

# ANALISIS NEURAL NETWORK STRUKTUR BACKPROPAGATION SEBAGAI METODE PERAMALAN PADA PERHITUNGAN TINGKAT KEMISKINAN DI INDONESIA

Astria Mulyani

Jurusan Teknik Informatika

STMIK Nusa Mandiri

Jl. Kramat Raya No. 18, Jakarta Pusat

astria.mulyani@nusamandiri.ac.id

**Abstract-** Poverty is a condition in which people lack ability in traditionally devoted their primary. (Boa,2008) defines "Structural poverty is poverty, suspected uncaused structure from the condition of structures, or unfavorable life order". The expected amount of poverty can be reduced.

In order to reduce the number of poverty to be known beforehand what factors are the cause of the poverty level is high or low. With the Backpropagation Neural Network structures for forecasting the calculation of the poverty level in Indonesia.

Based on the analysis carried out apparently backpropagation neural network method yields more accurate in forecasting the calculation of the poverty level in Indonesia because these methods do training repeatedly to get the best models and can also be analyzed mathematically.

**Keywords:** backpropagation neural network, method, Poverty,

**Intisari-** Kemiskinan merupakan kondisi masyarakat dimana ketidak mampuannya dalam memenuhi kebutuhan primer mereka. (Boa, 2008) mendefinisikan "Kemiskinan struktural adalah kemiskinan yang ditengarai atau didalihkan bersebab dari kondisi struktur, atau tatanan kehidupan yang tak menguntungkan". Yang diharapkan jumlah kemiskinan dapat diturunkan.

Agar dapat menurunkan jumlah kemiskinan sebelumnya harus diketahui faktor-faktor apa saja yang menjadi sebab tingkat kemiskinan tinggi atau rendah. Dengan *Neural Network struktur Backpropagation* untuk peramalan pada perhitungan tingkat kemiskinan di Indonesia. Berdasarkan analisa yang dilakukan ternyata *neural network backpropagation* metode yang memberikan hasil yang lebih akurat dalam peramalan pada perhitungan tingkat kemiskinan di Indonesia karena metode ini melakukan training yang berulang-ulang untuk mendapatkan model terbaik dan juga bisa dianalisa secara matematika.

**Kata kunci:** Kemiskinan, Metode, *Neural network backpropagation*

## PENDAHULUAN

Dalam hasil Penelitian (Rohmatullah, 2007) dan kawan-kawan yang berjudul *Logika Fuzzy dan Jaringan Syaraf Tiruan untuk Peningkatan Mutu The Hitam* mengatakan "bahwa logika fuzzy dapat digunakan untuk menentukan parameter proses terpenting guna dijadikan faktor utama guna meningkatkan mutu teh hitam dan model jaringan syaraf tiruan sebagai alat yang dapat digunakan untuk prediksi pengawasan parameter proses secara otomatis".

Menurut (Septiani, 2005) dengan kawan-kawan dalam jurnal yang berjudul *Sistem Intelijen Prediksi dan Penilaian Kualitas Susu Pasteurisasi dengan menggunakan Logika Fuzzy dan Jaringan Syaraf Tiruan* menjelaskan bahwa " Jaringan Syaraf Tiruan menghasilkan prediksi suhu pasteurisasi untuk satu bulan yang akan datang"

Menurut Penelitian (M.Fuad, 2011) yang berjudul *Prediksi Ketersediaan Beras di Masyarakat dengan Menggunakan Logika Fuzzy dan Jaringan Syaraf Tiruan dalam Upaya Peningkatan Ketahanan Pangan* mengatakan bahwa "Logika Fuzzy dan Jaringan Syaraf Tiruan dapat digunakan dalam membangun system intelejen untuk memprediksi ketersediaan beras di Masyarakat".

Berdasarkan hasil penelitian-penelitian di atas, maka dalam penelitian ini penulis akan memprediksi atau peralaman dalam perhitungan tingkat kemiskinan di Indonesia dengan neural network struktur backpropagation.

## BAHAN DAN METODE

### 1. Neural network

Langkah-langkah yang harus dilakukan untuk pemodelan perkiraan adalah sebagai berikut (Kusumadewi dan Purnomo, 2010):

#### a. Preprocessing/ Normalisasi

Pada proses perkiraan menggunakan Backpropagation Neural Network, sebelum dilakukan pelatihan, data input dan target output harus dinormalisasi terlebih dahulu. Normalisasi adalah penskalaan terhadap nilai-nilai masuk ke dalam suatu range tertentu. Hal ini dilakukan agar nilai input dan target output sesuai dengan range dari fungsi aktivasi yang digunakan dalam jaringan.

Data input dan target output dinormalisasi dengan cara membawa data ke bentuk normal yang memiliki mean = 0 dan standar deviasi =1, berdasarkan metode Z-Score dengan formula sebagai berikut:

$$\text{Nilai Baru} = \frac{\text{Nilai Lama} - \text{Rata 2}}{\text{Standar Deviasi}}$$

kemudian nilai baru tersebut dikonversi dan dicari luasnya di bawah kurva dengan menggunakan kurva normal-Z.

