

## PENERAPAN METODE JARINGAN SYARAF TIRUAN *BACKPROPAGATION* UNTUK PREDIKSI TINGKAT KEMISKINAN DI PROVINSI JAWA TENGAH

Dian Finaliamartha<sup>\*1</sup>, Didi Supriyadi<sup>2</sup>, dan Gita Fadila Fitriana<sup>3</sup>

<sup>1,2,3</sup>Institut Teknologi Telkom Purwokerto, Kabupaten Banyumas  
Email: <sup>1</sup>17102149@ittelkom-pwt.ac.id, <sup>2</sup>didisupriyadi@ittelkom-pwt.ac.id, <sup>3</sup>gita@ittelkom-pwt.ac.id

<sup>\*</sup>Penulis Korespondensi

(Naskah masuk: 26 Februari 2021, diterima untuk diterbitkan: 18 Agustus 2022)

### Abstrak

Kemiskinan merupakan permasalahan yang semestinya dipandang sebagai suatu masalah sosial yang kompleks (multidimensional). Berdasarkan data dari Badan Pusat Statistik, presentase kemiskinan nasional pada Maret 2019 sebesar 9,41 persen. Sementara, Provinsi Jawa Tengah memiliki tingkat kemiskinan lebih tinggi dibandingkan dengan tingkat kemiskinan nasional yakni sebesar 10,8 persen. Tingginya tingkat kemiskinan dapat menyebabkan tindak kriminal, tingginya angka pengangguran, kekacauan sosial, politik dan lain sebagainya. Oleh karena itu, penelitian ini bertujuan untuk menganalisis tingkat kemiskinan dengan menentukan model yang tepat yang selanjutnya dapat digunakan untuk melakukan prediksi tingkat kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah. Data yang digunakan diperoleh dari Badan Pusat Statistik Provinsi Jawa Tengah dari tahun 2010 sampai dengan 2019 yang terdiri dari data Laju Pertumbuhan Ekonomi, Tingkat Pengangguran Terbuka, Indeks Pembangunan Manusia, dan Tingkat Kemiskinan menurut kabupaten/kota. Metode yang digunakan yaitu Jaringan Syaraf Tiruan *Backpropagation*. Jaringan Syaraf Tiruan *Backpropagation* memiliki kinerja yang baik dalam menyelesaikan permasalahan, salah satunya masalah prediksi. Berdasarkan model arsitektur terbaik yang dihasilkan pada penelitian ini, yaitu model arsitektur 3-35-1 dapat dihasilkan tingkat akurasi sebesar 95,2% menggunakan *MSE* pada proses pengujian menggunakan data pengujian. Sehingga dapat disimpulkan bahwa Jaringan Syaraf Tiruan *Backpropagation* dengan menerapkan model yang tepat dapat menghasilkan tingkat akurasi yang baik yang selanjutnya dapat digunakan sebagai alternatif untuk melakukan prediksi tingkat kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah di masa mendatang.

**Kata kunci:** Jaringan Syaraf Tiruan, *Backpropagation*, Tingkat Kemiskinan, Prediksi, Matlab

## IMPLEMENTATION OF BACKPROPAGATION ARTIFICIAL NEURAL NETWORK METHODS FOR POVERTY LEVEL PREDICTION IN CENTRAL JAVA PROVINCE

### Abstract

Poverty is a problem that should be seen as a complex (multi-dimensional) social problem. Based on data from the Central Statistics Agency, the percentage of national poverty in March 2019 was 9.41 percent. Meanwhile, Central Java Province has a higher poverty rate than the national poverty level, which is 10.8 percent. High levels of poverty can lead to criminal acts, high levels of unemployment, social, political chaos, and so on. Therefore, this study aims to analyze the level of poverty by determining the appropriate model which can then be used to predict poverty levels according to districts/cities in Central Java Province. The data used was obtained from the Central Java Province Central Bureau of Statistics from 2010 to 2019 which consisted of data on economic growth rates, open unemployment rates, human development indexes, and poverty levels according to districts/cities. The method used is the Backpropagation Neural Network. The Backpropagation Artificial Neural Network has a good performance in solving problems, one of which is prediction problems. Based on the best architectural model produced in this study, the 3-35-1 architectural model can produce an accuracy rate of 95.2% using MSE in the testing process using test data. So it can be concluded that the Backpropagation Neural Network by applying the right model can produce a good level of accuracy which can then be used as an alternative to predict poverty levels according to regencies/cities in Central Java Province in the future.

**Keywords:** Artificial Neural Network, *Backpropagation*, Poverty Level, Prediction, Matlab

### 1. PENDAHULUAN

Agenda nasional yang terdapat dalam Rencana Pembangunan Jangka Menengah Nasional salah satunya

adalah untuk meningkatkan kesejahteraan rakyat dengan mengurangi kemiskinan dan pengangguran (*Badan Pusat Statistik Kabupaten Cilacap, 2019*). Kemiskinan

merupakan suatu masalah sosial yang kompleks (multidimensional). Secara umum, kemiskinan dicirikan oleh angka pendapatan yang rendah, kondisi kesehatan yang buruk, pendidikan yang rendah, rentan terhadap gejala ekonomi, rendahnya partisipasi dalam proses pengambilan kebijakan terkait penanggulangan kemiskinan, serta keamanan individu yang kurang (Puguh Budro Irawan, M.A, 2016).

Berdasarkan data dari Badan Pusat Statistik Indonesia, persentase penduduk miskin pada bulan Maret 2019 tercatat sebesar 9,41 persen, menurun sebesar 0,25 persen poin terhadap bulan September 2018 dan menurun sebesar 0,41 persen poin terhadap bulan Maret 2018 (Badan Pusat Statistik, 2018). Bersumber dari data Badan Pusat Statistik Provinsi Jawa Tengah, pada tahun 2019 Provinsi Jawa Tengah memiliki tingkat kemiskinan lebih tinggi dibandingkan dengan tingkat kemiskinan nasional, yaitu sebesar 10,8 persen. Tingginya tingkat kemiskinan di suatu negara dapat menyebabkan tingginya angka pengangguran, tindakan kriminal, gangguan kesehatan, terjadi kekacauan sosial, politik dan lain sebagainya (Putri, 2019).

Jika dilakukan suatu prediksi dini terhadap tingkat kemiskinan, seperti di Provinsi Jawa Tengah menurut kabupaten/kota, maka Pemerintah Provinsi Jawa Tengah dapat cepat tanggap dalam membuat suatu kebijakan dan mengambil keputusan yang tepat untuk menanggulangi permasalahan tersebut. Akan tetapi, dibutuhkan model dasar dan data - data terdahulu untuk melakukan prediksi, sehingga membutuhkan teknik penyelesaian yang lebih maju dan menghasilkan tingkat akurasi yang akurat (Andriani, Silitonga and Wanto, 2018).

