text{perf}_2}{\text{perf}} > \text{max\_perf\_inc} \\ \alpha * lr_{inc} ; & \text{jika } \frac{\text{perf}_2}{\text{perf}} < \text{max\_perf\_inc} \\ \alpha ; & \text{jika } \frac{\text{perf}_2}{\text{perf}} = \text{max\_perf\_inc} \end{cases} \quad (31)$$

c. *Gradient Descent* dengan *Momentum* dan *Adaptive Learning Rate* (GDX / *traingdx*)

Algoritma *traingdx* melakukan *update* pada *weight* sesuaikan dengan metode *gradient descent* dengan *adaptive learning rate* dikombinasikan dengan *momentum* [21]. Kinerja jaringan sesuai persamaan 31 dihitung setelah mendapatkan nilai dari persamaan 26 – 30. Maka:

$$\Delta w_{jk} = \alpha \phi 2_{jk} \quad (32)$$

$$\Delta b_{2k} = \alpha \beta 2_k \quad (33)$$

$$\Delta v_{ij} = \alpha \phi 1_{ij} \quad (34)$$

$$\Delta b_{1j} = \alpha \beta 1_j \quad (35)$$

d. *Resilient Backpropagation* (RP / *traingrp*)

*Traingrp* merupakan algoritma *backpropagation* yang mana perubahan membesar / mengecilnya *weights* ditentukan oleh perubahan tanda (*sign*) dari turunan pertama perubahan *error*-nya [21]. Besarnya perubahan bobot ( $\Delta v$ ,  $\Delta w$ ,  $\Delta b_1$ , dan  $\Delta b_2$ ) diinisialisasikan dengan parameter  $\delta_0$ . Besarnya perubahan tersebut tidak boleh melebihi batas maksimum yang terdapat pada parameter  $\delta_{\text{max}}$ , apabila perubahan bobot melebihi maksimum perubahan bobot, maka perubahan bobot akan diset sama dengan maksimum perubahan bobot.

e. *Algoritma Levenberg-Marquardt* (LM / *trainlm*)

Salah satu bentuk fungsi *backpropagation* teknik optimasi numerik standar. Fungsi pembelajaran *trainlm* merupakan algoritma tercepat untuk *training feedforward neural network* berukuran besar (sampai ratusan *weight*) [21].

f. *Algoritma One Step Secant* (OSS / *trainoss*)

Algoritma *trainoss* menggunakan pendekatan (aproksimasi) *secant* sehingga perhitungan rumit tersebut bisa lebih cepat dan tidak membutuhkan banyak kapasitas memori [21]. Algoritma ini menjembatani algoritma Quasi-Newton dengan *gradient conjugate*. Akan tetapi algoritma ini tidak menyimpan matriks Hessian secara lengkap karena mengasumsikan matriks Hessian sebelumnya merupakan matriks identitas untuk setiap iterasi. Sehingga pencarian arah baru dihitung tanpa menghitung invers matriks.

### 3. METODE PENELITIAN

#### 3.1 Persiapan

a. *Review Jurnal*

*Review jurnal* dilakukan untuk mempelajari beberapa jurnal sebagai acuan penelitian. Jurnal didapat dari hasil *browsing* internet.

b. *Studi Pustaka*

*Studi pustaka* dilakukan untuk mempelajari metode jaringan syaraf tiruan *backpropagation*, *missing data*, saham IHSG, dan faktor berpengaruh pergerakan saham IHSG. Sumber didapat dari buku dan jurnal yang telah dibukukan.

#### 3.2 Pelatihan

a. *Pemilihan Data*

Data dependen yang digunakan adalah data *indeks* harga saham gabungan (IHSG) di Bursa Efek Indonesia (BEI). Data independen sebagai faktor pengaruh pergerakan IHSG adalah data bursa global yang diwakili oleh Hang Seng, KLSE, KOSPI, Nikkei, STI dan Taiex. Semua data yang digunakan diunduh dari website [www.finance.yahoo.com](http://www.finance.yahoo.com) dari data harian periode bulan Januari 2011 sampai bulan Desember 2011.

b. *Perbaikan Missing Data*

Data awal IHSG, Hang Seng, KLSE, KOSPI, Nikkei, STI dan Taiex yang dimiliki mengalami *missing data* sehingga data yang dimiliki tidaklah lengkap. Usaha untuk mendapatkan data lengkap dengan melakukan penggantian data *missing*, memanfaatkan metode ARIMA dengan bantuan Minitab 16. Dari usaha perbaikan *missing data* tersebut didapat data lengkap.

c. *Pengelompokan Data*

Sebelum memasuki langkah pelatihan jaringan syaraf tiruan, data lengkap dikenai perlakuan pengelompokan data. Pengelompokan data didasarkan pada periode pengambilan dan kombinasi *input* dengan bantuan *ms.excel*. Periode waktu pengambilan data lengkap *input* yang digunakan adalah sebagai berikut:

