

Identifikasi Publikasi Dosen dalam Mewujudkan Internasionalisasi Universitas Negeri Semarang Menggunakan Neural Network

Walid[✉], Y L Sukestiyarno, Sunarmi

Jurusan Matematika, FMIPA, Universitas Negeri Semarang, Indonesia

Info Artikel

Sejarah Artikel:

Diterima 11 August 2018

Disetujui 23 September 2018

Dipublikasikan 1 October 2018

Keywords:

backpropagation neural network, internasionalisasi, sinta

Abstrak

Penelitian ini difokuskan pada penerapan *Backpropagation Neural Network* (BPNN) untuk melakukan identifikasi publikasi dosen dalam mewujudkan internasionalisasi UNNES dengan pengambilan datanya didasarkan pada penelusuran hasil publikasi ilmiah dosen di SINTA dan didukung dengan kuesioner yang diberikan pada responden dosen di FMIPA UNNES. Selanjutnya hasil data yang ada dianalisis dengan menggunakan program Matlab dan didukung dengan studi literatur terkait kajian NN. Studi simulasi dilakukan untuk mengetahui kondisi perkembangan hasil publikasi ilmiah dosen dengan menggunakan BPNN. Tujuan penelitian ini adalah untuk mengetahui hasil identifikasi publikasi dosen dalam mewujudkan internasionalisasi UNNES menggunakan BPNN. Berdasarkan hasil penelitian dan hasil pembahasan di bagian sebelumnya, maka disimpulkan bahwa Hasil pelatihan dan analisis numerik memberikan bahwa model terbaik BPNN pada algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* diperoleh model BPNN terbaik dengan algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* pada arsitektur jaringan 9-10-1 dengan momentum = 0,8 dan LR = 0,1 pada fungsi aktivasi tansig. Hasil identifikasi publikasi dosen menggunakan BPNN dan analisis terhadap pelatihan di atas diperoleh bahwa publikasi dosen dengan karya jurnal bereputasi internasional sebanyak 23,75%, untuk jurnal internasional sebanyak 21,23%, dan jurnal nasional terakreditasi sebesar 19,02% dan karya publikasi dalam prosiding internasional sebanyak 31,0%.

Abstract

This research is focused on the application of Backpropagation Neural Network (BPNN) to identify lecturer publications in realizing internationalization of UNNES with the retrieval of data based on the results of scientific publication of lecturers at SINTA and supported by questionnaires given to lecturer respondents at FMIPA UNNES. Furthermore, the results of the existing data were analyzed using the Matlab program and supported by literature studies related to the NN study. Simulation studies were conducted to determine the conditions for the development of the results of lecturers' scientific publications by using BPNN. The purpose of this study was to find out the results of the identification of lecturer publications in realizing the internationalization of UNNES using BPNN. Based on the results of the study and the results of the discussion in the previous section, it was concluded that the training results and numerical analysis provided that the best BPNN model on the momentum gradient descent training algorithm and adaptive learning rate obtained the best BPNN model with a gradient descent training algorithm with momentum and adaptive learning rate on network architecture 9-10-1 with momentum = 0.8 and LR = 0.1 on tansig activation function. The results of the identification of lecturer publications using BPNN and analysis of the training above obtained that lecturer publications with works of international reputation 23.75%, 21.23% for international journals, and 19.02% for national journals and 31.0% for publications in international proceedings.

© 2018 Universitas Negeri Semarang

[✉] Alamat korespondensi:

E-mail: walid.mat@mail.unnes.ac.id

PENDAHULUAN

Perkembangan ilmu pengetahuan begitu dinamis dan kompleks, salah satunya perkembangan di bidang Jaringan saraf tiruan (JSN). JSN merupakan model nonparametrik yang mempunyai bentuk fungsional yang fleksibel, mengandung beberapa parameter yang tidak dapat diinterpretasikan seperti pada model parametrik. Dalam penerapannya, JSN mengandung sejumlah parameter (*weight*) yang terbatas. Bagaimana mendapatkan model JSN yang sesuai, yaitu bagaimana menentukan kombinasi yang tepat antara jumlah variabel *input* dan jumlah unit pada *hidden layer* (yang berimplikasi pada jumlah parameter yang optimal), merupakan topik sentral dalam beberapa literatur JSN yang telah banyak dibahas pada banyak artikel dan banyak buku seperti pada Bishop (1995), Ripley (1996), Beale & Finlay (1992), Fine (1999), Haykin (1999).

Penelitian lain telah berhasil menerapkan sub peta intermittency untuk pemodelan lalu lintas internet, yang merupakan contoh yang berhubungan namun berbeda dengan fenomena yang menunjukkan long memory dan distribusi marginal dapat dilihat pada Swanson & White H (1995). Dewasa ini banyak penelitian yang menggunakan model peramalan dengan menggunakan neural network pada berbagai bidang, antara lain di bidang pelabelan, pembelajaran konteks bahasa dan bidang kelistrikan (lihat pada chen (1995), Husen (2001), Taylor (2003), Taylor et. al (2006) dan Ristiana (2008). Penelitian berkaitan dengan pengembangan jaringan saraf tiruan telah cukup banyak dilakukan oleh ahli dibidang ini, antara lain hasil penelitian yang dilakukan oleh Walid et. al (2015) tentang peramalan menggunakan model jaringan saraf tiruan pada data berpola long memory dengan konsentrasi ada penggunaan beban konsumsi listrik PLN di Jawa dan Bali. Menurut Ren et. al (2013) telah mengembangkan penggunaan terkait dengan trend data dari data historis dan menyatakan

bahwa BPNN efektif untuk meramalkan data yang baru.

Penelitian yang sudah dilakukan dengan menggunakan model RNN antara lain Walid et. al (2018) tentang Analisis Produktivitas Kinerja Dosen dan Tenaga Kependidikan dalam Mewujudkan Tahun Reputasi Universitas Negeri Semarang (UNNES) Menggunakan Jaringan Saraf Tiruan dengan menggunakan RNN. Produktivitas di lingkungan Universitas Negeri Semarang (UNNES) merupakan suatu interkasi terpadu secara serasi dari tiga faktor esensial, yaitu optimalisasi kemampuan, manajemen dan sumber daya yang dimiliki oleh UNNES. Pemberian motivasi kepada seluruh civitas akademika UNNES sangatlah berpengaruh terhadap jalannya kemajuan UNNES dalam mewujudkan tahun reputasi yang dicanangkan pada tahun 2017 ini oleh Walid et. al (2017).

