

## Implementasi Metode Jaringan Saraf Tiruan *Backpropagation* Pada Prediksi *Payload* 4G di Telkomsel Jember

M. Rikzal Humam Al Kholili<sup>1</sup>, Budi Darma Setiawan<sup>2</sup>, Randy Cahya Wihandika<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>erick.alkholili12@gmail.com, <sup>2</sup>budidarma@ub.ac.id, <sup>3</sup>rendicahya@ub.ac.id

### Abstrak

PT. Telkomsel merupakan salah satu *provider* telekomunikasi terbesar di Indonesia yang sekaligus memiliki pelanggan paling banyak yang tersebar di seluruh Indonesia. Pelanggan PT. Telkomsel dari tahun ke tahun mengalami terjadinya peningkatan dan hal ini akan mengakibatkan pemakaian jumlah *payload* yang terus bertambah karena *payload* adalah semua paket yang diterima dan dikirim oleh *handphone* ke suatu receiver (penerima sinyal) serta jika jumlah pemakaian *payload* lebih kecil dari jumlah pemakai maka akan terjadi *over lagging* dan pemakai akan merasa tidak nyaman. Dengan permasalahan tersebut, maka dibutuhkan sebuah implementasi dari beberapa metode prediksi jumlah pemakaian *payload* 4G agar PT. Telkomsel dapat mengetahui jumlah pemakaian *payload* 4G di hari atau bulan kemudian sehingga dapat mengantisipasi terjadinya kerugian ataupun komplain dari pelanggan. Dari sekian banyak metode prediksi yang ada, penulis menggunakan metode jaringan saraf tiruan *backpropagation* untuk melakukan proses prediksi dengan menggunakan arsitektur jaringan saraf tiruan 4 neuron *input node*, 6 neuron *hidden node* dan 1 neuron *output node*. Dengan menggunakan perhitungan MAPE (*Mean Absolute Percentage Error*) didapatkan nilai paling optimal sebesar 6,0154830745999%.

**Kata kunci:** *implementasi, jaringan saraf tiruan, backpropagation, payload 4G*

### Abstract

PT. Telkomsel is one of the largest telecommunication providers in Indonesia which also has the most customers spread throughout Indonesia. Customers of PT. Telkomsel from year to year has experienced an increase and this will result in the use of an increasing number of payloads because the payload is all packages that are received and sent by mobile to a receiver (signal receiver) and if the amount of payload usage is smaller than the number of users it will occur over lagging and users will feel uncomfortable. With these problems, it requires an implementation of several methods to predict the amount of 4G payload usage so that PT. Telkomsel can find out the number of 4G payload usage in the next day or month so that it can anticipate losses or complaints from customers. Of the many prediction methods available, the authors use the backpropagation neural network method to perform a prediction process using artificial neural network architecture 4 input node neurons, 6 hidden node neurons and 1 node output neuron. By using the MAPE calculation (*Mean Absolute Percentage Error*) the most optimal value is 6.0154830745999%.

**Keywords:** *implementation, neural networks, backpropagation, payload 4G*

### 1. PENDAHULUAN

PT. Telekomunikasi Seluler Indonesia merupakan *provider* telekomunikasi seluler yang paling besar dan paling banyak menyebar di seluruh kawasan Indonesia hingga daerah yang terpencil sekalipun, hal ini dapat

dibuktikan pada halaman resmi telkomsel yaitu [www.telkomsel.com](http://www.telkomsel.com) (2017) yang menyatakan bahwa banyaknya pelanggan Telkomsel yang mencapai 178 juta pelanggan serta 146 ribu BTS yang tersebar di seluruh pelosok Indonesia.

Sudah sepatutnya PT. Telekomunikasi Seluler memiliki jumlah pelanggan yang

semakin bertambah, hal ini dapat dibuktikan pada situs <https://id.techinasia.com> (2014) yang menyatakan bahwa pada kuartal 1 tahun 2014 memiliki pengguna sebanyak 132,7 juta dan meningkat pada kuartal 3 tahun 2014 yakni sebanyak 139,2 juta pengguna. Pertambahan pelanggan tersebut dapat mempengaruhi pertambahan penggunaan *payload* 4G yang mengakibatkan gangguan jika tidak dilakukan *upgrade* sistem dengan segera. Sudah sepatutnya bagi pihak Telkomsel dapat memprediksi jumlah penggunaan *payload* 4G, sehingga pihak Telkomsel dapat merencanakan dan mengetahui site mana yang perlu dilakukan *upgrade* sistem di kemudian hari.