1. 1 minggu 2 kali, pada penelitian ini diberi tanda 1. Cara pengambilannya dengan:  
t dipengaruhi oleh  $t_{-3}, t_{-5}, t_{-8}, \dots, t_{-20}$ , dst.
  2. 1 minggu 1 kali, pada penelitian ini diberi tanda 1. Cara pengambilannya dengan:  
t dipengaruhi oleh  $t_{-5}, t_{-10}, t_{-15}, \dots, t_{-40}$ , dst.
  3. 2 minggu 1 kali, pada penelitian ini diberi tanda 1. Cara pengambilannya dengan:  
t dipengaruhi oleh  $t_{-10}, t_{-20}, t_{-30}, \dots, t_{-80}$ , dst.
  4. 4 minggu 1 kali, pada penelitian ini diberi tanda 1. Cara pengambilannya dengan:  
t dipengaruhi oleh  $t_{-20}, t_{-40}, t_{-60}, \dots, t_{-160}$ , dst.
- Kombinasi *input* yang digunakan pada penelitian ini adalah sebagai berikut:

1. Kombinasi I (A) terdiri *close* IHSG. Penggunaan data IHSG yakni 8 data input, 23 *hidden layer* dengan target sejumlah 1 data.
  2. Kombinasi II (B) yaitu *open, high, low, close past, dan close now* IHSG. Penggunaan data masing-masing *open, high, low, close past, dan close now* IHSG yakni 44 data input, 131 *hidden layer* dengan target sejumlah 1 data.
  3. Kombinasi III (C) yaitu *close* dari IHSG, Hang Seng, KLSE, KOSPI, Nikkei, STI dan Taiex. Penggunaan data masing-masing IHSG, Hang Seng, KLSE, KOSPI, Nikkei, STI dan Taiex yakni 56 data input, 167 *hidden layer* dengan target sejumlah 1 data.
- d. *Pelatihan Jaringan Syaraf Tiruan*

Data yang telah dikelompokkan kemudian digunakan pada pelatihan *backpropagation* dengan bantuan program matlab. Fungsi aktivasi ke lapisan tersembunyi yang akan digunakan pada proses pelatihan adalah *tansig* dan *logsig*. Sedangkan fungsi aktivasi ke lapisan keluaran adalah *purelin*. Sehingga dari tahap pengelompokan dan tahap pelatihan ini didapat susunan *input* seperti pada tabel 3. Pelatihan dilakukan hingga 10 kali pelatihan. Dari tahap ini didapat jumlah nilai dari jumlah *max epoch* untuk mencapai target nilai *error*, total rata-rata *cpitime* mencapai target nilai *error* dan *generalization accuracy* (%).

**Tabel 3. Susunan input**

Periode Waktu Pengambilan	Model Kombinasi Input	Fungsi Aktivasi ke Lapisan Tersembunyi	Fungsi Pembelajaran Bobot <i>Backpropagation</i>	Pemberian Nama Pelatihan
1	Kombinasi I (A)	Tansig (Tan)	<i>Traingdm</i> (Gdm)	1ATanGdm
			<i>Traingda</i> (Gda)	1ATanGda
			<i>Traingdx</i> (Gdx)	1ATanGdx
			<i>Trainrp</i> (Rp)	1ATanRp
			<i>Trainlm</i> (Lm)	1ATanLm
			<i>Trainoss</i> (Oss)	1ATanOss
		Logsig (Log)	<i>Traingdm</i> (Gdm)	1ALogGdm
			<i>Traingda</i> (Gda)	1ALogGda
			<i>Traingdx</i> (Gdx)	1ALogGdx
			<i>Trainrp</i> (Rp)	1ALogRp
			<i>Trainlm</i> (Lm)	1ALogLm
			<i>Trainoss</i> (Oss)	1ATanOss
	Kombinasi II (B)	Tansig (Tan)	<i>Traingdm</i> (Gdm)	1BTanGdm
			<i>Traingda</i> (Gda)	1BTanGda
			<i>Traingdx</i> (Gdx)	1BTanGdx
			<i>Trainrp</i> (Rp)	1BTanRp
			<i>Trainlm</i> (Lm)	1BTanLm
			<i>Trainoss</i> (Oss)	1BTanOss
		Logsig (Log)	<i>Traingdm</i> (Gdm)	1BLogGdm
			<i>Traingda</i> (Gda)	1BLogGda
			<i>Traingdx</i> (Gdx)	1BLogGdx
			<i>Trainrp</i> (Rp)	1BLogRp
			<i>Trainlm</i> (Lm)	1BLogLm
			<i>Trainoss</i> (Oss)	1BLogOss
Kombinasi III (C)	Tansig (Tan)	<i>Traingdm</i> (Gdm)	1CTanGdm	
		<i>Traingda</i> (Gda)	1CTanGda	
		<i>Traingdx</i> (Gdx)	1CTanGdx	
		<i>Trainrp</i> (Rp)	1CTanRp	
		<i>Trainlm</i> (Lm)	1CTanLm	
		<i>Trainoss</i> (Oss)	1CTanOss	
	Logsig (Log)	<i>Traingdm</i> (Gdm)	1CLogGdm	
		<i>Traingda</i> (Gda)	1CLogGda	
		<i>Traingdx</i> (Gdx)	1CLogGdx	
		<i>Trainrp</i> (Rp)	1CLogRp	
		<i>Trainlm</i> (Lm)	1CLogLm	
		<i>Trainoss</i> (Oss)	1CLogOss	
2	3	8	<i>Traingdm</i> (Gdm)	2ATanGdm