Menurut Hardianto et.al (BPNN pada dasarnya adalah fungsi pemetaan dari neuron input ke neuron output tanpa mengetahui korelasi antar data. BPNN adalah model NN yang diakui secara kritis karena kapasitas pemodelan non parametrik dan non liniernya, kemampuan beradaptasi yang kuat, dan kemampuan komputasi paralel. Standar BPNN terdiri dari lapisan input, satu atau lebih lapisan tersembunyi, dan lapisan output. Secara umum, BPNN dengan satu lapisan tersembunyi dapat menghasilkan akurasi yang diinginkan untuk aplikasi peramalan runtun waktu (Wang et.al, 2018). BPNN adalah suatu kelas dari FFNN dengan aturan pembelajaran terawasi. Proses pembelajaran terawasi adalah proses membandingkan setiap perkiraan jaringan dengan jawaban benar yang diketahui dan menyesuaikan berat berdasarkan error perkiraan yang dihasilkan untuk meminimalkan fungsi *error*. Pendekatan BPNN memberikan alternatif kompetitif untuk prosedur yang ada untuk pembelajaran serta peramalan data independen.

Tahun 2018 merupakan tahun internasionalisasi bagi Universitas Negeri Semarang (UNNES) untuk dapat sejajar dengan perguruan tinggi lainnya di nasional ataupun internasional, salah satunya dalam bidang

penelitian dan pengabdian. Sebagai indikator keberhasilan program itu adalah adalah meningkatnya kualitas dan kuantitas publikasi ilmiah Dosen, baik di tingkat nasional maupun internasional. Oleh karena itu, upaya untuk meningkatkan publikasi dosen-dosen, terutama untuk publikasi ilmiah tingkat internasional menjadi salah satu agenda penting. Selain itu, juga akan mendorong mahasiswa-mahasiswa, terutama mahasiswa kelas internasional untuk mengikuti berbagai kegiatan yang berskala nasional dan internasional.

Berdasarkan uraian di atas, fokus permasalahan dalam penelitian ini adalah Bagaimana model identifikasi publikasi dosen dalam mewujudkan internasionalisasi UNNES menggunakan BPNN?. Tujuan penelitian ini adalah untuk mengetahui hasil identifikasi publikasi dosen dalam mewujudkan internasionalisasi UNNES menggunakan BPNN.

METODE

Metode penelitian ini dibagi menjadi beberapa tahapan penelitian, secara lengkap dijabarkan sebagai berikut.

Metode Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan metode penelusuran hasil publikasi dosen di SINTA terkait dengan publikasi Dosen, populasi diambil dosen di FMIPA UNNES, dengan sampel diambil secara acak dosen di beberapa jurusan yang ada di FMIPA. Berdasarkan data ini dilakukan analisis sintesis terkait dengan identifikasi publikasi dengan menggunakan neural network dengan model BPNN. Variabel-variabel penelitian ini meliputi variabel tentang publikasi dosen pada jurnal bereputasi internasional dinyatakan dengan X_1 , jurnal internasional dinyatakan dengan X_2 , jurnal nasional terakreditasi dinyatakan dengan X_3 dan karya publikasi dalam prosiding internasional dinyatakan dengan X_4 .

Metode Analisis Data

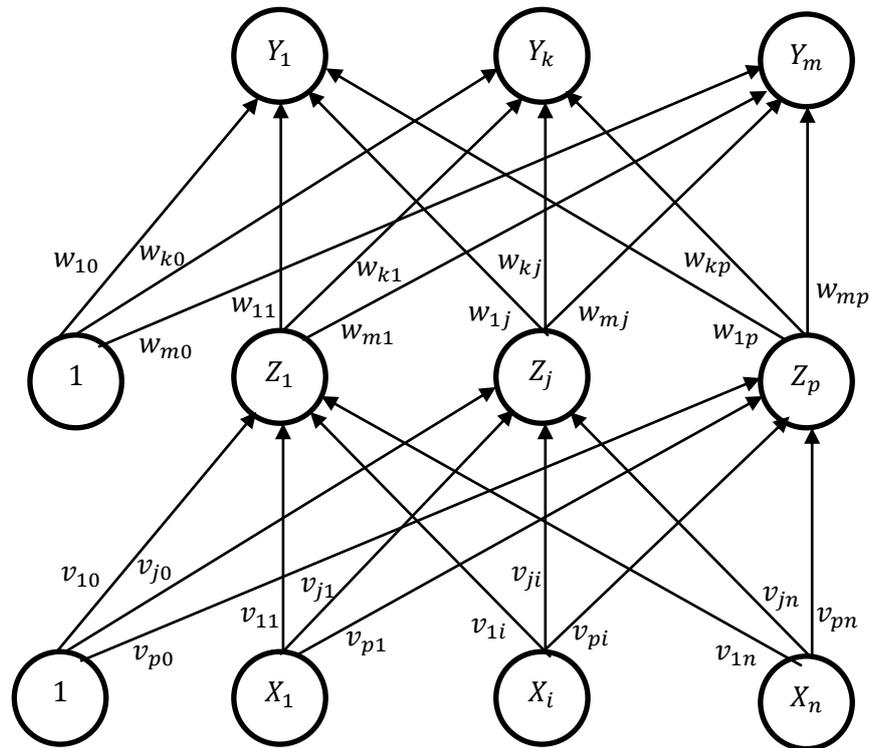
Metode analisis data yang digunakan dalam penelitian ini adalah metode analisis data didasarkan pada kajian teori, kajian komputasi dan kajian terapan. Berdasarkan fokus penelitian tersebut, tahapan penelitian ini dijabarkan secara detail dalam pemaparan berikut. Kajian Teori dengan studi *literature* dan kajian teoritis mendalam dilakukan untuk memperoleh kajian identifikasi data yang memuat model jaringan syaraf tiruan, produktivitas dosen dalam mempublikasikan jurnal baik ditingkat nasional maupun internasional. Fokus variabel ini yang menjadi objek dalam penelitian ini. Selain itu untuk memperoleh kajian pemetaan terhadap penggunaan aplikasi model jaringan saraf tiruan yang berkembang saat ini diuji secara mendalam sebagai dasar penggunaan keilmuaan untuk menganalisis kondisi yang sebenarnya. Kajian Komputasi dilakukan untuk memperoleh hasil analisis dengan menggunakan jaringan syaraf tiruan dan komputasional tahap-tahap identifikasi data yang memuat penentuan variabel, penyusunan fokus pada pemilihan indikator disetiap variabel yang menjadi objek pilihan data pada penelitian ini. Analisis selanjutnya dilakukan simulasi dan analisis komputasional model jaringan syaraf tiruan dengan menggunakan model BPNN dengan menggunakan simulasinya menggunakan program MATLAB dan R.