Berdasarkan permasalahan tersebut, maka akan dilakukan analisis dengan menerapkan model prediksi yang tepat untuk selanjutnya dapat digunakan sebagai alternatif dalam melakukan prediksi tingkat kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah berdasarkan variabel Laju Pertumbuhan Ekonomi, Tingkat Pengangguran Terbuka dan Indeks Pembangunan Manusia. Pemilihan ketiga variabel ini dengan pertimbangan bahwa variabel ini terukur, terjamin ketersediaan datanya, dan cukup berpengaruh terhadap kemiskinan, seperti yang ditunjukkan oleh penelitian terdahulu, meliputi penelitian (Kaunang, 2018) tentang analisis tingkat kemiskinan berdasarkan dimensi dasar IPM, penelitian (Rahmawati, Safitri and Fairuzdhiya, 2015) tentang analisis pengangguran terhadap kemiskinan, dan penelitian (Prayoga, 2019) tentang analisis pengaruh pertumbuhan ekonomi, pengangguran dan belanja pemerintah terhadap kemiskinan.

Metode yang digunakan yaitu Jaringan Syaraf Tiruan (JST) *Backpropagation*. Berdasarkan penelitian (Mulyani, 2016) yang melakukan prediksi tingkat kemiskinan di Indonesia dengan membandingkan JST *Backpropagation* dengan *Fuzzy Mamdani*, dihasilkan tingkat akurasi dan kinerja dari JST *Backpropagation* lebih baik dibandingkan *Fuzzy Mamdani* dalam melakukan prediksi tingkat kemiskinan di Indonesia.

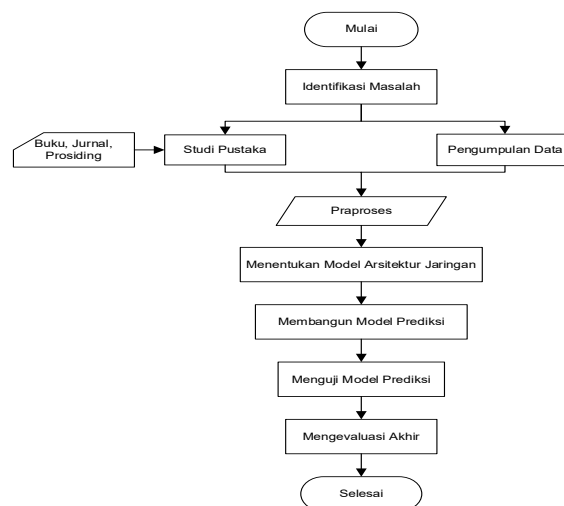
Pada penelitian (Wanto, 2018) yang menggunakan JST *Backpropagation* dan penelitian (Harliana and Kirono, 2019) yang menggunakan JST *Learning Vector Quantization (LVQ)* untuk prediksi kemiskinan, diperoleh tingkat akurasi dan kinerja dari metode *Backpropagation* lebih baik dibandingkan metode *LVQ*. Hal ini dikarenakan JST *Backpropagation* menggunakan aturan pembelajaran dalam melakukan prediksi berdasarkan data-data di masa lalu yang terjadi sebelumnya (Andriani, Silitonga and Wanto, 2018).

Kemampuan yang dimiliki metode *Backpropagation* adalah menghasilkan kinerja yang lebih baik dalam proses pelatihan yang dilakukan secara berulang. Dengan demikian, bobot Jaringan Syaraf Tiruan yang saling terhubung dapat mendekati berat yang seharusnya. Metode *Backpropagation* juga memiliki kemampuan dalam melakukan proses pembelajaran terhadap data - data secara adaptif dan *multilayer* dengan melakukan perubahan bobot sehingga kesalahan yang dihasilkan lebih sedikit (Yanto *et al.*, 2018). JST metode *Backpropagation* ini sering digunakan untuk menyelesaikan beberapa permasalahan, salah satunya adalah masalah prediksi (Windarto, Lubis and Solikhun, 2018).

## 2. METODE PENELITIAN

### 2.1. Kerangka Kerja Penelitian

Pada bagian ini kerangka kerja penelitian yang digunakan dalam menyelesaikan masalah penelitian:



Gambar 1. Kerangka Kerja Penelitian  
Sumber : (Andriani, Silitonga and Wanto, 2018)

Berdasarkan Gambar 3.1 dapat dijelaskan dari masing-masing tahapan penelitian sebagai berikut:

#### 1. Identifikasi Masalah

Melakukan identifikasi masalah yaitu bagaimana cara menentukan model prediksi yang tepat yang selanjutnya dapat digunakan untuk melakukan prediksi tingkat kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah dengan menerapkan Jaringan Syaraf Tiruan metode *Backpropagation*.

## 2. Studi Pustaka

Suatu penelitian perlu dilengkapi teori-teori dan pengetahuan dasar dengan melakukan studi pustaka berdasarkan buku, jurnal, prosiding atau penelitian terdahulu yang sesuai dengan penelitian yang dilakukan untuk dijadikan sebagai acuan.

## 3. Pengumpulan Data

Data yang digunakan yaitu data sekunder yang diperoleh dari *website* resmi Badan Pusat Statistik Provinsi Jawa Tengah dengan total *record* 350 data. Data tersebut meliputi data *input* yaitu Laju Pertumbuhan Ekonomi, Tingkat Pengangguran Terbuka, Indeks Pembangunan Manusia dan data target yaitu Tingkat Kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah dari tahun 2010 sampai dengan 2019 berupa data persentase. Data tersebut terbagi menjadi dua dengan perbandingan 70:30. 70% atau 245 data untuk pelatihan (tahun 2010 sampai dengan 2016) dan 30% atau 105 data untuk pengujian (tahun 2017 sampai dengan 2019).