3	Kombinasi II (B)	Logsig (Log)	<i>Traingda</i> (Gda)	2ATanGda
			<i>Traingdx</i> (Gdx)	2ATanGdx
			<i>Trainrp</i> (Rp)	2ATanRp
			<i>Trainlm</i> (Lm)	2ATanLm
			<i>Trainoss</i> (Oss)	2ATanOss
			<i>Traingdm</i> (Gdm)	2ALogGdm
		<i>Traingda</i> (Gda)	2ALogGda	
		<i>Traingdx</i> (Gdx)	2ALogGdx	
		<i>Trainrp</i> (Rp)	2ALogRp	
		<i>Trainlm</i> (Lm)	2ALogLm	
		<i>Trainoss</i> (Oss)	2ALogOss	
		Tansig (Tan)	<i>Traingdm</i> (Gdm)	2BTanGdm
	<i>Traingda</i> (Gda)		2BTanGda	
	<i>Traingdx</i> (Gdx)		2BTanGdx	
	<i>Trainrp</i> (Rp)		2BTanRp	
	<i>Trainlm</i> (Lm)		2BTanLm	
	<i>Trainoss</i> (Oss)		2BTanOss	
	Kombinasi III (C)	Logsig (Log)	<i>Traingdm</i> (Gdm)	2BLogGdm
			<i>Traingda</i> (Gda)	2BLogGda
			<i>Traingdx</i> (Gdx)	2BLogGdx
			<i>Trainrp</i> (Rp)	2BLogRp
			<i>Trainlm</i> (Lm)	2BLogLm
			<i>Trainoss</i> (Oss)	2BLogOss
		Tansig (Tan)	<i>Traingdm</i> (Gdm)	2CTanGdm
<i>Traingda</i> (Gda)			2CTanGda	
<i>Traingdx</i> (Gdx)			2CTanGdx	
<i>Trainrp</i> (Rp)			2CTanRp	
<i>Trainlm</i> (Lm)			2CTanLm	
<i>Trainoss</i> (Oss)			2CTanOss	
Kombinasi I (A)	Logsig (Log)	<i>Traingdm</i> (Gdm)	2CLogGdm	
		<i>Traingda</i> (Gda)	2CLogGda	
		<i>Traingdx</i> (Gdx)	2CLogGdx	
		<i>Trainrp</i> (Rp)	2CLogRp	
		<i>Trainlm</i> (Lm)	2CLogLm	
		<i>Trainoss</i> (Oss)	2CLogOss	
	Tansig (Tan)	<i>Traingdm</i> (Gdm)	3ATanGdm	
		<i>Traingda</i> (Gda)	3ATanGda	
		<i>Traingdx</i> (Gdx)	3ATanGdx	
		<i>Trainrp</i> (Rp)	3ATanRp	
		<i>Trainlm</i> (Lm)	3ATanLm	
		<i>Trainoss</i> (Oss)	3ATanOss	
Kombinasi II (B)	Logsig (Log)	<i>Traingdm</i> (Gdm)	3ALogGdm	
		<i>Traingda</i> (Gda)	3ALogGda	
		<i>Traingdx</i> (Gdx)	3ALogGdx	
		<i>Trainrp</i> (Rp)	3ALogRp	
		<i>Trainlm</i> (Lm)	3ALogLm	
		<i>Trainoss</i> (Oss)	3ALogOss	
	Tansig (Tan)	<i>Traingdm</i> (Gdm)	3BTanGdm	
		<i>Traingda</i> (Gda)	3BTanGda	
		<i>Traingdx</i> (Gdx)	3BTanGdx	
		<i>Trainrp</i> (Rp)	3BTanRp	
		<i>Trainlm</i> (Lm)	3BTanLm	
		<i>Trainoss</i> (Oss)	3BTanOss	

4	Kombinasi III (C)	Logsig (Log)	Traingdm (Gdm)	3BLogGdm
			Traingda (Gda)	3BLogGda
			Traingdx (Gdx)	3BLogGdx
			Trainrp (Rp)	3BLogRp
			Trainlm (Lm)	3BLogLm
			Trainoss (Oss)	3BLogOss
		Tansig (Tan)	Traingdm (Gdm)	3CTanGdm
			Traingda (Gda)	3CTanGda
			Traingdx (Gdx)	3CTanGdx
			Trainrp (Rp)	3CTanRp
			Trainlm (Lm)	3CTanLm
			Trainoss (Oss)	3CTanOss
	Logsig (Log)	Traingdm (Gdm)	3CLogGdm	
		Traingda (Gda)	3CLogGda	
		Traingdx (Gdx)	3CLogGdx	
		Trainrp (Rp)	3CLogRp	
		Trainlm (Lm)	3CLogLm	
		Trainoss (Oss)	3CLogOss	
	Kombinasi I (A)	Tansig (Tan)	Traingdm (Gdm)	4ATanGdm
			Traingda (Gda)	4ATanGda
			Traingdx (Gdx)	4ATanGdx
			Trainrp (Rp)	4ATanRp
			Trainlm (Lm)	4ATanLm
			Trainoss (Oss)	4ATanOss
Logsig (Log)		Traingdm (Gdm)	4ALogGdm	
		Traingda (Gda)	4ALogGda	
		Traingdx (Gdx)	4ALogGdx	
		Trainrp (Rp)	4ALogRp	
		Trainlm (Lm)	4ALogLm	
		Trainoss (Oss)	4ALogOss	
Kombinasi II (B)	Tansig (Tan)	Traingdm (Gdm)	4BTanGdm	
		Traingda (Gda)	4BTanGda	
		Traingdx (Gdx)	4BTanGdx	
		Trainrp (Rp)	4BTanRp	
		Trainlm (Lm)	4BTanLm	
		Trainoss (Oss)	4BTanOss	
	Logsig (Log)	Traingdm (Gdm)	4BLogGdm	
		Traingda (Gda)	4BLogGda	
		Traingdx (Gdx)	4BLogGdx	
		Trainrp (Rp)	4BLogRp	
		Trainlm (Lm)	4BLogLm	
		Trainoss (Oss)	4BLogOss	
Kombinasi III (C)	Tansig (Tan)	Traingdm (Gdm)	4CTanGdm	
		Traingda (Gda)	4CTanGda	
		Traingdx (Gdx)	4CTanGdx	
		Trainrp (Rp)	4CTanRp	
		Trainlm (Lm)	4CTanLm	
		Trainoss (Oss)	4CTanOss	
	Logsig (Log)	Traingdm (Gdm)	4CLogGdm	
		Traingda (Gda)	4CLogGda	
		Traingdx (Gdx)	4CLogGdx	
		Trainrp (Rp)	4CLogRp	