HASIL DAN PEMBAHASAN

Penelitian ini menggunakan model BPNN yang memiliki beberapa neuron yang ada dalam satu atau lebih lapisan tersembunyi. Gambar 1 adalah arsitektur BPNN dengan n input (ditambah sebuah bias), sebuah lapisan tersembunyi yang terdiri dari p neuron (ditambah sebuah bias), serta m buah neuron output. v_{ji} merupakan bobot garis dari neuron input X_i ke neuron lapisan tersembunyi Z_j (v_{j0} merupakan bobot garis yang menghubungkan bias di neuron input ke neuron lapisan tersembunyi Z_j). w_{kj} merupakan bobot dari

neuron lapisan tersembunyi Z_j ke neuron $output Y_k$ (w_{k0} merupakan bobot dari bias di

lapisan tersembunyi ke neuron $output Y_k$) (Siang, 2004: 98).



Gambar 1. Arsitektur Backpropagation Neural Network

Selama *feedforward*, setiap neuron *input* (X_i) menerima sinyal *input* dan mengirimkannya ke setiap neuron tersembunyi (Z_1, \dots, Z_p). Setiap neuron tersembunyi menghitung aktivasi dan mengirimkan sinyal (z_j) ke setiap neuron *output*. Setiap neuron *output* (Y_k) menghitung aktivasi (y_k) untuk membentuk respons jaringan untuk pola *input* yang diberikan.

Selama pelatihan, setiap neuron *output* membandingkan perhitungan aktivasi y_k dengan nilai targetnya t_k untuk menentukan *error* terkait untuk pola tersebut dengan neuron itu. Berdasarkan *error* ini, faktor δ_k ($k = 1, \dots, m$) dihitung. δ_k digunakan untuk menyalurkan *error* pada neuron *output* Y_k kembali ke semua neuron di lapisan sebelumnya (neuron tersembunyi terhubung ke Y_k). Itu juga digunakan (nanti) untuk memperbaiki bobot diantara lapisan *output* dan lapisan tersembunyi. Dengan cara yang sama, faktor δ_j ($j = 1, \dots, p$) dihitung untuk

setiap neuron tersembunyi Z_j . Itu tidak perlu untuk menyebarkan *error* kembali ke lapisan *input*, tetapi δ_j digunakan untuk memperbaiki bobot-bobot diantara lapisan tersembunyi dan lapisan *input*.

Menurut Fausett (1994) setelah semua faktor δ telah ditentukan, bobot-bobot untuk semua lapisan disesuaikan secara bersamaan. Penyesuaian bobot w_{kj} (dari neuron tersembunyi Z_j ke neuron *output* Y_k) didasarkan pada faktor δ_k dan aktivasi z_j dari neuron tersembunyi Z_j . Penyesuaian bobot v_{ji} (dari neuron *input* X_i ke neuron tersembunyi Z_j) didasarkan pada faktor δ_j dan aktivasi x_i dari neuron *input* X_i .

Fungsi aktivasi yang digunakan dalam penelitian ini memiliki beberapa karakteristik penting, yaitu: harus kontinu, terdiferensiasi, dan monoton tidak menurun. Selanjutnya, untuk efisiensi komputasi, diharapkan turunannya mudah dikomputasi. Untuk fungsi aktivasi yang paling banyak digunakan, nilai

dari turunan (pada nilai tertentu dari variabel dependen) dapat dinyatakan dalam bentuk nilai fungsi (pada nilai dari variabel independen). Salah satu fungsi aktivasi yang paling umum adalah fungsi sigmoid biner yang memiliki *range* (0,1) dan didefinisikan sebagai (Fausett, 1994: 292-293).

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

dengan: $f'(x) = f(x)[1 - f(x)]$.

Fungsi aktivasi yang umum lainnya adalah sigmoid bipolar yang memiliki *range* (-1,1) dan didefinisikan sebagai:

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

dengan: $f'(x) = \frac{1}{2}[1 + f(x)][1 - f(x)]$.

Catatan bahwa fungsi sigmoid bipolar terkait erat dengan fungsi

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad 3)$$

Menurut Siang (2004) fungsi sigmoid memiliki nilai maksimum sama dengan 1. Maka untuk pola targetnya lebih dari 1, pola *input* dan *output* harus terlebih dahulu ditransformasi sehingga semua polanya memiliki *range* yang sama seperti sigmoid yang dipakai. Alternatif lain adalah menggunakan fungsi aktivasi sigmoid hanya pada lapisan yang bukan lapisan *output*. Fungsi aktivasi yang dipakai pada lapisan *output* adalah fungsi identitas dan didefinisikan sebagai.

$$f(x) = x \quad 4)$$

Kusumadewi (2004) menyatakan bahwa salah satu fungsi aktivasi yang didefinisikan pada persamaan 4 dapat digunakan dalam algoritma BPNN umumnya yang akan diberikan pada algoritma pelatihan BPNN. Bentuk data (terutama nilai target) merupakan faktor penting dalam memilih fungsi yang sesuai. Pemilihan bobot awal juga sangat memengaruhi NN dalam mencapai minimum global atau lokal terhadap nilai *error*, serta cepat tidaknya proses pelatihan menuju kekonvergenan. Apabila nilai bobot awal terlalu besar, maka *input* ke setiap lapisan tersembunyi atau lapisan *output* akan jatuh pada daerah dimana turunan fungsi

sigmoidnya akan sangat kecil. Sebaliknya, apabila nilai bobot awal terlalu kecil, maka *input* ke setiap lapisan tersembunyi atau lapisan *output* akan sangat kecil, yang akan menyebabkan proses pelatihan akan berjalan sangat lambat. Biasanya bobot awal diinisialisasi secara acak dengan nilai antara -0,5 sampai 0,5 atau -1 sampai 1.

Menurut Fauset (1994) algoritma pelatihan dari BPNN mempunyai sintaks sebagai berikut.