## 4. Praproses

Melakukan praproses untuk mempermudah dalam memahami isi *record* data tingkat kemiskinan dengan melakukan normalisasi data agar keluaran dari jaringan dapat sesuai dengan fungsi aktivasi yang digunakan, yaitu fungsi aktivasi *sigmoid* (biner) yang merupakan fungsi *asimtotik* (tidak pernah mencapai 0 ataupun 1) (Sihotang and Wanto, 2018). Sehingga normalisasi data dilakukan pada interval yang lebih kecil, misalnya interval  $[0,1;0,9]$  menggunakan persamaan (Satria, 2018):

$$X = \frac{0,8(x-a)}{b-a} 0,1 \quad (1)$$

Keterangan:

0,8 = Ketetapan

x = Nilai data ke-n

a = Nilai data yang paling rendah

b = Nilai data yang paling tinggi

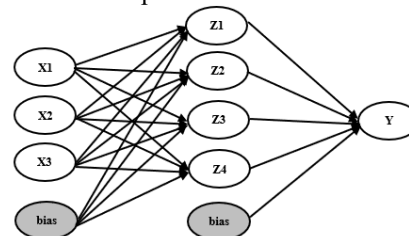
Sebelum melakukan normalisasi data, dilakukan pendataan variabel dan inisialisasi data tingkat kemiskinan sebagai berikut:

Tabel 1. Pendataan Variabel dan Inisialisasi Data

No	Variabel	Inisialisasi	Nilai
1	Laju Pertumbuhan Ekonomi	X1	a = 1,5 b = 83,19
2	Tingkat Pengangguran Terbuka	X2	
3	Indeks Pembangunan Manusia	X3	
4	Tingkat Kemiskinan	Y	

## 5. Menentukan Model Arsitektur Jaringan

Penentuan model arsitektur jaringan terkait permasalahan tingkat kemiskinan. Berikut ini merupakan desain model arsitektur jaringan yang digunakan dalam penelitian:



Gambar 2. Desain Arsitektur Jaringan  
Sumber : (Ashshiddiqi, 2018)

Gambar 2 merupakan desain model arsitektur jaringan yang digunakan untuk melakukan analisis tingkat kemiskinan dengan menentukan model prediksi yang tepat yang selanjutnya dapat digunakan untuk melakukan prediksi tingkat kemiskinan di Provinsi Jawa Tengah. Model arsitektur tersebut terdiri dari 3 *input layer* (X1, X2, X3), 4 *neuron hidden layer* (Z1, Z2, Z3, Z4) dan 1 *output layer* (Y). X1 merupakan variabel Laju Pertumbuhan Ekonomi, X2 merupakan variabel Tingkat Pengangguran Terbuka dan X3 merupakan variabel Indeks Pembangunan Manusia. *Output* atau keluaran jaringan yang dihasilkan yaitu tingkat kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah.

## 6. Membangun Model Prediksi

Membangun beberapa model arsitektur jaringan menggunakan data pelatihan (*training*) yang telah dinormalisasi, kemudian melakukan pelatihan data menggunakan Matlab R2019a dengan menggunakan variasi jumlah *neuron* pada *hidden layer*, laju pembelajaran dan konstanta momentum sehingga akan diperoleh model arsitektur terbaik.

## 7. Menguji Model Prediksi

Melakukan pengujian menggunakan data pengujian (*testing*) yang telah dinormalisasi terhadap model arsitektur terbaik menggunakan Matlab R2019a. Setelah data diuji, maka akan diperoleh data yang sesuai target (data aktual) dan data yang tidak sesuai target. Berdasarkan data yang sesuai target, kemudian akan diperoleh tingkat akurasi dari hasil pengujian menggunakan model arsitektur terbaik. Model terbaik tersebut selanjutnya dapat digunakan sebagai alternatif untuk melakukan prediksi tingkat kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah di masa mendatang.

## 8. Mengevaluasi Akhir

Melakukan evaluasi akhir terhadap hasil pengolahan data untuk mengetahui apakah data tersebut sesuai seperti yang diinginkan serta

mengetahui keakuratan jaringan menggunakan *Mean Square Error (MSE)* berdasarkan persamaan (Santoso and Hansun, 2019):

$$MSE = \frac{\sum_{i=1}^n e_i^2}{n} \quad (2)$$

Keterangan:

$e_i$  = selisih nilai target dengan keluaran jaringan

$n$  = jumlah data pembelajaran yang digunakan

## 2.2. Kemiskinan

Kemiskinan bersifat multidimensional, hal ini dikarenakan kebutuhan yang dimiliki manusia berbeda-beda sehingga memiliki banyak dimensi (Prayoga, 2019). Secara umum, kemiskinan dapat dikonseptualisasikan secara absolut atau kemiskinan absolut dan secara relatif atau kemiskinan relatif (Puguh Budro Irawan, M.A, 2016).

Kemiskinan absolut atau yang berarti kesengsaraan hidup (*destitution*) diartikan sebagai suatu deprivasi atau kekurangan atas kebutuhan dasar yang dimiliki manusia untuk dapat hidup layak, termasuk makanan, air bersih, sandang, tempat tinggal, sanitasi, kesehatan dan pendidikan. Sedangkan, kemiskinan relatif diartikan sebagai ketimpangan ekonomi, yang dapat diukur berdasarkan kelas pendapatan yang diperoleh dari kelompok masyarakat dengan kelas pendapatan terendah, menengah dan tertinggi (Puguh Budro Irawan, M.A, 2016).

## 2.3. Jaringan Syaraf Tiruan (JST)

JST adalah sistem yang digunakan dalam pengolahan suatu informasi berdasarkan sistem saraf secara biologis, seperti halnya proses informasi pada otak manusia (Irawan, Zarlis and Nababan, 2017). JST merupakan salah satu metode pada kecerdasan buatan. Kemampuan yang dimiliki yaitu mengidentifikasi pola, *signal processing* dan melakukan prediksi berdasarkan metode pembelajaran yang dilakukan (Satria, 2018).

JST memiliki lapisan-lapisan penyusun yang terbagi menjadi tiga antara lain (Jaya *et al.*, 2018):

### 1. Input Layer

Unit-unit *input* merupakan *node-node* di dalam *input layer*. Unit-unit *input* akan menerima *input* dari luar. Penggambaran suatu masalahnya yaitu *input* yang dimasukkan.

### 2. Hidden Layer

Unit-unit *hidden* merupakan *node-node* di dalam *hidden layer*. *Output* dari *hidden layer* tidak bisa terlihat secara langsung.

### 3. Output Layer

Unit-unit *output* merupakan *node-node* di dalam *output layer*. *Output* dari lapisan ini yaitu *output* Jaringan Syaraf Tiruan terhadap suatu permasalahan.

## 2.4. Algoritma Backpropagation

*Backpropagation* merupakan metode yang banyak digunakan untuk menyelesaikan suatu permasalahan yang berkaitan dengan prediksi, identifikasi, dan pengenalan pola (Rahul *et al.*, 2020). *Backpropagation* yaitu sebuah metode atau algoritma pembelajaran yang termasuk ke dalam *supervised learning* yang memiliki banyak lapisan atau disebut dengan *multilayer perceptron* yang biasa digunakan oleh *perceptron* dalam melakukan perubahan bobot yang saling terhubung dengan *neuron* di dalam *hidden layer* (Santoso and Hansun, 2019).