#### b. Pemilihan Variabel Input

Pemilihan variabel dilakukan dengan mengeliminasi variabel yang tidak berguna dan mempertahankan variabel-variabel yang memberikan nilai korelasi yang cukup signifikan terhadap variabel output .

#### c. Perancangan Struktur Jaringan yang Optimum

Langkah selanjutnya adalah menentukan jumlah lapisan *input*, lapisan tersembunyi (*hidden layer*), dan lapisan *output*. Jumlah lapisan input

berdasarkan pada banyaknya data yang mempengaruhi perkiraan. Sedangkan banyaknya lapisan output adalah banyaknya hasil output perkiraan yang dicari.

d. Pemilihan Koefisien Pembelajaran (*Learning Rate*) dan *Momentum*

Besarnya *learning rate* akan berimplikasi pada besarnya langkah pembelajaran. *Momentum* dalam *neural network* adalah perubahan bobot yang didasarkan pada arah gradient pola terakhir dan pola sebelumnya. Pada pembangunan jaringan *Backpropagation* yang akan digunakan dalam perkiraan, hasil keputusan yang kurang memuaskan dapat diperbaiki dengan menggunakan *learning rate* dan *momentum* secara trial and error untuk mendapatkan nilai bobot yang optimum agar *mean square error* (MSE) dan *mean absolute percentage error* (MAPE) jaringan dapat diperbaiki.

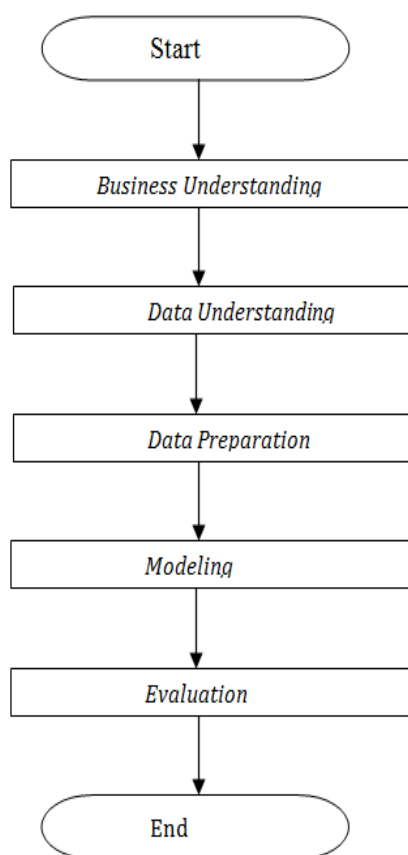
e. Pemilihan Struktur Jaringan yang Optimum dan Penggunaannya untuk Peramalan

Jaringan yang memiliki nilai MSE dan MAPE terendah dipilih sebagai jaringan yang optimum dan digunakan untuk perkiraan.

f. Postprocessing/Denormalisasi

Setelah proses pelatihan dan pengujian selesai, untuk mengembalikan nilai ternormalisasi output jaringan ke nilai yang sebenarnya, dilakukan proses denormalisasi atau postprocessing.

penelitian ini adalah semua pustaka dan laporan penelitian yang telah dilakukan baik dalam bentuk jurnal maupun kumpulan seminar nasional yang berhubungan dengan jaringan syaraf tiruan. Terdapat beberapa tahap dalam pengolahan data eksperimen, pada penelitian ini menggunakan model Cross-Standard Industry for Data Mining (CRISP-DM), yaitu seperti terlihat pada diagram di bawah ini: (Sumathi, 2006)



Sumber: (Sumathi, 2006)

Gambar 1. Diagram Tahap Penelitian

Business understanding adalah pemahaman tentang substansi dari kegiatan data mining yang akan dilakukan, kebutuhan dari perspektif bisnis. Kegiatannya antara lain: menentukan sasaran atau tujuan bisnis, memahami situasi bisnis, menentukan tujuan data mining, dan membuat perencanaan strategi serta jadwal penelitian. Dalam penelitian ini akan dilakukan *Neural Network Struktur Backpropagation* sebagai metode peramalan pada perhitungan tingkat kemiskinan di Indonesia.

Data Understanding adalah fase mengumpulkan data awal, mempelajari data untuk bisa mengenal data yang akan dipakai, mengidentifikasi masalah yang berkaitan dengan kualitas data, mendeteksi subset yang menarik dari data untuk membuat hipotesa awal. Pada fase ini akan ditentukan atribut-atribut yang digunakan untuk membangun model.