Langkah 0. Inisialisasi bobot (tetapkan untuk nilai acak kecil).

Langkah 1. Selama kondisi berhenti salah, lakukan langkah 2-9.

Langkah 2. Untuk setiap pasang pelatihan, lakukan langkah 3-8.

Feedforward:

Langkah 3. Setiap neuron *input* ($X_i, i = 1, 2, \dots, n$) menerima sinyal *input* x_i dan menyebarkan sinyal tersebut ke semua neuron pada lapisan atas (neuron tersembunyi).

Langkah 4. Setiap neuron tersembunyi ($Z_j, j = 1, 2, \dots, p$) menjumlahkan bobot sinyal *input*,

$$z_in_j = v_{j0} + \sum_{i=1}^n x_i v_{ji} \quad 5)$$

menerapkan fungsi aktivasi untuk menghitung sinyal *output*,

$$z_j = f(z_in_j) \quad 6)$$

dan mengirimkan sinyal tersebut ke semua neuron di lapisan atas (neuron *output*).

Langkah 5. Setiap neuron *output* ($Y_k, k = 1, 2, \dots, m$) menjumlahkan bobot sinyal *input*,

$$y_in_k = w_{k0} + \sum_{j=1}^p z_j w_{kj} \quad 7)$$

dan menerapkan fungsi aktivasi untuk menghitung sinyal *output*,

$$y = f(y_in_k) \quad 8)$$

Backpropagation of error:

Langkah 6. Setiap neuron *output* ($Y_k, k = 1, 2, \dots, m$) menerima sebuah pola target yang sesuai dengan pola *input* pelatihan, hitung informasi *error*,

$$\delta_k = (t_k - y_k) f'(y_{in_k}) \quad 9)$$

hitung koreksi bobot (yang nanti digunakan untuk memperbaiki w_{kj}),

$$\Delta w_{kj} = \alpha \delta_k z_j \quad 10)$$

hitung koreksi bias (yang nanti digunakan untuk memperbaiki w_{k0}),

$$\Delta w_{k0} = \alpha \delta_k \quad 11)$$

dan mengirimkan δ_k ke neuron pada lapisan bawah.

Langkah 7. Setiap neuron tersembunyi ($Z_j, j = 1, 2, \dots, p$) menjumlahkan *input* delta dari neuron pada lapisan atas (lapisan *output*).

$$\delta_{in_j} = \sum_{i=1}^m \delta_i w_{kj} \quad 12)$$

Kalikan dengan turunan dari fungsi aktivasi untuk menghitung informasi *error*,

$$\delta_j = \delta_{in_j} f'(z_{in_j}) \quad 13)$$

hitung koreksi bobot (yang nanti digunakan untuk memperbaiki v_{ji}),

$$\Delta v_{ji} = \alpha \delta_j x_i \quad 14)$$

dan hitung koreksi bias (yang nanti digunakan untuk memperbaiki v_{j0}),

$$\Delta v_{j0} = \alpha \delta_j \quad 15)$$

Memperbaiki bobot dan bias:

Langkah 8. Setiap neuron *output* ($Y_k, k = 1, 2, \dots, m$) memperbaiki bias dan bobot ($j = 0, \dots, p$):

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad 16)$$

$$w_{k0}(\text{baru}) = w_{k0}(\text{lama}) + \Delta w_{k0} \quad 17)$$

Setiap neuron tersembunyi ($Z_j, j = 1, 2, \dots, p$) memperbaiki bias dan bobot ($i = 0, \dots, n$):

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad 18)$$

$$v_{j0}(\text{baru}) = v_{j0}(\text{lama}) + \Delta v_{j0} \quad 19)$$

Langkah 9. Kondisi tes berhenti.

Keterangan:

z_{in_j} : sinyal *input* yang diterima oleh neuron tersembunyi dari neuron *input*

v_{j0} : bobot bias pada lapisan tersembunyi, $j = 1, 2, \dots, p$

v_{ji} : bobot dari neuron *input* ke- i menuju neuron tersembunyi ke- j

z_j : sinyal *output* yang telah diaktivasi pada neuron tersembunyi

y_{in_k} : sinyal *input* yang diterima oleh neuron *output* dari neuron tersembunyi

w_{kj} : bobot dari neuron tersembunyi ke- j menuju neuron *output* ke- k

w_{k0} : bobot bias pada lapisan *output*, $k = 1, 2, \dots, m$

y_k : sinyal *output* yang telah diaktivasi pada neuron *output*

t_k : target yang ingin dicapai

α : laju percepatan

Bentuk umum model BPNN secara sistematis dapat dituliskan sebagai berikut.

$$y_t = f \left(\sum_{j=1}^p w_{kj} \cdot f \left(v_{j0} + \sum_{i=1}^n X_i v_{ji} \right) + w_{k0} \right) \quad 20)$$

dengan:

y_t : variabel *output*

w_{kj} : bobot dari neuron tersembunyi ke- j menuju neuron *output* ke- k

v_{j0} : bobot bias pada lapisan tersembunyi, $j = 1, 2, \dots, p$

X_i : neuron *input* ke- i , $i = 1, 2, \dots, n$

v_{ji} : bobot dari neuron *input* ke- i menuju neuron tersembunyi ke- j

w_{k0} : bobot bias pada lapisan *output*, $k = 1, 2, \dots, m$

Pengembangan analisis sintesis yang dihasilkan menggunakan prosedur jaringan *Backpropagation Neural Network* (BPNN) beserta langkah-langkah pemodelannya disajikan dalam beberapa langkah berikut.

1. Penentuan *Input* dan Target Jaringan

Prosedur awal dalam menggunakan BPNN adalah penentuan *input* dan target jaringan. Penentuan *input* dan target jaringan dalam skripsi ini dilakukan dengan melihat lag-

lag yang signifikan pada plot PACF dari masing-masing variabel. Plot PACF menunjukkan lag-lag yang signifikan, yaitu lag 2, lag 5, lag 9, lag 10, lag 11, sehingga jaringan yang akan dibangun memiliki 9 *input*, yaitu x_{t-2} , x_{t-5} , x_{t-9} , x_{t-10} , x_{t-11} dengan target x_t atau dapat dikatakan bahwa x_t

2. Pembagian Data

Pembagian data dilakukan dengan membagi data menjadi dua bagian, yaitu bagian pertama untuk data latih (*training*) yang digunakan pada tahap pelatihan jaringan dan bagian kedua untuk data uji (*testing*) yang digunakan pada tahap pengujian. Komposisi pembagian data yang akan digunakan untuk membangun jaringan BPNN untuk identifikasi publikasi dosen sebesar 80% untuk data latih dan 20% untuk data uji.