Algoritma pelatihan untuk Jaringan Syaraf Tiruan *Backpropagation* adalah sebagai berikut (Satria, 2018):

### 1. Langkah 1:

Melakukan inisialisasi bobot-bobot menggunakan nilai acak yang cukup kecil.

### 2. Langkah 2:

Apabila kondisi perhitungan masih belum memenuhi target yang ditetapkan, maka lakukan langkah 3 sampai langkah 9 untuk tiap-tiap pasang pola data pelatihan.

### Tahap umpan maju:

### 3. Langkah 3:

Pada tiap-tiap *input neuron* akan menerima sinyal *input*  $X_i$  dan mengirimkannya pada *neuron* di *hidden layer*.

### 4. Langkah 4:

Nilai *input* pada tiap-tiap unit *hidden neuron* akan dihitung menggunakan nilai bobotnya :

$$Z_{netj} = V_{0j} + \sum_{i=1}^n X_i V_{ij} \quad (3)$$

Selanjutnya menghitung nilai *output* pada tiap-tiap unit  $Y_k$  dengan fungsi aktivasi:

$$Z_j = f(Z_{netj})$$

Fungsi aktivasinya yaitu fungsi aktivasi *sigmoid* (biner) dengan persamaan:

$$Z_j = \frac{1}{1 + e^{(-Z_{netj})}} \quad (4)$$

### 5. Langkah 5 :

Setiap *output neuron* ( $Y_k$ ,  $k=1,2,3,...,m$ ) nilai *input* dihitung dengan nilai bobotnya:

$$Y_{netk} = W_{0k} + \sum_{j=1}^p Z_j W_{jk} \quad (5)$$

Kemudian nilai *output* dihitung menggunakan fungsi aktivasi:

$$Y_k = f(Y_{netk})$$

### Tahap umpan mundur:

### 6. Langkah 6:

Tiap-tiap *output neuron* mendapatkan pola data target yang sesuai dengan pola data *input*, lalu informasi kesalahan dari lapisan *output* dihitung:

$$\delta_k = (t_k - Y_k) f'(Y_{netk}) = (t_k - Y_k) Y_k (1 - Y_k) \quad (6)$$

Kemudian menghitung koreksi nilai bobot yang nantinya akan digunakan untuk memperbaharui  $W_{jk}$ :

$$W_{jk} : \Delta W_{jk} = \alpha \delta_k Z_j$$

Selanjutnya koreksi nilai bias dihitung untuk memperbaharui  $W_{0k}$ :

$$W_{0k} : \Delta W_{0k} = \alpha \delta_k$$

Kemudian nilai  $\delta_k$  dari lapisan *output* akan dikirimkan ke *neuron* pada lapisan sebelumnya.

#### 7. Langkah 7:

Pada setiap *hidden neuron*

$$\delta_{net_j} = \sum_{k=1}^m \delta_k W_{jk} \quad (7)$$

Kemudian nilai tersebut digunakan untuk menghitung informasi kesalahan dengan cara dikalikan dengan nilai turunan dari fungsi aktivasi:

$$\delta_j = \delta_{net_j} f'(Z_{net_j}) = \delta_{net_j} Z_j (1 - Z_j) \quad (8)$$

Selanjutnya koreksi nilai bobot dihitung untuk memperbaharui  $V_{ij}$ :

$$\Delta V_{ij} = \alpha \delta_j X_i$$

Kemudian nilai koreksi bias dihitung untuk memperbaharui  $V_{0j}$ :

$$V_{0j} : \Delta V_{0j} = \alpha \delta_j$$

#### Tahap peng-update-an bobot dan bias :

#### 8. Langkah 8:

Tiap-tiap nilai bias dan bobot pada *hidden neuron* dilakukan peng-update-an bobot dan bias sehingga menjadi bobot dan bias baru dengan persamaan:

$$V_{ij}(\text{baru}) = V_{ij}(\text{lama}) + \Delta V_{ij} \quad (9)$$

Tiap - tiap nilai bias dan bobot pada *output neuron* dilakukan peng-update-an bobot dan bias, sehingga menjadi bobot dan bias baru dengan persamaan :

$$W_{jk}(\text{baru}) = W_{jk}(\text{lama}) + \Delta W_{jk} \quad (10)$$

#### 9. Langkah 9:

Tes kondisi berhenti.

### 2.5. Matlab

Matlab merupakan bahasa pemrograman yang berasal dari singkatan *Matirix Laboratory*. Matlab dikembangkan dan diperkenalkan oleh *The Mathwork Inc*. Saat ini, Matlab mempunyai kemampuan serta fitur yang lebih lengkap karena saat ini terdapat *toolbox*, seperti *toolbox neural network* yang telah disediakan oleh Matlab. Matlab dapat memberikan manfaat sebagai komputasi numerik, perhitungan matematika, analisis dan visualisasi data, pembuatan grafik, simulasi dan pemodelan, serta pengembangan sistem berbasis *General User Interface (GUI)* (Kurniawati, 2018).

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Hasil Normalisasi

Berdasarkan langkah-langkah algoritma pelatihan *backpropagation*, langkah pertama yaitu melakukan inisialisasi bobot-bobot menggunakan nilai acak yang cukup kecil. Fungsi yang disediakan Matlab *R2019a* untuk melakukan inisialisasi terhadap bobot dan bias seperti di bawah ini:

```

bobot_hidden = net_keluaran.IW{1,1};
bobot_keluaran = net_keluaran.LW{2,1};
bias_hidden = net_keluaran.b{1};
bias_keluaran = net_keluaran.b{2};

```

`Net_keluaran.IW{1,1}` untuk menginisialisasi bobot awal secara acak untuk lapisan pertama (*input layer*), sedangkan `net_keluaran.LW{2,1}` untuk menginisialisasi bobot secara acak pada lapisan kedua (*hidden layer*). Kemudian `net_keluaran.b{1}` untuk menentukan nilai bias secara acak dari *input layer* ke *hidden layer*, sedangkan `net_keluaran.b{2}` untuk menentukan nilai bias secara acak dari *hidden layer* ke *output layer*.

Berdasarkan inisialisasi bobot dan bias, langkah selanjutnya yaitu melakukan proses pelatihan menggunakan data pelatihan (*training*) pada Matlab *R2019a*. Proses pelatihannya meliputi:

1. Membuat Jaringan Syaraf Tiruan dengan fungsi pada Matlab *R2019a* sebagai berikut:

```

net = newff(minmax(data_latih),[35 1],{'logsig','purelin'},'traingdm');

```

Perintah di atas digunakan untuk membuat Jaringan Syaraf Tiruan *Backpropagation* menggunakan data pelatihan (*training*) yang sudah dinormalisasi dengan variasi *neuron* pada *hidden layer* 4, 10, 18, 25, 35, 50, 70, 100 dengan target atau *output* 1. Fungsi aktivasi yang digunakan yaitu *logsig* (*sigmoid biner*) dan *purelin* (*linier*), serta menggunakan fungsi pelatihan *traingdm*.