Data preparation sering disebut sebagai fase yang padat karya. Aktifitas yang dilakukan antara lain memilih table atau field sebagai bahan untuk data mining (set data mentah). Dalam penelitian ini menggunakan data online yang bersumber dari Badan Pusat Statistika, data tentang kemiskinan yang diperoleh untuk penelitian ini sebanyak 231 record, dimana 80% dari data tersebut sebagai data training, dan 20% dari data sebagai data testing. Untuk mendapatkan data yang berkualitas, beberapa teknik *preprocessing* digunakan, yaitu: (Vecellis, 2009)

1. *Data validation*, untuk mengidentifikasi dan menghapus data yang ganjil (*outlier/noise*), data yang tidak konsisten, dan data yang tidak lengkap (*missing value*)
2. *Data integration and transformation*, untuk meningkatkan akurasi dan efisiensi algoritma.
3. *data size reduction and discretization*, untuk memperoleh data set dengan jumlah atribut dan *record* yang lebih sedikit tetapi bersifat informatif.

## HASIL DAN PEMBAHASAN

Pada penelitian ini, data asli yang digunakan ada 231 record yang terbagi atas data training yang digunakan adalah data indeks kedalaman kemiskinan(P1) dan keparahan kemiskinan (P2) menurut propinsi periode maret 2007-maret 2012 sebanyak 198 record atau 80% dari jumlah data keseluruhan. Sedangkan data testing yang digunakan adalah data indeks kedalaman kemiskinan(P1) dan keparahan kemiskinan (P2) menurut propinsi periode september 2012 sebanyak 33 record atau 20% dari data keseluruhan. Agar dapat memberikan gambaran perubahan hasil pengolahan data, dibawah ini diberikan deskripsi data asli, karena terlalu banyak penulis hanya mengambil dari data testing sesuai data yang akan diuji atau di test keakuratan datanya seperti tertera dalam tabel di bawah ini:

**Tabel 1 Angka Indeks Kedalaman Kemiskinan(P1) dan Keperahan Kemiskinan(P2) Menurut Propinsi, September 2012**

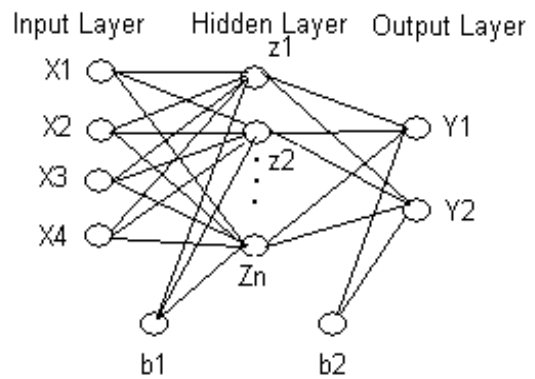
| No | Propinsi         | JPM(000)  |           | GK(Rp)  | P1(%)     |           | P2(%)     |           |
|----|------------------|-----------|-----------|---------|-----------|-----------|-----------|-----------|
|    |                  | Kota+Desa | Kota+Desa |         | Kota+Desa | Kota+Desa | Kota+Desa | Kota+Desa |
| 1  | Aceh             | 876.60    | 18.58     | 321,893 | 3.07      | 0.83      |           |           |
| 2  | Sumatera Utara   | 1,378.40  | 10.41     | 271,738 | 1.82      | 0.50      |           |           |
| 3  | Sumatera Barat   | 397.90    | 8.00      | 292,052 | 1.24      | 0.31      |           |           |
| 4  | Riau             | 481.30    | 8.05      | 310,603 | 1.13      | 0.25      |           |           |
| 5  | Jambi            | 270.10    | 8.28      | 273,267 | 1.37      | 0.44      |           |           |
| 6  | Sumatera Selatan | 1,042.00  | 13.48     | 259,668 | 1.85      | 0.43      |           |           |
| 7  | Bengkulu         | 310.50    | 17.51     | 283,252 | 3.05      | 0.80      |           |           |
| 8  | Lampung          | 1,219.00  | 15.65     | 263,088 | 2.53      | 0.62      |           |           |
| 9  | Bangka Belitung  | 70.20     | 5.37      | 382,412 | 0.66      | 0.14      |           |           |
| 10 | Kep Riau         | 131.20    | 6.83      | 363,450 | 0.85      | 0.19      |           |           |
| 11 | DKI Jakarta      | 366.80    | 3.70      | 392,571 | 0.56      | 0.15      |           |           |
| 12 | Jawa Barat       | 4,421.50  | 9.89      | 242,104 | 1.62      | 0.42      |           |           |
| 13 | Jawa Tengah      | 4,863.40  | 14.98     | 233,769 | 2.39      | 0.57      |           |           |
| 14 | DI Yogyakarta    | 562.10    | 15.88     | 270,110 | 2.89      | 0.75      |           |           |
| 15 | Jawa Timur       | 4,960.50  | 13.08     | 243,783 | 1.93      | 0.44      |           |           |
| 16 | Banten           | 648.30    | 5.71      | 251,161 | 0.95      | 0.28      |           |           |
| 17 | Bali             | 161.00    | 3.95      | 254,221 | 0.39      | 0.07      |           |           |
| 18 | NTB              | 828.30    | 18.02     | 248,758 | 3.20      | 0.83      |           |           |
| 19 | NTT              | 1,000.30  | 20.41     | 222,507 | 3.47      | 0.91      |           |           |
| 20 | KalBar           | 355.70    | 7.96      | 239,162 | 1.24      | 0.33      |           |           |
| 21 | KalTeng          | 141.90    | 6.19      | 277,407 | 1.08      | 0.27      |           |           |
| 22 | Kalsel           | 189.20    | 5.01      | 269,714 | 0.76      | 0.17      |           |           |
| 23 | KalTim           | 246.10    | 6.38      | 363,887 | 0.99      | 0.25      |           |           |
| 24 | SulUt            | 177.50    | 7.64      | 223,883 | 1.18      | 0.30      |           |           |
| 25 | SulTeng          | 409.60    | 14.94     | 266,718 | 2.82      | 0.82      |           |           |
| 26 | SulSel           | 805.90    | 9.82      | 195,627 | 1.68      | 0.42      |           |           |
| 27 | SulTengg         | 304.30    | 13.06     | 203,333 | 1.92      | 0.49      |           |           |
| 28 | Gorontalo        | 187.70    | 17.22     | 212,476 | 3.21      | 0.84      |           |           |
| 29 | SulBar           | 160.60    | 13.01     | 207,072 | 1.74      | 0.40      |           |           |
| 30 | Maluku           | 338.90    | 20.76     | 295,904 | 4.38      | 1.31      |           |           |
| 31 | Maluku Utara     | 88.30     | 8.06      | 250,184 | 0.85      | 0.14      |           |           |
| 32 | Papua Barat      | 223.20    | 27.04     | 354,626 | 5.71      | 1.71      |           |           |
| 33 | Papua            | 976.40    | 30.66     | 297,502 | 7.35      | 2.44      |           |           |