3. Normalisasi Data

Data *input* dan target pada data latih harus dinormalisasikan terlebih dahulu. Perintah untuk menormalisasi data *input* dan target pada data latih sebagai berikut.

```
[pn,ps] = mapstd(inputlatih)
```

```
[tn,ts] = mapstd(targetlatih)
```

4. Tahap Pembentukan Model

Perintah untuk *membangun* jaringan BPNN untuk peramalan penjualan semen, yaitu: `net = newff(minmax(pn), [n 1], {'tansig' 'purelin'}, 'traingdx');`

Perintah `newff` digunakan untuk membangun jaringan BPNN dengan n neuron pada satu lapisan tersembunyi dan satu neuron pada satu lapisan *output*. Perintah {'tansig' 'purelin'} menunjukkan fungsi aktivasi yang digunakan dengan `tansig` pada lapisan tersembunyi dan `purelin` pada lapisan *output*.

Perintah `traingdx` menunjukkan algoritma pelatihan yang digunakan.

5. Denormalisasi Data

Setelah proses pelatihan selesai dilakukan, maka hasil *output* jaringan yang ternormalisasi dikembalikan lagi seperti semula yang disebut dengan denormalisasi data. Perintah untuk mendenormalisasi *output* pada tahap pelatihan sebagai berikut.

```
a = mapstd('reverse',an,ts)
```

Perintah untuk mendenormalisasi *output* pada tahap pengujian sebagai berikut.

```
anew = mapstd('reverse',anewn,ts)
```

6. Tahap Penentuan Model

Tahap penentuan model terdiri dari penentuan model terbaik dan uji kesesuaian model. Setelah dilakukan pelatihan jaringan akan terbentuk berbagai macam model jaringan yang telah dihasilkan melalui tahap pembentukan model. Model terbaik terbentuk berdasarkan *trial and error* terhadap beberapa macam arsitektur jaringan. Model terbaik adalah model dengan MSE dan MAPE terkecil dengan jaringan paling sederhana. Berikut adalah hasil pelatihan jaringan BPNN dengan menggunakan algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* dan algoritma pelatihan *Levenberg-Marquardt*.

Pelatihan jaringan menggunakan algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* (`traingdx`). Berikut akan ditampilkan hasil pelatihan pada BPNN berdasarkan model terbaik dari masing-masing momentum dan LR.

Tabel 1. Hasil Pelatihan BPNN dengan Momentum = 0,1 dan LR = 0,01

Arsitektur Jaringan	$f(x)$	Epoch	Perf	Pelatihan			Pengujian		
				MSE	MAPE (%)	R	MSE	MAPE (%)	R
9-2-1	Logsig	5000	0,0288	4,1452e+09	2,8196	0,98537	4,3113e+10	7,6157	0,82283

Berdasarkan Tabel 1 dapat dilihat bahwa model terbaik BPNN pada algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* dengan momentum = 0,1

dan LR = 0,01 diperoleh pada arsitektur jaringan 9-2-1 pada fungsi aktivasi `logsig` dengan MSE dan MAPE pengujian masing-masing sebesar 4,3113e+10 dan 7,6157%.

Tabel 2. Hasil Pelatihan BPNN dengan Momentum = 0,1 dan LR = 0,02

Arsitektur Jaringan	$f(x)$	Epoch	Perf	Pelatihan			Pengujian		
				MSE	MAPE (%)	R	MSE	MAPE (%)	R
9-2-1	Logsig	5000	0,02	4,1972e	2,9810	0,985	5,3712e	8,5527	0,826
			92	+09		17	+10		81

Berdasarkan Tabel 2 dapat dilihat bahwa model terbaik BPNN pada algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* dengan momentum = 0,1

dan LR = 0,02 diperoleh pada arsitektur jaringan 9-2-1 pada fungsi aktivasi logsig dengan MSE dan MAPE pengujian masing-masing sebesar 5,3712e+10 dan 8,5527%.

Tabel 3. Hasil Pelatihan BPNN dengan Momentum = 0,1 dan LR = 0,03

Arsitektur Jaringan	$f(x)$	Epoch	Perf	Pelatihan			Pengujian		
				MSE	MAPE (%)	R	MSE	MAPE (%)	R
9-2-1	Logsig	5000	0,02	3,9427e	2,7371	0,986	5,6229e	8,5507	0,778
			74	+09		1	+10		91

Berdasarkan Tabel 3 dapat dilihat bahwa model terbaik BPNN pada algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* dengan momentum = 0,1

dan LR = 0,03 diperoleh pada arsitektur jaringan 9-2-1 pada fungsi aktivasi logsig dengan MSE dan MAPE pengujian masing-masing sebesar 5,6229e+10 dan 8,5507%.

Tabel 4. Hasil Pelatihan BPNN dengan Momentum = 0,1 dan LR = 0,1

Arsitektur Jaringan	$f(x)$	Epoch	Perf	Pelatihan			Pengujian		
				MSE	MAPE (%)	R	MSE	MAPE (%)	R
9-10-1	Logsig	5000	0,04	6,1582e	3,5523	0,978	6,0151e	9,1318	0,796
			29	+09		17	+10		64

Berdasarkan Tabel 4 dapat dilihat bahwa model terbaik BPNN pada algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* dengan momentum = 0,1 dan LR = 0,1 diperoleh pada arsitektur jaringan 9-10-1 pada fungsi aktivasi logsig dengan MSE

dan MAPE pengujian masing-masing sebesar 6,0151e+10 dan 9,1318%.

Berikut akan ditampilkan hasil pelatihan pada BPNN berdasarkan model terbaik dari masing-masing Mu.