2. Inisialisasi parameter pelatihan dengan fungsi sebagai berikut:

```

net.performFcn = 'mse';
net.trainParam.show = 1000;
net.trainParam.goal = 0.001;
net.trainParam.epochs = 10000;
net.trainParam.mc = 0.8;
net.trainParam.lr = 0.4;

```

*Show* untuk menampilkan frekuensi pada perubahan *MSE* dengan *show* yang digunakan sebesar 1000. *Goal* untuk menentukan batas nilai *MSE* agar iterasi dapat dihentikan dengan *goal* yang digunakan sebesar 0.001. *Epoch* untuk menentukan jumlah iterasi maksimum pada pelatihan dengan *epoch* yang digunakan sebesar 10000. Momentum (*mc*) untuk menentukan

konstanta momentum pada pelatihan dengan momentum yang digunakan sebesar 0.5 sampai 0.8. *Learning rate* (lr) untuk menentukan laju pembelajaran pada pelatihan dengan *learning rate* sebesar 0.1 sampai 0.5.

### 3.2. Hasil Pelatihan

Setelah membuat Jaringan Syaraf Tiruan *Backpropagation* dan melakukan inisialisasi parameter jaringan, selanjutnya yaitu melakukan pelatihan Jaringan Syaraf Tiruan *Backpropagation* menggunakan Matlab R2019a dengan fungsi sebagai berikut:

```
[net_keluaran,tr,Y,E] = train(net,data_latih,target_latih);
```

Perintah ini digunakan untuk melakukan proses pelatihan jaringan menggunakan Matlab sampai dihasilkan nilai *error* yang sedikit menggunakan data pelatihan dan target pelatihan. Proses pelatihannya terbagi menjadi 2 meliputi:

1. Pelatihan 1, mencari jumlah variasi *neuron* di *hidden layer*

Pelatihan 1 bertujuan untuk menentukan banyaknya jumlah variasi *neuron* pada *hidden layer* untuk memperoleh kinerja terbaik berdasarkan proses pelatihan dengan menunjukkan nilai *MSE* paling kecil dan waktu komputasi (*time*) paling cepat. Dalam menentukan banyaknya jumlah variasi *neuron* pada *hidden layer*, pengamatan pada jaringan dilakukan dengan menggunakan satu *hidden layer* dengan jumlah variasi *neuron* pada *hidden layer* adalah 4, 10, 18, 25, 35, 50, 70, dan 100 *neuron*. Jumlah variasi *neuron* pada *hidden layer* ini dipilih secara *random* dengan proses percobaan menggunakan Matlab. Pelatihan ini dilakukan dengan menggunakan laju pembelajaran (*learning rate*) 0,1, konstanta momentum 0,7, maksimum iterasi (*epoch*) 10000 *epoch*, target *error* (*goal*) 0,001, dan menggunakan algoritma pelatihan *traingdm*. Parameter pelatihan yang digunakan berdasarkan referensi dari penelitian (Windarto, Lubis and Solikhun, 2018) dan (Supriyadi, 2013).

Berikut ini hasil pelatihan 1 dengan jumlah variasi *neuron* pada satu *hidden layer* menggunakan Matlab R2019a:

Tabel 2. Hasil pelatihan menggunakan satu hidden layer dengan variasi jumlah neuron pada hidden layer

Jumlah Neuron	MSE	Time
4	0,001049	0:00:13
10	0,0010437	0:00:15
18	0,0010014	0:00:11
25	0,001	0:00:07
35	<b>0,00099998</b>	<b>0:00:10</b>
50	0,00099999	0:00:11
70	0,0010083	0:00:13
100	0,0010689	0:00:15

Berdasarkan Tabel 2, dari delapan variasi jumlah *neuron* pada *hidden layer* dengan

mengamati nilai *MSE* dan waktu komputasi yang paling kecil, maka dapat disimpulkan bahwa model arsitektur terbaik pada pelatihan 1 adalah model arsitektur 3-35-1 dengan nilai *MSE* sebesar 0,00099998 dan waktu komputasi 0:00:10 detik. Model arsitektur terbaik tersebut selanjutnya akan digunakan pada proses pelatihan 2.

2. Pelatihan 2, mencari laju pembelajaran (*learning rate*) dan konstanta momentum terbaik

Pelatihan ini akan digunakan untuk mengamati berdasarkan variasi laju pembelajaran dan konstanta momentum terhadap kinerja jaringan dengan menunjukkan besar kecilnya nilai *MSE* dan lamanya iterasi dalam proses pelatihan. Pengamatan model arsitektur terbaik dilakukan dengan menggunakan *learning rate* 0,1 sampai dengan 0,5 dan konstanta momentum 0,5 sampai dengan 0,8, maksimum iterasi 10000 *epoch*, target *error* (*goal*) 0.001, dan algoritma pelatihannya menggunakan *traingdm*. Berikut ini hasil pengaruh laju pembelajaran dan momentum menggunakan model arsitektur terbaik yaitu 3-35-1:

Tabel 3. Hasil pengaruh laju pembelajaran dan momentum pada model arsitektur terbaik 3-35-1

Lr	Mc	05		06	
		MSE	TIME	MSE	TIME
0,1		0,00099999	0:00:07	0,00099999	0:00:13
0,2		0,00099999	0:00:12	0,00099998	0:00:07
0,3		0,00099999	0:00:10	0,00099999	0:00:10
0,4		0,0013629	0:00:11	0,0010456	0:00:11
0,5		0,0014955	0:00:16	0,00099999	0:00:04

Tabel 3. (lanjutan)

Lr	Mc	07		08	
		MSE	TIME	MSE	TIME
0,1		0,001	0:00:11	0,00099999	0:00:20
0,2		0,00099997	0:00:09	0,00099998	0:00:08
0,3		0,00099996	0:00:07	0,00099995	0:00:05
0,4		0,00099998	0:00:06	<b>0,00099997</b>	<b>0:00:03</b>
0,5		0,00099999	0:00:08	0,00099998	0:00:05

Berdasarkan Tabel 3, dengan mengamati pengaruh laju pembelajaran dan momentum pada model arsitektur terbaik 3-35-1, maka akan diperoleh laju pembelajaran dan konstanta momentum terbaik yaitu dengan laju pembelajaran 0,4 dan momentum 0,8 menghasilkan nilai *MSE* sebesar 0,00099997 dan waktu komputasi tercepat yaitu 0:00:03. Oleh karena itu, dapat disimpulkan secara keseluruhan untuk proses pelatihan jaringan dapat menghasilkan model arsitektur terbaik 3-35-1 dengan laju pembelajaran 0,4 dan momentum 0,8.