Sumber: (Badan Pusat Statistika,2012)

**Metode Neural Network Struktur**

**Backpropagation**

Proses pelatihan jaringan syaraf tiruan menggunakan 80% dari total data , pada proses ini akan dilakukan pelatihan dengan arsitektur neural network yang berbeda-beda. Setiap arsitektur neural network akan menghasilkan bobot pelatihan yang terakhir yang akan digunakan sebagai bobot awal saat melakukan pengujian. Berdasarkan tabel 4.1 arsitektur jaringan pada penelitian ini terdiri dari input layer sejumlah 4 neuron yaitu jumlah penduduk miskin X1, prosentase jumlah penduduk miskin X2, garis kemiskinan X3 dan propinsi X4, hidden layer sejumlah n neuron, output layer terdiri dari 2 neuron Indeks kedalaman kemiskinan Y1 dan indeks keparahan kemiskinan Y2. Berikut gambar arsitektur jaringan pada penelitian ini:



Sumber: (Kusrini,2009)

**Gambar 2 Arsitektur Jaringan**

Untuk setiap data pada data *training* dari Tabel 1, dihitung input untuk simpul berdasarkan nilai input dan jaringan saat itu. Bobot awal untuk *input layer*, *hidden layer*, dan bias diinisialisasi secara acak. Simpul bias terdiri dari dua, yaitu pada *input layer* yang terhubung dengan simpul-simpul pada *hidden layer*, dan pada *hidden layer* yang terhubung pada *output layer*. Setelah semua nilai awal diinisialisasi, kemudian dihitung masukan, keluaran, dan *error*. Selanjutnya membangun output untuk simpul menggunakan fungsi aktivasi sigmoid. Setelah didapat nilai dari fungsi aktivasi, hitung nilai *error* antara nilai yang diprediksi dengan nilai yang sesungguhnya. Setelah nilai *error* dihitung, selanjutnya dibalik ke *layer* sebelumnya (*backpropagated*). Nilai *Error* yang dihasilkan dari langkah sebelumnya digunakan untuk memperbarui bobot relasi. Begitu seterusnya diulangi lagi ke langkah awal sampai kondisi yang diinginkan tercapai.

Mencari network terbaik dilakukan dengan cara mengubah jumlah neuron/node pada lapisan tersembunyi secara trial and error, maka di dapatkan konfigurasi terbaik. Berikut tabel Mean Square Error(MSE) dari beberapa variasi jaringan yang dilatih menggunakan matlab versi 7.0 pada data training dengan maximal epoch(iterasi) = 5000, momentum = 0.8, dan target error = 0.01:

**Tabel 2 Hasil Pelatihan Arsitektur Jaringan**

| Learning Rate | Arsitektur Jaringan | MSE     |
|---------------|---------------------|---------|
| 0.5           | 4-3-2               | 0.56815 |
| 0.6           | 4-3-2               | 0.56802 |
| 0.7           | 4-3-2               | 0.56791 |
| 0.8           | 4-3-2               | 0.56781 |
| 0.9           | 4-3-2               | 0.56772 |
| 0.5           | 4-5-2               | 0.56816 |
| 0.6           | 4-5-2               | 0.56800 |
| 0.7           | 4-5-2               | 0.56787 |
| 0.8           | 4-5-2               | 0.56777 |
| 0.9           | 4-5-2               | 0.56769 |
| 0.5           | 4-10-2              | 0.56834 |
| 0.6           | 4-10-2              | 0.56810 |
| 0.7           | 4-10-2              | 0.56764 |

|            |               |                |
|------------|---------------|----------------|
| 0.8        | 4-10-2        | 0.56760        |
| 0.9        | 4-10-2        | 0.56750        |
| 0.5        | 4-15-2        | 0.56838        |
| 0.6        | 4-15-2        | 0.56794        |
| 0.7        | 4-15-2        | 0.56770        |
| 0.8        | 4-15-2        | 0.56756        |
| 0.9        | 4-15-2        | 0.56732        |
| 0.5        | 4-30-2        | 0.56838        |
| 0.6        | 4-30-2        | 0.56808        |
| 0.7        | 4-30-2        | 0.56780        |
| 0.8        | 4-30-2        | 0.56755        |
| <b>0.9</b> | <b>4-30-2</b> | <b>0.56730</b> |

Ket: 4-30-2 = neuron input layer = 4, neuron hidden layer = 30, dan neuron output layer = 2.

Berdasarkan tabel di atas, maka didapatkan konfigurasi jaringan terbaik sebagai berikut:

1. Jumlah neuron lapisan input = 4
2. Jumlah neuron lapisan tersembunyi = 30
3. Jumlah neuron lapisan output = 2

Sebagai sampel perhitungan metode neural network struktur backpropagation, kita akan menggunakan arsitektur jaringan 4-30-2, artinya jumlah neuron lapisan input sebanyak 4, jumlah neuron lapisan tersembunyi sebanyak 30, dan jumlah neuron lapisan output sebanyak 2. Sebagai kondisi berhenti adalah nilai ambang *Mean Square Error* (MSE) sebesar 0,01 atau maksimum iterasi sebesar 5000 epoch. *Learning rate* kita set 0.5. Selanjutnya nilai-nilai bobot awal untuk *input layer*, *hidden layer*, dan bias kita inisialisasi secara acak dalam bentuk matrik sebagai berikut:

1. Bobot awal lapisan input ke lapisan tersembunyi

A = [...  
 0.4500 0.0894 -0.8542 0.8416  
 0.9579 -0.1978 -0.5595 -0.6146  
 -0.6119 -0.9953 0.1435 -0.4015  
 -0.6320 -0.7867 -0.4583 0.5986  
 -0.0152 0.1257 0.4672 -0.1576];

2. Bobot bias awal lapisan input ke lapisan tersembunyi

B = [...  
 -0.9086  
 -0.6140  
 0.9221  
 0.9021  
 -0.9166];

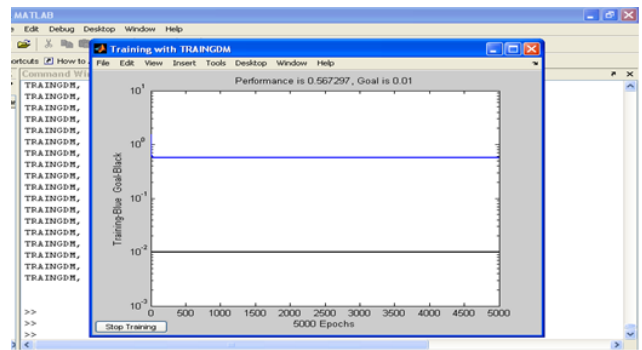
3. Bobot awal lapisan tersembunyi ke lapisan output :

C = [...  
 0.6595 -0.3997 0.4192 -0.8508 0.5004  
 0.4500 0.0894 0.4672 -0.1576 0.9579];

4. Bobot bias awal lapisan tersembunyi ke lapisan output :

D=[...  
 -0.0622  
 0.9021];

Langkah berikutnya adalah pelatihan jaringan. Source code menggunakan matlab versi 7.0 untuk pelatihan neural network struktur backpropagation arsitektur jaringan di atas secara lengkap ada pada lampiran. Hasil pelatihan oleh matlab diperlihatkan oleh gambar di bawah ini:



Gambar 3 Hasil Pelatihan Data Training dengan arsitektur Jaringan 4-30-2

Arsitektur jaringan terbaik yang didapat saat pelatihan di atas akan digunakan sebagai arsitektur jaringan untuk tingkat kemiskinan pada data testing. Bobot awal pada data testing adalah bobot terakhir saat pelatihan dari arsitektur jaringan. Secara lengkap nilai bobot awal data testing atau bobot akhir jaringan terbaik secara lebih. Hasil peramalan tingkat kemiskinan pada data testing diperlihatkan pada tabel di bawah ini:

Tabel di atas menggambarkan hasil kerja neural network dalam matlab dimana setelah terpilih arsitektur jaringan yang tepat yaitu 4-30-2 data testing dimasukkan akan langsung menghasilkan indeks kedalaman kemiskinan (P1) dan indeks keparahan kemiskinan (P2) lebih cepat.