Tabel 5. Hasil Pelatihan BPNN dengan Mu = 0,001

Arsitektur Jaringan	$f(x)$	Epoch	Perf	Pelatihan			Pengujian		
				MSE	MAPE (%)	R	MSE	MAPE (%)	R
9-10-1	Logsig	12	0,00	1,0884e	0,4077	0,999	1,0762e	11,6650	0,701
			0757	+08		81	+11		41

Berdasarkan Tabel 5 dapat dilihat bahwa model terbaik BPNN pada algoritma pelatihan

Levenberg-Marquardt dengan Mu = 0,001 diperoleh pada arsitektur jaringan 9-20-1 pada

fungsi aktivasi logsig dengan MSE dan MAPE dan 11,6650%.
pengujian masing-masing sebesar 1,0762e+11

Tabel 6. Hasil Pelatihan BPNN dengan Mu = 0,002

Arsitektur Jaringan	$f(x)$	Epoch	Perf	Pelatihan			Pengujian		
				MSE	MAPE (%)	R	MSE	MAPE (%)	R
9-10-1	Tansig	15	0,00	8,2889e	0,3857	0,999	4,7038e	8,1636	0,801
			0577	+07		71	+10		27

Berdasarkan Tabel 6 dapat dilihat bahwa model terbaik BPNN pada algoritma pelatihan *Levenberg-Marquardt* dengan Mu = 0,002 diperoleh pada arsitektur jaringan 9-20-1 pada

fungsi aktivasi tansig dengan MSE dan MAPE pengujian masing-masing sebesar 4,7038e+10 dan 8,1636%.

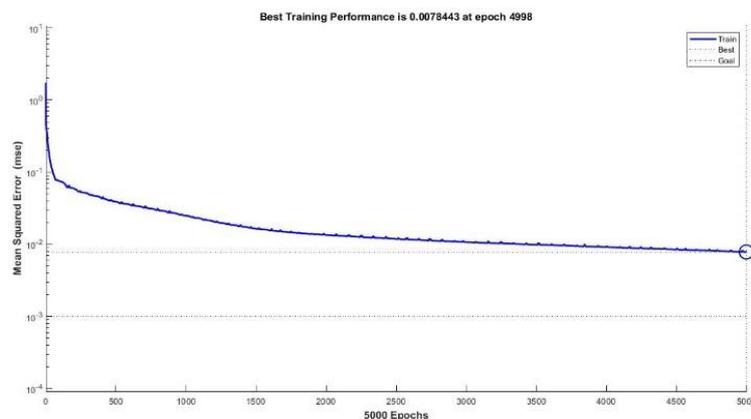
Tabel 7. Hasil Pelatihan BPNN dengan Mu = 0,003

Arsitektur Jaringan	$f(x)$	Epoch	Perf	Pelatihan			Pengujian		
				MSE	MAPE (%)	R	MSE	MAPE (%)	R
9-20-1	Tansig	10	3,32	4,7675e	0,1008	0,999	9,2593e	11,6191	0,762
			e-05	+06		98	+10		04

Berdasarkan Tabel 7 dapat dilihat bahwa model terbaik BPNN pada algoritma pelatihan *Levenberg-Marquardt* dengan Mu = 0,003 diperoleh pada arsitektur jaringan 9-20-1 pada fungsi aktivasi tansig dengan MSE dan MAPE pengujian masing-masing sebesar 9,2593e+10 dan 11,6191%.

Hasil pelatihan dan analisis numerik memberikan bahwa model terbaik BPNN pada algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* diperoleh model BPNN terbaik dengan algoritma pelatihan *gradient descent* dengan

momentum dan *adaptive learning rate* pada arsitektur jaringan 9-10-1 dengan momentum = 0,8 dan LR = 0,1 pada fungsi aktivasi tansig. Model BPNN 9-10-1 memperoleh hasil *epoch* ke 5000 yang memberikan nilai MSE dan MAPE pengujian masing-masing sebesar 3,6069e+10 dan 6,7670%. Iterasi berhenti pada *epoch* ke 5000 meskipun target *error* yang diinginkan (Goal = 0,001) belum tercapai. Model ini pada *epoch* ke 5000 telah mencapai konvergen dengan *performance* = 0,00784. Proses pelatihan pada setiap *epoch* ditunjukkan pada Gambar 2



Gambar 2. Proses Pelatihan BPNN dengan Algoritma Pelatihan *Gradient Descent*

Proses identifikasi menggunakan model BPNN terbangun dari 5 neuron pada lapisan *input*, 4 neuron pada lapisan tersembunyi, dan 1 neuron pada lapisan *output*. Diperoleh nilai *input* $x_{t-1} = x_1 = 0,251$, $x_{t-2} = x_2 = 0,247$, $x_{t-3} = x_3 = 0,339$, $x_{t-4} = x_4 = 0,153$. Selanjutnya *input* dinormalisasikan menjadi

$$\begin{aligned}
 & [x_1 \cdot v_{11}] [x_1 \cdot v_{21}] [x_1 \cdot v_{31}] [x_1 \cdot v_{41}] [x_1 \cdot v_{51}] \\
 & = [1,4579 \cdot 5,0754] [1,4579 \cdot 8,0478] [1,4579 \cdot 14,9385] [1,4579 \cdot 91,5653] \\
 & [1,4579 \cdot -0,6292] \\
 & = [7,3994] [11,7329] [21,7788] [21,7788] [133,4931] [-0,9173] \\
 & [x_2 \cdot v_{12}] [x_2 \cdot v_{22}] [x_2 \cdot v_{32}] [x_2 \cdot v_{42}] [x_2 \cdot v_{52}] \\
 & = [0,3642 \cdot -4,0790] [0,3642 \cdot -11,2145] [0,3642 \cdot -26,7816] \\
 & [0,3642 \cdot -125,5131] [0,3642 \cdot 2,2996] \\
 & = [-1,4856] [-4,0843] [-9,7539] [-45,7119] [0,8375] \\
 & [x_3 \cdot v_{13}] [x_3 \cdot v_{23}] [x_3 \cdot v_{33}] [x_3 \cdot v_{43}] [x_3 \cdot v_{53}] \\
 & = [2,4609 \cdot -0,1994] [2,4609 \cdot 4,7796] [2,4609 \cdot -160,8956] [2,4609 \cdot -84,4353] \\
 & [2,4609 \cdot -0,3413] \\
 & = [-0,4907] [11,7621] [-395,9480] [-207,7868] [-0,8399] \\
 & [x_4 \cdot v_{14}] [x_4 \cdot v_{24}] [x_4 \cdot v_{34}] [x_4 \cdot v_{44}] [x_4 \cdot v_{54}] \\
 & = [1,7404 \cdot 3,6101] [1,7404 \cdot -6,8196] [1,7404 \cdot -2,5029] [1,7404 \cdot 119,1507] \\
 & [1,7404 \cdot 0,0139] \\
 & = [6,2830] [-11,8688] [-4,3560] [207,3699] [0,0242]
 \end{aligned}$$