			Trainlm (Lm)	4CLogLm
			Trainoss (Oss)	4CLogOss

### 3.3 Perbandingan Hasil Pelatihan

Hasil pelatihan yang didapat pada pelatihan jaringan syaraf tiruan *backpropagation* berupa nilai dari jumlah max *epoch* mencapai target nilai *error*, total *cpitime* mencapai target nilai *error*, dan *generalization accuracy* (%). Dari hasil pelatihan tersebut dapat dihitung nilai *learning performance index (LPI)*. LPI merupakan *ratio* dari *generalization accuracy* dengan total *cpitime* untuk mencapai target nilai *error*.

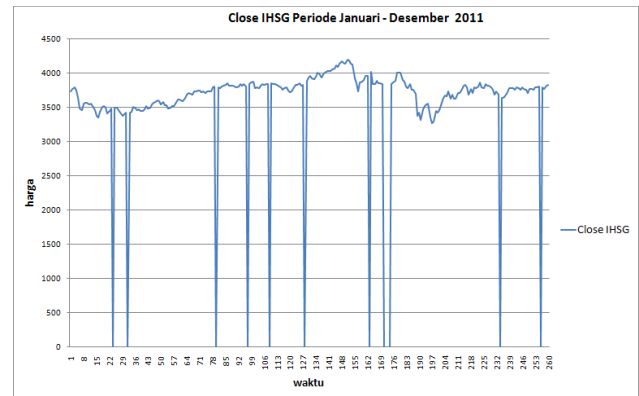
$$LPI = \frac{accuracy (\%)}{\text{total rata-rata } cpitime \text{ untuk target nilai } error} \quad (6)$$

Analisa hasil pelatihan yang dilakukan dengan cara melakukan perbandingan nilai LPI yang dihasilkan. Dari proses analisa ini dapat diketahui fungsi pembelajaran *backpropagation* yang paling efektif dengan kasus *missing data*. Fungsi pembelajaran *backpropagation* dikatakan efektif atau bagus jikalau nilai LPI yang dimiliki besar.

## 4. HASIL DAN PEMBAHASAN

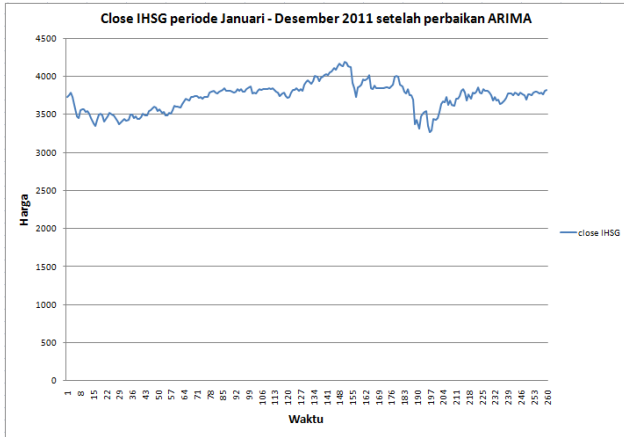
### 4.1 Hasil Perbaikan *Missing Data*

Pada gambar 1 menggambarkan kondisi data hasil unduh dari *www.finance.yahoo.com* yang digunakan pada penelitian. Nilai dari harga *close* IHSG yang ditampilkan pada gambar 1 mencapai nilai 0 dikarenakan terdapat data yang hilang.



Gambar 1. Contoh Grafik Harga *Close* IHSG

Kondisi kehilangan data pada harga *close* pada periode waktu tertentu tersebut diikuti kehilangan data harga *open*, *high*, dan *low* pada periode tersebut pula. Maka upaya untuk mendapatkan data lengkap, penelitian ini melakukan perbaikan *missing data*. Selain itu perbaikan *missing data* dilakukan untuk meminimalisir masalah yang ditimbulkan karena ketidaklengkapan data. Perbaikan data *missing* tersebut dilakukan dengan menggunakan ARIMA (1,0,1). Contoh hasil perbaikan data *missing* harga *close* IHSG pada gambar 1 dengan menggunakan ARIMA (1,0,1), dapat dilihat pada gambar 2.