**Pengujian Model**

Untuk mengukur ketepatan dan keakuratan model prediksi tingkat kemiskinan di Indonesia ditentukan dengan menghitung nilai *Percentage Error* (PE), *Mean Square Error* (MSE), *Mean Absolute Deviation* (MAD) dan *Mean Absolute Percentage Error* (MAPE). MAPE digunakan untuk menilai prestasi jaringan yang dilatih karena MAPE mengenal secara pasti signifikansi hubungan diantara data ramalan dengan data aktual melalui persentase dari data aktual serta indikator positif atau negatif pada galat (error) serta memberikan petunjuk seberapa besar kesalahan peramalan dibandingkan dengan nilai sebenarnya dari data tersebut. Berdasarkan tabel 2 dan tabel 3 di atas akan disusun neural network struktur backpropagation, seperti terlihat pada tabel 4 dibawah ini:

Tabel 3 Perhitungan Nilai MAD neural network struktur backpropagation.

| No | Nilai Real |       | NN Backpropagation |        | MAD NNB |        |
|----|------------|-------|--------------------|--------|---------|--------|
|    | P1(%)      | P2(%) | P1(%)              | P2(%)  | P1      | P2     |
| 1  | 3.07       | 0.83  | 3.0734             | 0.8304 | 0.0034  | 0.0004 |
| 2  | 1.82       | 0.50  | 2.1177             | 0.5703 | 0.2977  | 0.0703 |
| 3  | 1.24       | 0.31  | 2.1176             | 0.5703 | 0.8776  | 0.2603 |
| 4  | 1.13       | 0.25  | 2.1176             | 0.5703 | 0.9876  | 0.3203 |
| 5  | 1.37       | 0.44  | 2.1176             | 0.5703 | 0.7476  | 0.1303 |
| 6  | 1.85       | 0.43  | 2.1433             | 0.5726 | 0.2933  | 0.1426 |
| 7  | 3.05       | 0.80  | 3.0628             | 0.8033 | 0.0128  | 0.0033 |
| 8  | 2.53       | 0.62  | 2.4926             | 0.6035 | 0.0374  | 0.0165 |
| 9  | 0.66       | 0.14  | 2.1176             | 0.5703 | 1.4576  | 0.4303 |
| 10 | 0.85       | 0.19  | 2.1177             | 0.5703 | 1.2677  | 0.3803 |
| 11 | 0.56       | 0.15  | 2.1176             | 0.5703 | 1.5576  | 0.4203 |
| 12 | 1.62       | 0.42  | 2.118              | 0.5703 | 0.4980  | 0.1503 |
| 13 | 2.39       | 0.57  | 2.3774             | 0.5735 | 0.0126  | 0.0035 |
| 14 | 2.89       | 0.75  | 2.8562             | 0.7373 | 0.0338  | 0.0127 |
| 15 | 1.93       | 0.44  | 2.1306             | 0.5706 | 0.2006  | 0.1306 |
| 16 | 0.95       | 0.28  | 2.1176             | 0.5703 | 1.1676  | 0.2903 |
| 17 | 0.39       | 0.07  | 2.1176             | 0.5703 | 1.7276  | 0.5003 |
| 18 | 3.20       | 0.83  | 3.2354             | 0.8461 | 0.0354  | 0.0161 |
| 19 | 3.47       | 0.91  | 3.4729             | 0.9025 | 0.0029  | 0.0075 |
| 20 | 1.24       | 0.33  | 2.1176             | 0.5703 | 0.8776  | 0.2403 |
| 21 | 1.08       | 0.27  | 2.1176             | 0.5703 | 1.0376  | 0.3003 |
| 22 | 0.76       | 0.17  | 2.1176             | 0.5703 | 1.3576  | 0.4003 |
| 23 | 0.99       | 0.25  | 2.1177             | 0.5704 | 1.1277  | 0.3204 |
| 24 | 1.18       | 0.30  | 2.1176             | 0.5703 | 0.9376  | 0.2703 |
| 25 | 2.82       | 0.82  | 2.8273             | 0.8191 | 0.0073  | 0.0009 |



|    |      |      |        |        |        |        |
|----|------|------|--------|--------|--------|--------|
| 26 | 1.68 | 0.42 | 2.1176 | 0.5703 | 0.4376 | 0.1503 |
| 27 | 1.92 | 0.49 | 2.1193 | 0.5709 | 0.1993 | 0.0809 |
| 28 | 3.21 | 0.84 | 3.2007 | 0.8394 | 0.0093 | 0.0006 |
| 29 | 1.74 | 0.40 | 2.1192 | 0.571  | 0.3792 | 0.1710 |
| 30 | 4.38 | 1.31 | 3.635  | 1.0586 | 0.7450 | 0.2514 |
| 31 | 0.85 | 0.14 | 2.1176 | 0.5704 | 1.2676 | 0.4304 |
| 32 | 5.71 | 1.71 | 3.636  | 1.0587 | 2.0740 | 0.6513 |
| 33 | 7.35 | 2.44 | 3.636  | 1.0587 | 3.7140 | 1.3813 |
|    |      |      |        |        | 0.77   | 0.24   |