Diperoleh,

$$\begin{aligned}
 \sum_{i=1}^4 x_i \cdot v_{1i} &= x_1 \cdot v_{11} + x_2 \cdot v_{12} + x_3 \cdot v_{13} + x_4 \cdot v_{14} + x_5 \cdot v_{15} + x_7 \cdot v_{17} \\
 &+ x_8 \cdot v_{18} + x_9 \cdot v_{19} \\
 &= 7,3994 + (-1,4856) + (-0,4907) + 6,2830 + (-0,1747) \\
 &+ 2,7350 + 1,7317 + 1,1482 + (-1,6437) \\
 &= 15,5027 \\
 \sum_{i=1}^9 x_i \cdot v_{2i} &= x_1 \cdot v_{21} + x_2 \cdot v_{22} + x_3 \cdot v_{23} + x_4 \cdot v_{24} + x_5 \cdot v_{25} + x_6 \cdot v_{26} + x_7 \cdot v_{27} \\
 &+ x_8 \cdot v_{28} + x_9 \cdot v_{29} \\
 &= 11,7329 + (-4,0843) + 11,7621 + (-11,8688) + (-0,1688) \\
 &+ (-26,0917) + 20,2088 + 6,8734 + (-2,5648) \\
 &= 5,7987 \\
 \sum_{i=1}^9 x_i \cdot v_{3i} &= x_1 \cdot v_{31} + x_2 \cdot v_{32} + x_3 \cdot v_{33} + x_4 \cdot v_{34} + x_5 \cdot v_{35} + x_6 \cdot v_{36} + x_7 \cdot v_{37} \\
 &+ x_8 \cdot v_{38} + x_9 \cdot v_{39} \\
 &= 21,7788 + (-9,7539) + (-395,9480) + (-4,3560) + (-0,2663) \\
 &+ (-84,3248) + 23,6936 + 22,5261 + 116,9036 \\
 &= -309,7468 \\
 \sum_{i=1}^9 x_i \cdot v_{4i} &= x_1 \cdot v_{41} + x_2 \cdot v_{42} + x_3 \cdot v_{43} + x_4 \cdot v_{44} + x_5 \cdot v_{45} + x_6 \cdot v_{46} + x_7 \cdot v_{47}
 \end{aligned}$$

$x_1 = 1,4579$, $x_2 = 0,3642$, $x_3 = 2,4609$, $x_4 = 1,7404$. Berdasarkan model BPNN yang terbentuk maka perhitungan identifikasi publikasi karya ilmiah dosen sebagai berikut.

Perhitungan pada lapisan *input* ke lapisan tersembunyi sebagai berikut.

$$\begin{aligned}
& +x_8 \cdot v_{48} + x_9 \cdot v_{49} \\
& = 133,4931 + (-45,7119) + (-207,7868) + 207,3699 + 0,6338 \\
& +109,4990 + (-18,7779) + 107,1429 + 187,8776 \\
& = 473,7395 \\
& \sum_{i=1}^9 x_i \cdot v_{5i} = x_1 \cdot v_{51} + x_2 \cdot v_{52} + x_3 \cdot v_{53} + x_4 \cdot v_{54} + x_5 \cdot v_{55} + x_6 \cdot v_{56} + x_7 \cdot v_{57} \\
& +x_8 \cdot v_{58} + x_9 \cdot v_{59} \\
& = (-0,9173) + 0,8375 + (-0,8399) + 0,0242 + 0,0191 + 1,3453 \\
& +0,3173 + (-1,9375) + (-0,7156) \\
& = -1,8668
\end{aligned}$$

Operasi keluaran pada lapisan input ke lapisan tersembunyi sebagai berikut.

$$\begin{aligned}
z_{in_1} &= v_{10} + \sum_{i=1}^9 x_i v_{1i} = -8,1217 + 15,5027 = 7,3810 \\
z_{in_2} &= v_{20} + \sum_{i=1}^9 x_i v_{2i} = 4,8887 + 5,7987 = 10,6874 \\
z_{in_3} &= v_{30} + \sum_{i=1}^9 x_i v_{3i} = -42,4552 + (-309,7468) = -352,2020 \\
z_{in_4} &= v_{40} + \sum_{i=1}^9 x_i v_{4i} = 115,7330 + 473,7396 = 589,4725 \\
z_{in_5} &= v_{50} + \sum_{i=1}^9 x_i v_{5i} = -1,6325 + (-1,8668) = -3,4993
\end{aligned}$$

Dengan menggunakan fungsi aktivasi sigmoid biner, diperoleh

$$\begin{aligned}
z_1 &= \frac{1}{1 + e^{-z_{in_1}}} = \frac{1}{1 + e^{-7,3810}} = 0,9994 \\
z_2 &= \frac{1}{1 + e^{-z_{in_2}}} = \frac{1}{1 + e^{-10,6874}} = 2,2830e - 05 \\
z_3 &= \frac{1}{1 + e^{-z_{in_3}}} = \frac{1}{1 + e^{-(-352,2020)}} = 1,0980e - 153 \\
z_4 &= \frac{1}{1 + e^{-z_{in_4}}} = \frac{1}{1 + e^{-589,4725}} = 1 \\
z_5 &= \frac{1}{1 + e^{-z_{in_5}}} = \frac{1}{1 + e^{-(-3,4993)}} = 0,0293
\end{aligned}$$

Perhitungan pada lapisan tersembunyi ke lapisan output sebagai berikut.

$$\begin{aligned}
\sum_{j=1}^5 w_{1j} \cdot z_j &= w_{11} \cdot z_1 + w_{12} \cdot z_2 + w_{13} \cdot z_3 + w_{14} \cdot z_4 + w_{15} \cdot z_5 \\
&= (1,5583 \cdot 0,9994) + (1,1092 \cdot 2,2830e - 05) \\
&+ ((-0,6066) \cdot 1,0980e - 153) + (0,6380 \cdot 1) + (2,6156 \cdot 0,0293) \\
&= 1,5574 + 2,5323e - 05 - 6,6605e - 154 + 0,6380 + 0,0766 \\
&= 2,2720
\end{aligned}$$