### 3.3. Hasil Pengujian

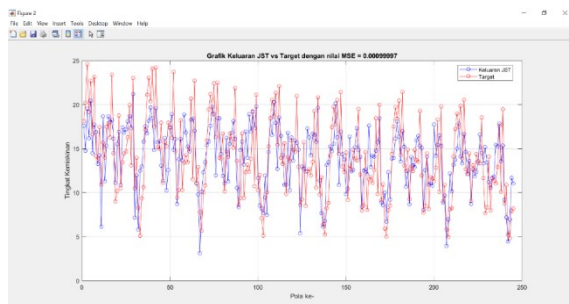
Setelah melakukan proses pelatihan menggunakan data pelatihan, selanjutnya yaitu melakukan proses pengujian menggunakan data pelatihan dan data pengujian.

1. Pengujian dengan data pelatihan



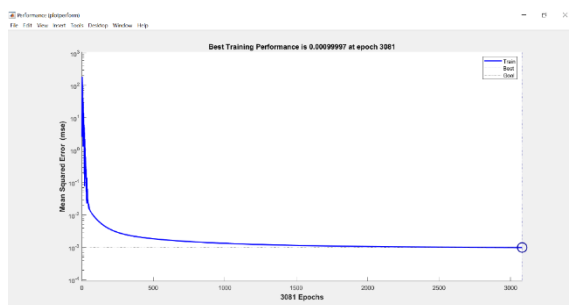
Berdasarkan hasil pelatihan diperoleh bobot optimal pada jaringan yang selanjutnya digunakan pada proses pengujian. Pengujian jaringan akan dilakukan dengan data pelatihan (*training*) sebanyak 245 pasangan data yang bertujuan untuk mengetahui apakah jaringan sudah mampu untuk mengenali pola data pelatihan dengan baik. Pengujian jaringan dilakukan menggunakan arsitektur satu *hidden layer*, jumlah *neuron* pada *hidden layer* 35 *neuron*, dengan laju pembelajaran (*learning rate*) 0,4 dan konstanta momentum 0,8.

Berikut ini grafik hasil pengujian dengan menggunakan data pelatihan (*training*) yang diperoleh menggunakan Matlab R2019a:



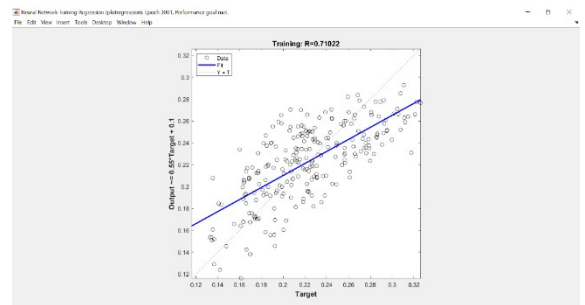
Gambar 3. Grafik hasil pelatihan dengan data pelatihan

Berdasarkan Gambar 3, dihasilkan target data dan *output* data yang merupakan hasil keluaran dari Jaringan Syaraf Tiruan *Backpropagation* pada Matlab R2019a dengan total 245 pola data pelatihan. Berdasarkan *output* jaringan terhadap target bahwa jaringan sudah mampu mengenali pola data pelatihan dengan tepat, dengan nilai korelasi antara *output* jaringan dan target sebesar 0,71022 dan *MSE* hasil pengujian dengan data pelatihan sebesar 0,0009997/0,001 yang tercapai pada iterasi 3081 *epoch* seperti yang ditunjukkan Gambar 4 dan 5.



Gambar 4. Proses pelatihan dan pengujian dengan data pelatihan

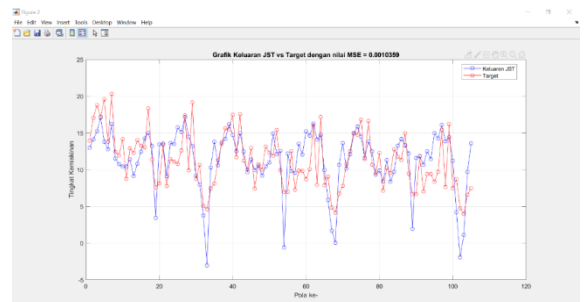
Berdasarkan Gambar 3, 4, dan 5 diperoleh hasil pengujian dengan data pelatihan berdasarkan target dan *output* dijabarkan pada Microsoft Excel, sehingga dengan 245 pola data pelatihan dapat memperoleh total nilai *SSE* yaitu sebesar 0,245023, dimana nilai *SSE* dihasilkan dari nilai  $Error^2$  dan memperoleh nilai *MSE* sebesar 0,001 yang dihasilkan dari total nilai *SSE* dibagi banyaknya data pelatihan.



Gambar 5. Nilai korelasi regresi pada proses pelatihan dan pengujian dengan data pelatihan

## 2. Pengujian dengan data pengujian

Pengujian dengan data pengujian bertujuan untuk mengetahui seberapa jauh sistem dapat mengenali pola yang telah dilatihkan. Pengujian dilakukan dengan menggunakan data yang belum pernah dilatih pada proses pelatihan sebelumnya. Pengujian dilakukan menggunakan arsitektur terbaik dengan satu *hidden layer*, jumlah *neuron* pada *hidden layer* 35 *neuron*, dengan laju pembelajaran 0,4 dan konstanta momentum 0,8. Berikut ini grafik hasil pengujian dengan data pelatihan (*testing*) yang diperoleh menggunakan Matlab R2019a:



Gambar 6. Grafik hasil pengujian dengan data pengujian

Berdasarkan Gambar 6, dihasilkan target data dan *output* data yang merupakan hasil keluaran dari Jaringan Syaraf Tiruan *Backpropagation* pada Matlab dengan total 105 pola data pengujian. Dari hasil pengujian menggunakan data pengujian dengan 105 pola data dapat diketahui total dari *SSE* yang dihasilkan sebesar 0,108777 dan nilai *MSE* sebesar 0,001036.

Hasil pengujian diperoleh berdasarkan nilai *error* yang tidak sesuai target (0) adalah 5 data, sedangkan 100 data lainnya sesuai dengan target (1). Hasil diperoleh jika nilai *error* dalam pengujian data  $\leq 0,05$  maka hasilnya benar (1) dan jika tidak maka hasilnya salah (0). Nilai  $\leq 0,05$  diperoleh berdasarkan penelitian (Andriani, Silitonga and Wanto, 2018).