Tabel diatas menunjukkan perbedaan hasil perhitungan antara data real (data testing) dengan mamdani yang jauh lebih tinggi dari realnya sedangkan metode *neural network* memberikan hasil yang mendekati data aslinya. Sehingga hasil *mean absolute deviation* (MAD) yang diperoleh *neural network* memberikan selisih yang kecil .

**Tabel 4 Perhitungan Nilai MSE neural network struktur backpropagation.**

| No | Nilai Real |       | NN Backpropagation |        | MSE NNB |        |
|----|------------|-------|--------------------|--------|---------|--------|
|    | P1(%)      | P2(%) | P1(%)              | P2(%)  | P1      | P2     |
| 1  | 3.07       | 0.83  | 3.0734             | 0.8304 | 0.0000  | 0.0000 |
| 2  | 1.82       | 0.50  | 2.1177             | 0.5703 | 0.0886  | 0.0049 |
| 3  | 1.24       | 0.31  | 2.1176             | 0.5703 | 0.7702  | 0.0678 |
| 4  | 1.13       | 0.25  | 2.1176             | 0.5703 | 0.9754  | 0.1026 |
| 5  | 1.37       | 0.44  | 2.1176             | 0.5703 | 0.5589  | 0.0170 |
| 6  | 1.85       | 0.43  | 2.1433             | 0.5726 | 0.0860  | 0.0203 |
| 7  | 3.05       | 0.80  | 3.0628             | 0.8033 | 0.0002  | 0.0000 |
| 8  | 2.53       | 0.62  | 2.4926             | 0.6035 | 0.0014  | 0.0003 |
| 9  | 0.66       | 0.14  | 2.1176             | 0.5703 | 2.1246  | 0.1852 |
| 10 | 0.85       | 0.19  | 2.1177             | 0.5703 | 1.6071  | 0.1446 |
| 11 | 0.56       | 0.15  | 2.1176             | 0.5703 | 2.4261  | 0.1767 |
| 12 | 1.62       | 0.42  | 2.118              | 0.5703 | 0.2480  | 0.0226 |
| 13 | 2.39       | 0.57  | 2.3774             | 0.5735 | 0.0002  | 0.0000 |
| 14 | 2.89       | 0.75  | 2.8562             | 0.7373 | 0.0011  | 0.0002 |
| 15 | 1.93       | 0.44  | 2.1306             | 0.5706 | 0.0402  | 0.0171 |
| 16 | 0.95       | 0.28  | 2.1176             | 0.5703 | 1.3633  | 0.0843 |
| 17 | 0.39       | 0.07  | 2.1176             | 0.5703 | 2.9846  | 0.2503 |
| 18 | 3.20       | 0.83  | 3.2354             | 0.8461 | 0.0013  | 0.0003 |
| 19 | 3.47       | 0.91  | 3.4729             | 0.9025 | 0.0000  | 0.0001 |
| 20 | 1.24       | 0.33  | 2.1176             | 0.5703 | 0.7702  | 0.0577 |
| 21 | 1.08       | 0.27  | 2.1176             | 0.5703 | 1.0766  | 0.0902 |
| 22 | 0.76       | 0.17  | 2.1176             | 0.5703 | 1.8431  | 0.1602 |
| 23 | 0.99       | 0.25  | 2.1177             | 0.5704 | 1.2717  | 0.1027 |
| 24 | 1.18       | 0.30  | 2.1176             | 0.5703 | 0.8791  | 0.0731 |
| 25 | 2.82       | 0.82  | 2.8273             | 0.8191 | 0.0001  | 0.0000 |
| 26 | 1.68       | 0.42  | 2.1176             | 0.5703 | 0.1915  | 0.0226 |
| 27 | 1.92       | 0.49  | 2.1193             | 0.5709 | 0.0397  | 0.0065 |
| 28 | 3.21       | 0.84  | 3.2007             | 0.8394 | 0.0001  | 0.0000 |
| 29 | 1.74       | 0.40  | 2.1192             | 0.571  | 0.1438  | 0.0292 |
| 30 | 4.38       | 1.31  | 3.635              | 1.0586 | 0.5550  | 0.0632 |
| 31 | 0.85       | 0.14  | 2.1176             | 0.5704 | 1.6068  | 0.1852 |
| 32 | 5.71       | 1.71  | 3.636              | 1.0587 | 4.3015  | 0.4242 |
| 33 | 7.35       | 2.44  | 3.636              | 1.0587 | 13.793  | 1.9080 |
|    |            |       |                    |        | 8       | 0.127  |
|    |            |       |                    |        | 1.2045  | 8      |

Tabel diatas menunjukkan hasil perhitungan data real (data testing) metode *neural network* memberikan hasil yang mendekati data aslinya.