Gambar 2. Grafik Data Perbaikan Missing Data dengan ARIMA (1,0,1)

## 4.2 Hasil Pelatihan

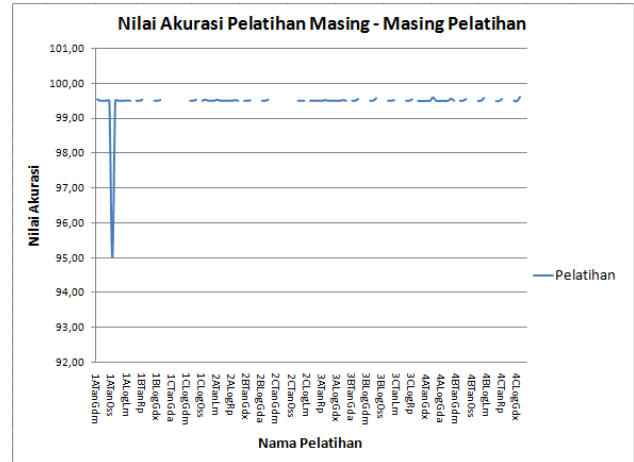
Tabel 6 merupakan hasil 10 kali pelatihan jaringan syaraf *backpropagation* dengan *missing data*.

Tabel 6. Contoh Hasil Pelatihan

Pemberian Nama Pelatihan	Pencapaian Epoch	CPU time (s)	General Accuracy (%)	Cpu time / Epoch ( $\times 10^{-2}$ )	Learning Performance Index ( $\times 10^{-2}$ )
1ATanGdm	47562	574,15	99,54	1,21	17,34
1ATanGda	38078	464,52	99,50	1,22	21,42
1ATanGdx	160976	4664,45	99,50	2,90	2,13
1ATanRp	3678	28,19	99,50	0,77	352,93
1ATanLm	116	4,30	99,50	3,71	2312,60

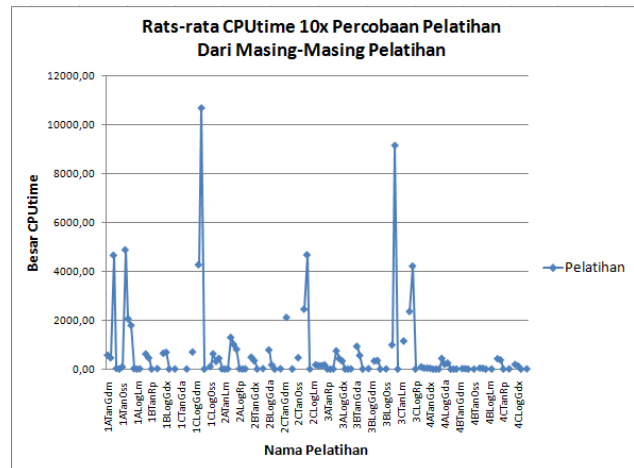
Dari tabel 6 di atas, pemberian nama berupa 1ATanGdm yang berarti pelatihan yang dilakukan dengan cara pengambilan data 1 minggu 2 kali, kombinasi input A yang berupa harga saham *close* dari IHSG, dengan fungsi aktivasi lapisan 1 berupa *tansig* dan fungsi pembelajaran bobot berupa *traingdm*, mencapai *epoch* maksimum sejumlah 47562 dengan total *CPUtime* mencapai target nilai *error* 574,15 s memiliki nilai keakuratan pelatihan sebesar 99,54 %. Rasio dari nilai *cpu time* dan *epoch* pelatihan 1ATanGdm ini adalah  $1,21 \times 10^{-2}$ . Sedangkan nilai *learning performance index* yang merupakan hasil rasio dari akurasi terhadap jumlah *cpu time* sebesar  $17,34 \times 10^{-2} \%$ .

Besar nilai keakuratan yang didapatkan pada masing-masing pelatihan pada dasarnya hampir sama, hanya saja ketika pelatihan 1ATanLm dijalankan didapatkan keakuratan terendah sebesar 95,00 %. Dan nilai akurasi tertinggi didapat dari pelatihan 4CLogRp sebesar 99,61 % dengan rata-rata akurasi keseluruhan pelatihan sebesar 99,47 %. Grafik nilai akurasi tersebut bisa dilihat pada grafik gambar 3.

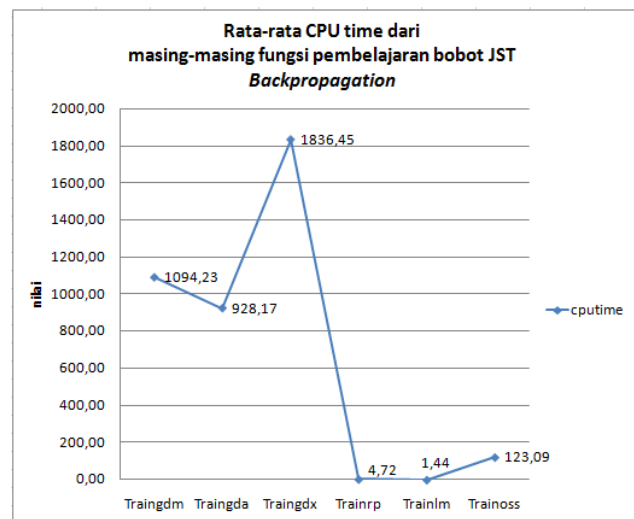


Gambar 3. Grafik Akurasi Pelatihan

*Cputime* yang dilalui pada keseluruhan proses pelatihan yang dilakukan bisa dilihat pada gambar 4. Pada gambar tersebut ditunjukkan bahwasannya nilai *cputime* tertinggi yang pernah dilakukan adalah hasil pelatihan 1CLogGdx.