## 3.4. Hasil Pengukuran Akurasi

Berdasarkan proses pengujian dengan data pengujian, diketahui bahwa dari 105 data

pengujian terdapat 5 data yang tidak sesuai dengan target, sedangkan 100 data lainnya sesuai dengan target. Sehingga dapat dihitung tingkat akurasi yang diperoleh pada proses pengujian dengan data pengujian dengan rumus persamaan sebagai berikut:

$$\text{Validasi (\%)} = \left[ \frac{\text{Jumlah\_data\_uji\_sesuai\_target}}{\text{Jumlah\_data\_uji}} \right] \times 100\% \quad (11)$$

$$\text{Validasi} = \frac{100}{105} \times 100\%$$

$$\text{Validasi} = 95,2\%$$

Setelah diketahui hasil pengujian menggunakan data pelatihan dan data pengujian, serta diperoleh tingkat akurasi sebesar 95,2% dari model arsitektur terbaik 3-35-1 yang digunakan pada proses pengujian, maka dapat diperoleh hasil prediksi dari data aktual (target) dan data prediksi keluaran jaringan pada Matlab R2019a untuk data pengujian tahun 2017 sampai dengan 2019 dengan perbandingan hasil prediksi sebagai berikut:

Tabel 4. Perbandingan hasil prediksi data pengujian dengan model terbaik 3-35-1

Tahun	Kabupaten/Kota	Data Aktual (%)	Data Prediksi (%)
2017	Kabupaten Cilacap	13,94	12,9908
	Kabupaten Banyumas	17,05	14,1235
	Kabupaten Purbalingga	18,80	15,2222
	Kabupaten Banjarnegara	17,21	17,0985
	Kabupaten Kebumen	19,60	13,8259
	Kabupaten Purworejo	13,81	12,8058
	Kabupaten Wonosobo	20,32	16,2337
	Kabupaten Magelang	12,42	11,5325
	Kabupaten Boyolali	11,96	10,7684
	Kabupaten Klaten	14,15	10,4229
	Kabupaten Sukoharjo	8,75	10,4160
	Kabupaten Wonogiri	12,90	11,3991
	Kabupaten Karanganyar	12,28	9,1280
	Kabupaten Sragen	14,02	10,8075
	Kabupaten Grobogan	13,27	12,4618
	Kabupaten Blora	13,04	14,2545
	Kabupaten Rembang	18,35	15,0325
	Kabupaten Pati	11,38	13,2063
	Kabupaten Kudus	7,59	3,4357
	Kabupaten Jepara	8,12	13,4182
	.	.	.
	.	.	.
2019	Kabupaten Blora	11,32	14,1683
	Kabupaten Rembang	14,95	13,3333
	Kabupaten Pati	9,46	12,1916
	Kabupaten Kudus	6,68	1,9461
	Kabupaten Jepara	6,66	11,5893

Tahun	Kabupaten/Kota	Data Aktual (%)	Data Prediksi (%)
	Kabupaten Demak	11,86	11,8270
	Kabupaten Semarang	7,04	10,6629
	Kabupaten Temanggung	9,42	12,4996
	Kabupaten Kendal	9,41	11,4307
	Kabupaten Batang	8,35	14,9749
	Kabupaten Pekalongan	9,71	14,2567
	Kabupaten Pemalang	15,41	16,0862
	Kabupaten Tegal	7,64	13,8552
	Kabupaten Brebes	16,22	14,4184
	Kota Magelang	7,46	11,2114
	Kota Surakarta	8,70	4,2106
	Kota Salatiga	4,76	-1,9360
	Kota Semarang	3,98	1,1218
	Kota Pekalongan	6,60	9,7053
	Kota Tegal	7,47	13,5800

Berdasarkan Tabel 4 dapat disimpulkan bahwa penerapan metode Jaringan Syaraf Tiruan *Backpropagation* dengan model arsitektur terbaik 3-35-1 dapat digunakan untuk melakukan prediksi pada data tingkat kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah berdasarkan data pengujian.

Pada hasil prediksi tersebut dapat diketahui bahwa data prediksi yang tidak sesuai dengan target (data aktual) adalah 5 data dari total 105 data pengujian, meliputi data tahun 2017 yaitu Kabupaten Kebumen, Kabupaten Brebes, dan Kota Semarang. Data tahun 2018 yaitu Kabupaten Kudus dan data tahun 2019 yaitu Kota Salatiga. Sedangkan 100 data lainnya sesuai dengan target.

Secara keseluruhan dari perbandingan data aktual dan data prediksi pada Tabel 4, masih terdapat beberapa data yang menghasilkan selisih yang cukup signifikan, seperti data tahun 2017 untuk Kabupaten Banyumas yang memperoleh data prediksi sebesar 14,1235%, sedangkan data aktual yang diperoleh sebesar 17,05%. Sehingga terdapat kurang lebih 8.637 ribu orang yang tidak terdeteksi sebagai penduduk miskin di Kabupaten Banyumas. Hal ini dikarenakan pada tahap pra proses atau normalisasi data yang dilakukan pada penelitian memperoleh hasil normalisasi yang kurang tepat sehingga akan berpengaruh terhadap hasil prediksi keluaran dari Jaringan Syaraf Tiruan *Backpropagation* pada Matlab R2019a.

Sehingga dari hasil pengujian tersebut dapat diperoleh tingkat akurasi sebesar 95,2% untuk model arsitektur terbaik 3-35-1. Dengan demikian, model arsitektur terbaik 3-35-1 selanjutnya dapat digunakan sebagai alternatif untuk melakukan prediksi tingkat kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah di masa mendatang.



#### 4. KESIMPULAN DAN SARAN

Berdasarkan hasil dan pembahasan, maka dapat diperoleh kesimpulan bahwa penerapan metode Jaringan Syaraf Tiruan *Backpropagation* dengan proses pelatihan menggunakan satu *hidden layer* dengan jumlah variasi *neuron* pada *hidden layer* 4, 10, 18, 25, 35, 50, 70, 100, serta menggunakan variasi *learning rate* 0,1 sampai dengan 0,5 dan konstanta momentum 0,5 sampai dengan 0,8, dan juga menggunakan parameter pelatihan berupa jumlah iterasi sebesar 10000 *epoch*, target *error* sebesar 0,001, *show* sebesar 1000, dan menggunakan fungsi aktivasi *sigmoid* (biner) dapat menghasilkan model arsitektur terbaik yaitu model 3-35-1. Model terbaik ini diperoleh dengan laju pembelajaran 0,4 dan konstanta momentum 0,8 dengan waktu komputasi selama 0:00:03 detik. Sehingga menghasilkan nilai korelasi antara *output* jaringan dan target sebesar 0,71022 dan nilai *MSE* hasil pengujian dengan data pelatihan sebesar 0,0009997/0,001 yang tercapai pada iterasi 3081 *epoch*. Berdasarkan perhitungan validasi dari model arsitektur terbaik 3-35-1 pada proses pengujian dengan data pengujian dapat diperoleh hasil tingkat akurasi sebesar 95,2%. Sehingga dapat disimpulkan penerapan metode Jaringan Syaraf Tiruan *Backpropagation* dengan model arsitektur terbaik 3-35-1 selanjutnya dapat digunakan sebagai alternatif untuk melakukan prediksi tingkat kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah di masa mendatang.