Sehingga hasil mean standard error (MSE) yang diperoleh *neural network* memberikan selisih yang kecil.

Dari kedua tabel di atas dapat dilihat nilai MAD dan MSE *neural network backpropagation* seperti terlihat pada tabel berikut:

**Tabel 5 Tabel Perbandingan Tingkat Error Model Fuzzy Logic Mamdani dan Neural Network Backpropagation.**

| Tingkat Error | NN Backpropagation |        |
|---------------|--------------------|--------|
|               | P1                 | P2     |
| MAD           | 0.7694             | 0.2405 |
| MSE           | 1.2045             | 0.1278 |

Berdasarkan tabel tersebut terlihat nilai error untuk metode *neural network struktur backpropagation* lebih kecil. Dengan perkataan lain, tingkat akurasi dan *performance* dari metode *neural network struktur backpropagation* dalam melakukan prediksi tingkat kemiskinan di Indonesia lebih baik.

**KESIMPULAN**

Dalam penelitian ini dilakukan pembuatan model menggunakan *neural network struktur backpropagation* menggunakan data tingkat kemiskinan di Indonesia. Model yang dihasilkan, untuk mengetahui metode ini memberikan hasil yang baik dalam peramalan pada perhitungan tingkat kemiskinan. Untuk mengukur kinerja tersebut digunakan metode pengujian *mean absolute deviation* (MAD) untuk indeks kedalaman kemiskinan sebesar 0,7694 dan indeks keparahan kemiskinan sebesar 0,2405 sedangkan *mean square error* (MSE) untuk indeks kedalaman kemiskinan sebesar 1,2045 dan indeks keparahan kemiskinan sebesar 0,1278, diketahui bahwa metode *neural network struktur backpropagation* memiliki nilai *accuracy* dan kinerja yang baik.

Dengan demikian, penulis tentukan benar ternyata metode *neural network struktur backpropagation* merupakan metode yang cukup baik dalam hal peramalan, sehingga metode *neural network struktur backpropagation* dapat memberikan pemecahan untuk permasalahan peramalan pada perhitungan tingkat kemiskinan di Indonesia.

**DAFTAR PUSTAKA**

Badan Pusat Statistika. 2012. Statistika Kemiskinan 2007-2012. Jakarta: BPS Indonesia.

Boa, Handayani.2008. Analisis Model Kemiskinan Perdesaan di Indonesia. EEP Vol 5 no 1. 27 Februari 2013.

Fuad FM.2011. Prediksi Ketersediaan Beras di Masyarakat dengan Menggunakan Logika Fuzzy dan Jaringan Syaraf Tiruan dalam Upaya Peningkatan Ketahanan Pangan. AGROINTEK Volume 5, No.1 Maret 2005.

Kusrini,&Luthfi, E. T.2009.Algoritma Data Mining. Yogyakarta: Andi Publishing.

Kusumadewi, Sri dan Purnomo, Hari.2010. Aplikasi Logika Fuzzy Untuk Pendukung Keputusan. Yogyakarta: Graha Ilmu.

Rohmatullah dan Marimin.2007. Logika Fuzzy dan Jaringan Syaraf Tiruan untuk Peningkatan Mutu The Hitam. Jurnal Teknologi dan Industri Pangan, Vol. XVIII No.2 tahun 2007.

Septiani, Winnie dan Marimin. 2005. Sistem Intelijen Prediksi dan Penilaian Kualitas Susu Pasteurisasi dengan menggunakan Logika Fuzzy dan Jaringan Syaraf Tiruan. Yogyakarta: Seminar Nasional Aplikasi Teknologi Informasi 2005, Juni 2005.

Sumathi, dan S., Sivanandam, S.N. 2006. *Introduction to Data Mining and its Applications*. Berlin Heidelberg New York: Springer

Vercellis, Carlo (2009). *Business Intelligent: Data Mining and Optimization for Decision Making*. Southern Gate, Chichester, West Sussex: John Wiley & Sons, Ltd.

#### BIODATA PENULIS



Astriana Mulyani. Lahir di Jakarta, 11 Agustus 1973. Menempuh S1 pada Fakultas Sains dan Matematika jurusan Matematika program studi Matematika Industri dan

Statistika, S2 Ilmu Komputer Konsentrasi Sistem Informasi Manajemen. Saat ini bekerja sebagai dosen di STMIK Nusa mandiri Jakarta adapun jurnal yang pernah saya hasilkan:

1. Monitoring Traffic Dan Manajemen Bandwidth Jaringan Komputer Pada Badan Sar Nasional Menggunakan Aplikasi PRTG
2. Perancangan Animasi Berkendara Yang Aman Dan Nyaman
3. Perhitungan Komponen Elektronika Dasar Berbasis Android.