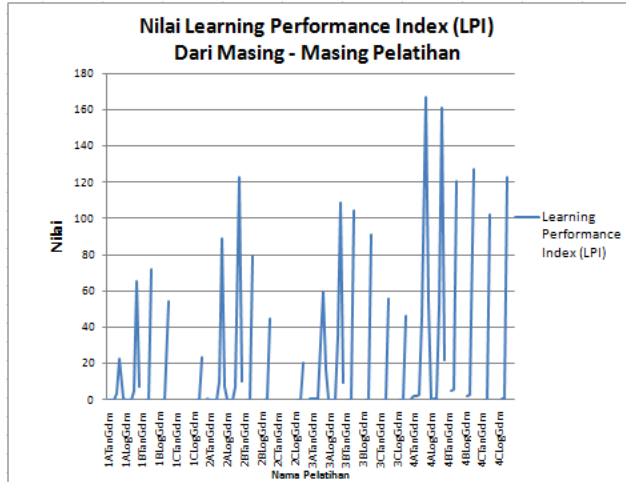
Gambar 4. Grafik CPUtime Mencapai Target Nilai Error





**Gambar 5. Rata-rata CPUtime Mencapai Target Nilai Error**

Pada gambar 5 ditunjukkan bahwasannya fungsi pembelajaran bobot Traingdx mempunyai jumlah nilai cputime yang paling banyak sebesar 1836,45 second.



**Gambar 6. Grafik LPI Keseluruhan Pelatihan**

Dari grafik yang disajikan pada gambar 6, dapat diketahui bahwa data pelatihan 4ATanLm mempunyai nilai LPI tertinggi dengan jumlah  $16669,90 \times 10^{-2}$  (%)/(s). Nilai LPI terendah didapatkan dari pelatihan 1CLogGdx dengan nilai LPI sejumlah  $0,93 \times 10^{-2}$  (%)/(s). Akan tetapi pergerakan nilai LPI dari keseluruhan pelatihan searah. Grafik-grafik terputus dikarenakan tidak semua proses pelatihan yang dilakukan bisa berjalan lancar dengan menghasilkan nilai yang dapat ditampilkan. Hal tersebut mengakibatkan data hasil LPI pelatihan yang didapat ikut kosong. Hingga di akhir pelatihan didapat grafik dengan kesan terputus pada pelatihan tertentu tersebut. Lebih jelasnya dapat dilihat pada tabel 7.

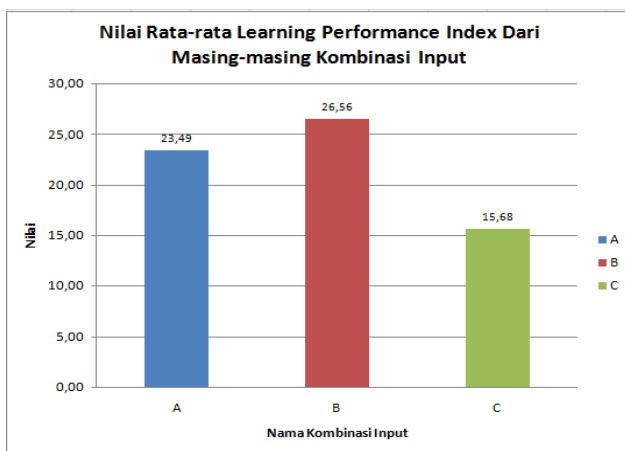
**Tabel 7. Hasil LPI Pelatihan**

Pemberian Nama Pelatihan	LPI ( $\times 10^{-2}$ )	Pemberian Nama Pelatihan	LPI ( $\times 10^{-2}$ )
1ATanGdm	17,34	3ATanGdm	69,03
1ATanGda	21,42	3ATanGda	69,56
1ATanGdx	2,13	3ATanGdx	56,75
1ATanRp	352,93	3ATanRp	3114,34
1ATanLm	2312,60	3ATanLm	5956,53
1ATanOss	102,88	3ATanOss	1666,62
1ALogGdm	2,04	3ALogGdm	13,33
1ALogGda	4,83	3ALogGda	23,10
1ALogGdx	5,54	3ALogGdx	29,71
1ALogRp	515,78	3ALogRp	3500,64
1ALogLm	6542,36	3ALogLm	10886,43
1ATanOss	718,10	3ALogOss	980,50
1BTanGdm	-	3BTanGdm	-
1BTanGda	15,99	3BTanGda	10,63
1BTanGdx	21,68	3BTanGdx	17,78
1BTanRp	7201,00	3BTanRp	10410,06
1BTanLm	-	3BTanLm	-
1BTanOss	485,44	3BTanOss	546,40
1BLogGdm	-	3BLogGdm	-