Saran yang dapat penulis berikan pada penelitian ini adalah perlu dilakukan tahap praproses atau normalisasi data dengan rumus persamaan yang lebih tepat agar diperoleh hasil normalisasi yang lebih sesuai, kemudian penelitian dapat dikembangkan lebih lanjut dengan berbasis *GUI* Matlab sebagai alternatif untuk melakukan prediksi tingkat kemiskinan menurut kabupaten/kota di Provinsi Jawa Tengah di masa mendatang.

#### DAFTAR PUSTAKA

- ANDRIANI, Y., SILITONGA, H. and WANTO, A. 2018. Analisis Jaringan Syaraf Tiruan untuk prediksi volume ekspor dan impor migas di Indonesia. *Register: Jurnal Ilmiah Teknologi Sistem Informasi*, 4(1), pp. 30–40. doi: 10.26594/register.v4i1.1157.
- ASHSHIDDIQI, A. J. R. 2018. Implementasi Jaringan Saraf Tiruan Backpropagation untuk Memprediksi Jumlah Penduduk Miskin di Indonesia dengan Optimasi Algoritme Genetika. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, 2(11), pp. 4638–4646.
- BADAN PUSAT STATISTIK. 2018. Profil Kemiskinan di Indonesia Maret 2019', pp. 1–8.
- BADAN PUSAT STATISTIK KABUPATEN CILACAP. 2019. *Indikator Pembangunan Daerah Kabupaten Cilacap Tahun 2019*.
- HARLIANA, H. and KIRONO, S. 2019. Penerapan Learning Vector Quantization Dalam Memprediksi Jumlah Rumah Tangga Miskin. *Jurnal Sains dan Informatika*, 5(2), pp. 118–127. doi: 10.34128/jsi.v5i2.192.
- IRAWAN, E., ZARLIS, M. and NABABAN, E. B. 2017. Analisis Penambahan Nilai Momentum Pada Prediksi Produktivitas Kelapa Sawit Menggunakan Backpropagation. *Jurnal Nasional Informatika dan Teknologi Jaringan*, 1(2), pp. 84–89.
- JAYA, H. et al. 2018. Kecerdasan Buatan, Fakultas MIPA Universitas Negeri Makassar.
- KAUNANG, F. J. 2018. Penerapan Algoritma J48 Decision Tree Untuk Analisis Tingkat Kemiskinan di Indonesia. *Cogito Smart Journal*, 4(2), pp. 348–357. doi: 10.31154/cogito.v4i2.141.348-357.
- KURNIAWATI, D. 2018. *Prediksi energi listrik dengan metode jaringan syaraf tiruan backpropagation menggunakan matlab untuk kota semarang tahun 2019 – 2024*.
- MULYANI, A. 2016. Analisis Neural Network Struktur Backpropagation Sebagai Metode Peramalan Pada Perhitungan Tingkat Kemiskinan di Indonesia. *Jurnal Techno Nusa Mandiri*, XIII(1), pp. 9–15.
- PRAYOGA, A. M. B. 2019. *Pengaruh Pertumbuhan Ekonomi, Pengangguran Dan Belanja Pemerintah Terhadap Kemiskinan di Daerah Istimewa Yogyakarta Periode 2003 – 2018*. doi: 10.1017/CBO9781107415324.004.
- PUGUH BUDRO IRAWAN, M.A, D. H. U. et al. 2016. *Official Statistics Sosial - Kependudukan Dasar*. 2016th edn. Bogor: IN MEDIA.
- PUTRI, S. R. S. 2019. *Analisis Pengaruh Jumlah Penduduk, Pengangguran, dan Pengeluaran Pemerintah Terhadap Tingkat Kemiskinan Di Indonesia Tahun 1990-2018*. doi: 10.5281/zenodo.1477753.
- RAHMAWATI, R., SAFITRI, D. and FAIRUZDHIYA, O. U. 2015. Analisis Spasial Pengaruh Tingkat Pengangguran

Terhadap Kemiskinan di Indonesia', *Media Statistika*, 8(1), pp. 23–30.

- RAHUL, M. *et al.* 2020. Analisa JST Untuk Memprediksi Pembuatan SIM Menggunakan Metode Algoritma Backpropagation. *Jurnal Media Informatika Budidarma*, 4(1), pp. 124–128. doi: 10.30865/mib.v4i1.1742.
- SANTOSO, A. and HANSUN, S. 2019. Prediksi IHSG dengan Backpropagation Neural Network. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 3(2), pp. 313–318. doi: 10.29207/resti.v3i2.887.
- SATRIA, B. 2018. Prediksi Volume Penggunaan Air PDAM Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 2(3), pp. 674–684. doi: 10.29207/resti.v2i3.575.
- SIHOTANG, B. K. and WANTO, A. 2018. Analisis Jaringan Syaraf Tiruan Dalam Memprediksi Jumlah Tamu Pada Hotel Non Bintang', *Techno.COM*, 17(4), pp. 333–346.
- SUPRIYADI, D. 2013. Sistem Informasi Penyebaran Penyakit Demam Berdarah Menggunakan Metode Jaringan Syaraf Tiruan Backpropagation. *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*, 1(3), pp. 6–11. doi: 10.21456/vol1iss3pp159-167.
- WANTO, A. 2018. Penerapan Jaringan Saraf Tiruan Dalam Memprediksi Jumlah Kemiskinan Pada Kabupaten/Kota Di Provinsi Riau. *Klik - Kumpulan Jurnal Ilmu Komputer*, 5(1), pp. 61–74. doi: 10.20527/klik.v5i1.129.
- WINDARTO, A. P., Lubis, M. R. and Solikhun, S. 2018. Implementasi JST Pada Prediksi Total Laba Rugi Komprehensif Bank Umum Konvensional Dengan Backpropagation. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 5(4), pp. 411–418. doi: 10.25126/jtiik.201854767.
- YANTO, M. *et al.* 2018. Peramalan Penjualan Pada Toko Retail Menggunakan Algoritma Backpropagation Neural Network. *Jurnal Media Informatika Budidarma*, 2(3), pp. 110–117. doi: 10.30865/mib.v2i3.811.