Operasi keluaran pada lapisan tersembunyi ke lapisan output sebagai berikut.

$$y_{in_1} = w_{10} + \sum_{j=1}^5 w_{1j} \cdot z_j = -1,5981 + 2,2720 = 0,6739$$

Dengan menggunakan fungsi aktivasi linear, diperoleh

$$y_t = 0,6739$$

Tabel 9. Hasil identifikasi pada publikasi dosen FMIPA UNNES

Variabel	Persentase (dalam %)
X ₁	23,75
X ₂	21,23
X ₃	19,02
X ₄	31,0

Hasil identifikasi publikasi dosen menggunakan BPNN dan analisis terhadap pelatihan di atas diperoleh bahwa publikasi dosen dengan karya jurnal bereputasi internasional sebanyak 23,75%, untuk jurnal internasional sebanyak 21,23%, dan jurnal nasional terakreditasi sebesar 19,02% dan karya publikasi dalam prosiding internasional sebanyak 31,0% (lihat Tabel 9). Berdasarkan perolehan data pada Tabel 9 maka hasil identifikasi ini memberikan bahwa jumlah persentase terbanyak untuk publikasi dosen adalah publikasi dengan jenis prosiding internasional sebanyak 31,0%.

SIMPULAN

Berdasarkan hasil penelitian dan hasil pembahasan di bagian sebelumnya, maka disimpulkan bahwa model identifikasi publikasi dosen dalam mewujudkan internasionalisasi UNNES menggunakan BPNN berdasarkan hasil pelatihan dan analisis numerik memberikan bahwa model terbaik BPNN pada algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* diperoleh model BPNN terbaik dengan algoritma pelatihan *gradient descent* dengan momentum dan *adaptive learning rate* pada arsitektur jaringan 9-10-1 dengan momentum = 0,8 dan LR = 0,1 pada fungsi aktivasi tansig. Hasil identifikasi publikasi dosen menggunakan BPNN dan analisis terhadap pelatihan di atas diperoleh bahwa publikasi dosen dengan karya jurnal bereputasi internasional sebanyak 23,75%, untuk jurnal internasional sebanyak 21,23%, dan jurnal nasional terakreditasi sebesar 19,02% dan

karya publikasi dalam prosiding internasional sebanyak 31,0%.

Sedangkan saran yang diberikan dalam penelitian, bisa dikembangkan dengan menggunakan fungsi aktivasi dan bobot awal bias yang berbeda sehingga akan memberikan nilai dan hasil yang berbeda.

DAFTAR PUSTAKA

- Bishop CM .1995. *Neural Networks for Pattern Recognition*. Birmingham: Oxford University Press.
- Beale & Finlay .1992. Testing for Neglected Nonlinearity in Time Series Models. A comparison of Neural Network methods and alternative test. *Journal of Econometrics* 56:269-290.
- Chen JF .1995. Analysis of an adaptive time-series autoregressive moving-average (ARMA) model for short-term load forecasting. *Electric Power System Research*, 34: 187-196.
- Fausett L .1994. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. New Jersey: Prentice-Hall.
- Fine TL .1999. *Feedforward Neural Network Methodology*. Springer: New York.
- Hardianto HNI, Suyanto & Purnama B .2011. Analisis dan Implementasi *Differential Evolution* dan *Recurrent Neural Network* untuk Prediksi Data *Time Series* Studi Kasus Kurs Jual Emas. *Tugas Akhir*. Universitas Telkom.
- Haykin H .1999. *Neural Networks: A Comprehensive Foundation*, 2nd edition. Prentice-Hall: Oxford.
- Husen W .2001. Forecasting of Maximum ShortTerm Electricity Consumption by Using Elman-Recurrent Neural Network. Unpublished Final Project, Department of Physics Engineering, Institut Teknologi Sepuluh Nopember: Indonesia.
- Kusumadewi F .2014. Peramalan Harga Emas Menggunakan *Feedforward Neural Networks* Dengan Algoritma *Backpropagation*. *Skripsi*. Universitas Negeri Yogyakarta.
- Kusumadewi S .2004. *Membangun Jaringan Syaraf Tiruan: menggunakan MATLAB & Excel Link*. Yogyakarta: Graha Ilmu.
- Ren, C., An, N., Wang, J., Li, L., Hu, B., & Shang, D. 2013. "Optimal Parameters Selection for BP Neural Network Based on Particle Swarm

- Optimization: A Case Study of Wind Speed Forecasting". *Knowledge-Based Systems*.
- Ripley BD .1996. Pattern Recognition and Neural Networks. Cambridge University Press: Cambridge.
- Ristiana Y .2008. Autoregressive Neural Network Model (ARNN) for Forecasting Short Term Electricity Consumption at PT. PLN Gresik. Unpublished Final Project, Department of Physics Engineering, Institut Teknologi Sepuluh Nopember, Indonesia.
- Siang JJ .2004. Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab. Penerbit Andi: Yogyakarta.
- Swanson & White H .1995. Learning in Artificial Neural Networks: A statistical Perspective. *Neural Computation*, Vol. 1, pp: 425-464.
- Taylor JW, Menezes LM & Mc Sharry PE .2006. A comparison of univariate methods for forecasting electricity demand up to a day ahead. *International Journal of Forecasting*, 22: 1-16.
- Taylor JW .2003. Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society*, 54: 799-805.
- Walid, Subanar, Dedi R, Suhartono .2012. Recurrent Neural Network (RNN) pada Beban Listrik di Semarang. *Prosiding Seminar Nasional Matematika*. Unnes.
- Walid, Subanar, Dedi R, Suhartono .2015. Fractional Integrated Recurrent Neural Network (FIRNN) for Forecasting of Time Series Data with Applications in Electric Load in Java-Bali. *Contemporary Engineering Sciences*, Vol. 8, no. 31: 1485 – 1500.
- Walid, Sugiman, Sunarmi, Dian TW .2018. Analisis Produktivitas Kinerja Dosen dan Tenaga Kependidikan dalam Mewujudkan Tahun Reputasi Universitas Negeri Semarang (UNNES) Menggunakan Jaringan Saraf Tiruan, *PRISMA 1* vol 1, no.1: 919-927.
- Wang L, Wang ZG & Liu S .2018. Optimal Forecast Combination Based on Neural Networks for Time Series Forecasting. *Applied Soft Computing Journal*.