1BLogGda	15,31	3BLogGda	30,11
1BLogGdx	14,47	3BLogGdx	28,18
1BLogRp	5452,52	3BLogRp	9079,16
1BLogLm	-	3BLogLm	-
1BLogOss	1274,10	3BLogOss	1204,79
1CTanGdm	-	3CTanGdm	-
1CTanGda	-	3CTanGda	10,06
1CTanGdx	-	3CTanGdx	1,09
1CTanRp	2731,82	3CTanRp	5586,20
1CTanLm	-	3CTanLm	-
1CTanOss	14,17	3CTanOss	8,66
1CLogGdm	-	3CLogGdm	-
1CLogGda	2,33	3CLogGda	4,21
1CLogGdx	0,93	3CLogGdx	2,36
1CLogRp	2346,48	3CLogRp	4647,29
1CLogLm	-	3CLogLm	-
1CLogOss	95,32	3CLogOss	109,28
2ATanGdm	15,84	4ATanGdm	251,04
2ATanGda	31,54	4ATanGda	238,00
2ATanGdx	22,48	4ATanGdx	285,21
2ATanRp	1035,92	4ATanRp	4542,85
2ATanLm	8885,92	4ATanLm	16669,90
2ATanOss	752,67	4ATanOss	5396,08
2ALogGdm	7,66	4ALogGdm	22,81
2ALogGda	9,84	4ALogGda	51,07
2ALogGdx	12,27	4ALogGdx	40,11
2ALogRp	707,90	4ALogRp	5193,95
2ALogLm	12268,17	4ALogLm	16075,59
2ALogOss	1043,21	4ALogOss	2234,03
2BTanGdm	-	4BTanGdm	-
2BTanGda	20,36	4BTanGda	499,86
2BTanGdx	28,78	4BTanGdx	590,06
2BTanRp	7914,15	4BTanRp	12040,32
2BTanLm	-	4BTanLm	-
2BTanOss	500,65	4BTanOss	4982,94
2BLogGdm	-	4BLogGdm	-
2BLogGda	12,65	4BLogGda	250,40
2BLogGdx	58,83	4BLogGdx	296,55
2BLogRp	4496,19	4BLogRp	12690,44
2BLogLm	-	4BLogLm	-
2BLogOss	728,18	4BLogOss	4068,52
2CTanGdm	-	4CTanGdm	-
2CTanGda	4,70	4CTanGda	23,35
2CTanGdx	-	4CTanGdx	26,45
2CTanRp	3195,79	4CTanRp	10193,88
2CTanLm	-	4CTanLm	-
2CTanOss	21,29	4CTanOss	937,74
2CLogGdm	-	4CLogGdm	-
2CLogGda	4,06	4CLogGda	52,06
2CLogGdx	2,13	4CLogGdx	76,43
2CLogRp	2046,25	4CLogRp	12232,21
2CLogLm	-	4CLogLm	-
2CLogOss	56,66	4CLogOss	1052,78

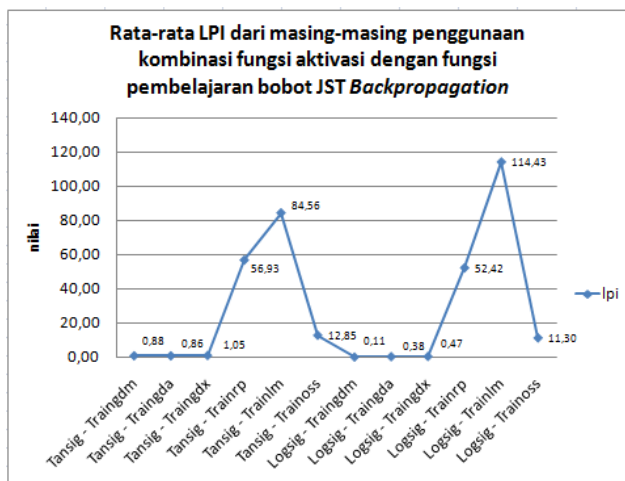


Dari tabel LPI hasil pelatihan di atas, dapat diketahui bahwa pelatihan yang dilakukan dengan fungsi pembelajaran bobot *traingdm* dan *trainlm* dari jaringan syaraf tiruan *backpropagation* dengan kasus *missing data* untuk *input* model B dan model C sering tidak bisa menampilkan hasil LPI dikarenakan terjadi penghentian proses pelatihan di tengah-tengah perjalanan pelatihan. Penghentian proses tersebut dilakukan karena pada pelatihan *traingdm* lebih sering menghasilkan *error* berupa *not a number*. Sedangkan *trainlm* terlalu lama memakan waktu dalam mencapai target nilai *error*. Selain waktu yang terlalu lama tersebut, *trainlm* juga membutuhkan *memory* yang sangat banyak sehingga mengakibatkan terjadinya *out of memory* dalam proses pelatihannya. Dan beberapa pelatihan seperti 1CTanGda, 1CTanGdx, dan 2CTanGdx juga dilakukan penghentian dikarenakan pelatihan tersebut terlampaui lama untuk mencapai target *error*, sehingga terpaksa dihentikan.



Gambar 7. Rata-rata LPI dari Model Kombinasi *Input*

Dari gambar 7 di atas ditunjukkan bahwa model variasi kombinasi *input* B yang berupa kombinasi dari harga *open, high, low, close past*, dan *close now* dari IHSB memberikan nilai LPI paling tinggi sebesar 26,56 (%)(s).



Gambar 8. Rata-rata LPI dari Kombinasi Fungsi Aktivasi dan Fungsi Pembelajaran Bobot

Variasi penggunaan kombinasi fungsi aktivasi dengan fungsi pembelajaran bobot jaringan syaraf tiruan *Backpropagation* terhadap besarnya nilai LPI yang didapat selama mencapai target *error*, kombinasi *Logsig - TrainLm* mempunyai nilai terbaik LPI sebesar 114,43 (%)(s), dengan nilai terendah didapatkan oleh *Logsig - TrainGdm* sebesar 0,11 (%)(s). Nilai LPI berdasarkan hanya fungsi pembelajaran bobot tertinggi didapatkan ketika menggunakan *Trainlm* dan nilai LPI akan bernilai tinggi dengan menggunakan *tansig* jika penilaian berdasarkan penggunaan fungsi aktivasi pada lapisan tersembunyi saja.

## 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

Berdasarkan pelatihan jaringan syaraf tiruan *backpropagation* yang telah dilakukan pada penelitian, didapatkan beberapa kesimpulan yakni sebagai berikut :

- Penggunaan fungsi pembelajaran *JST Resilient Backpropagation (traingrp)* lebih baik hasil *learning performance index* pelatihannya dibandingkan metode lain saat pelatihan. Akan tetapi untuk beberapa kasus latih dengan jumlah data *input* kurang dari 8 data, hasil pelatihan *JST backpropagation Levenberg-Marquardt (trainlm)* justru menjadi yang terbaik. Meskipun ketika menggunakan jumlah data *input* lebih dari 8 data, pelatihan menggunakan *JST backpropagation Gradient Descent dengan Momentum (traingdm)* mempunyai nilai *output* yang kurang menyenangkan karena nilai keluar dari *range*-nya.
- Penggunaan data dengan *range* waktu pengambilan data yang semakin besar akan mendapatkan hasil *learning performance index* pelatihan jaringan syaraf tiruan *backpropagation* yang lebih baik pula.
- Penggunaan data *input* pada pelatihan jaringan syaraf tiruan *backpropagation* berupa data IHSB yang dikombinasikan dengan data bursa dari negara lain sebagai faktor pengaruh IHSB, tidaklah lebih baik ketika hanya menggunakan data *input* berupa data IHSB itu sendiri. Meskipun pada penelitian lain membuktikan bahwa pergerakan IHSB dipengaruhi nilai bursa dari negara lain.

### 5.2 Saran

Saran untuk penelitian selanjutnya adalah menggunakan faktor analisis komponen utama (PCA) untuk menentukan arsitektur jaringan syaraf tiruan.

## 6. DAFTAR PUSTAKA

- Wicaksono, A.W., Widyaningsih, P. dan Sutanto., (2012) *Pengaruh Indeks Global Terhadap Fluktuasi Indeks Harga Saham Gabungan (IHSB) Menggunakan Hukum Pendingin Newton*, Jurusan Matematika, Universitas Sebelas Maret, Surakarta.
- Agarkar, A.M. dan Ghatol, A.A., (2010) *FFANN Based Cost Effective Major Infant Disease Management, International Journal of Computer Applications, Vol.7, 0975-8887.*
- Siang, J.J., (2009) *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan Matlab*, Penerbit Andi, Yogyakarta.

- [4] Rohim, Abd., (2011) *Implementasi Jaringan Syaraf Tiruan Backpropagation Stepest Descent untuk Prediksi Data Time Series*, Jurusan Matematika, Institut Teknologi Sepuluh November, Surabaya.
- [5] Palgunadi, S., dkk., (1998) *Missing Data Pada Analisis Diskriminan*, Universitas Sebelas Maret, Surakarta.
- [6] *Bursa Efek Indonesia*, www.idx.co.id, diakses 09 Juli 2013.
- [7] Anoraga, P. dan Pakarti, P., (2006) *Pengantar Pasar Modal*, Penerbit Rineka Cipta, Jakarta.
- [8] Mauliano, D. A., (2009) Analisis Faktor-Faktor yang Mempengaruhi Pergerakan Indeks Harga Saham Gabungan (IHSG) di Bursa Efek Indonesia, *Tugas Akhir Universitas Gunadarma*.
- [9] Cahyono, J.E., (2000) *22 Strategi dan Teknik Meraih Untung di Bursa Saham, Jilid 1*, Penerbit PT.Elex Media Komputindo, Jakarta.
- [10] Achsani, N. A., (2000) *Mencermati Kejatuhan Indeks Dow Jones: Akankah Indeks BEJ Ikut Terseret?*, University of Postdam, Postdam.
- [11] Little, R.J.A., dan Rubin, D.B., (1987) *Statistical Analysis with Missing Data*, Wiley, New York.
- [12] Arsyad, Lincolin., (1995) *Peramalan Bisnis*, Penerbit Ghalia Indonesia, Jakarta.
- [13] Haykin, S., (2004) *Digital Communication System*, Jhon Wiley & Sons, New York.
- [14] Fausett, L., (1994) *Fundamental of Neural Networks: Architectures, Algorithms dan Applications*, Prentice-Hall Inc., USA.
- [15] Kusumadewi, S., (2004) *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab & Excel Link, Edisi 1*, Penerbit Graha Ilmu, Yogyakarta.
- [16] Efendigil, T., Önut, S. dan Kahraman, C., (2008) A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy models: A comprehensive analysis, *Expert Systems with Applications*, 36, pp.6697-6707.
- [17] Beale, R dan Jackson, T., (1990) *Neural Computing: An Introduction*, Adam Hilger.
- [18] Refenes, A.N., Zapranis, A. dan Francis, G., (1994) Stock Performance Modeling using neural network: a comparative study with regression models, *Neural Network* 5, 961-970.
- [19] Freisleben.B., (1992) Stock Market Prediction with Backpropagation Network, Industrial and Engineering Application of Artificial Intelligence and Expert System, *5th International Conference*, Paderborn, Germany, June, 451-160.
- [20] Hermawan, A., (2006) *Jaringan Syaraf Tiruan teori dan aplikasi*. Penerbit Andi, Yogyakarta.
- [21] Demuth, H. dan Hagan, M., (2009) *Matlab Neural Network Toolbox Users Guide, Vol. 6*, MathWorks.