Prediksi adalah suatu cara untuk memperkirakan suatu hal yang akan terjadi di masa yang akan datang (Taylor, 2003). Untuk dapat memprediksi suatu keadaan yang akan terjadi, terdapat beberapa metode yang dapat melakukan prediksi dengan menggunakan perhitungan matematis berdasarkan rumus yang ada. Berbagai metode tersebut ialah metode *fuzzy time series* yang digunakan oleh Elfajar (2017) untuk kasus peramalan jumlah kunjungan wisatawan di Kota Batu. Peneliti menggunakan data pengunjung perbulan yang diperoleh dari Dinas Pariwisata Kota Batu. Berdasarkan hasil pengujian yang sudah dilakukan oleh peneliti dengan menggunakan *Average Forecasting Error Rate* (AFER) menghasilkan nilai error rata-rata sebesar 0,0056% dengan menggunakan data latih 60. Metode lain yang dapat digunakan untuk meramalkan suatu permasalahan adalah metode *support vector regression* oleh Rahyani (2018) untuk kasus prediksi jumlah pengunjung pariwisata. Data yang digunakan adalah jumlah data pengunjung pariwisata bulanan sebanyak 72 data dari tahun 2010 sampai 2015. Berdasarkan pengujian yang dilakukan oleh peneliti menghasilkan nilai MAPE terbaik yaitu 6,98% dengan parameter  $\sigma = 925,8409$ ,  $\lambda = 0,3868$ ,  $cLR = 0,0802$ ,  $\epsilon = 1,27E-10$ ,  $complexity = 3234,539$ , jumlah iterasi maksimal 5000.

Metode jaringan saraf tiruan *backpropagation* merupakan salah satu metode yang dapat digunakan untuk memprediksi suatu kondisi yang ada. Ada beberapa penelitian yang menggunakan metode jaringan saraf tiruan untuk peramalan suatu kondisi. Rachman (2018) melakukan penelitian tentang peramalan produksi gula dengan menggunakan metode jaringan saraf tiruan *backpropagation* dengan menggunakan 4 *input layer*, 3 *hidden layer* serta

1 *output layer* dan menghasilkan nilai MAPE terendah sebesar 17,85% dengan 800 iterasi pada pengujian jumlah iterasi maksimum serta nilai MAPE terendah sebesar 17,38 % dengan nilai *learning rate* 0,4 pada pengujian *learning rate*. Penelitian yang serupa dilakukan oleh Susanti (2014) yang membahas tentang penerapan model *neural network backpropagation* untuk prediksi harga ayam dengan data bertipe *time series* dari bulan agustus 2010 sampai bulan mei 2013. Peneliti menggunakan arsitektur jaringan 4 *input layer*, 10 *hidden layer* dan 1 *output layer* yang menghasilkan nilai MSE sebesar 0,0113 dengan menggunakan *learning rate* 0,05 dan maksimal iterasi sebanyak 5000.

## 2. LANDASAN KEPUSTAKAAN

### 2.1 4G LTE ( Long Term Evolution)

Long Term Evolution (LTE) adalah sebuah teknologi perkembangan dari 3G (Third Generation) yang memiliki transfer data rate yang lebih besar dari generasi yang sebelumnya. Teknologi LTE masih belum sepenuhnya bisa disebut teknologi 4 Generation tepatnya masih 3,9G. Kelebihan dari LTE diantaranya (Ariyanti, 2013):

- Latency/delay lebih rendah
- Data rate lebih tinggi
- Meningkatkan kapasitas dan coverage
- Cost-reduction

WCMA adalah teknologi yang dikembangkan sebelum adanya teknologi LTE pada tahun 2003/2004 yang memiliki data rate sekitar 384 kbps downlink, 128 kbps uplink dan round trip time 150 ms. Setelah WCMA dibangun, pada tahun 2005/2006 muncul teknologi HSDPA (High Speed Downlink Packet Access) yang memiliki data rate downlink mencapai 14 Mbps, uplink hingga 5.7 Mbps. Kemudian pada tahun 2008/2009 3GPP merelease teknologi HSPA+ dengan data rate mencapai 28Mbps untuk downlink dan uplink sebesar 11 Mbps. Teknologi yang terakhir yakni LTE yang di release pada tahun 2010 yang memiliki data rate uplink sebesar 150Mbps. Teknologi LTE menguntungkan bagi perangkat mobile dan home user karena memiliki transmisi yang berkecepatan 100 Mbps, dan juga mungkin home user akan menggunakan koneksi telepon untuk alternating kedua karena jangkauan LTE lebih jauh (Ariyanti, 2013).

### 2.2 Payload

Ketika data dikirim melalui internet, setiap unit yang ditransmisikan mencakup informasi *header* dan data actual yang dikirim. *Header* mengidentifikasi sumber dan tujuan paket, sedangkan data sebenarnya disebut *payload*. Karena informasi header, atau data overhead, hanya digunakan dalam proses transmisi, kemudian dilepaskan dari paket saat mencapai tujuannya. Oleh karena itu *payload* adalah satu-satunya data yang diterima oleh sistem tujuan. Informasi pemberian sinyal dan kontrol digunakan untuk memantau, mengawasi, memelihara, dan mengatur dan mengendalikan jaringan dan komponennya. Data di atas digunakan untuk tujuan seperti penanganan, manajemen kemacetan, kontrol kesalahan, indikasi prioritas, dan panggilan routing, blok, frame, paket, dan sel yang berisi muatan.

Data-data yang perlu atau yang penting dibawa dengan paket atau transmisi yang lain. *Payload* tidak termasuk data "overhead" yang dibutuhkan untuk mendapatkan paket ke tujuannya. Untuk lapisan komunikasi yang membutuhkan beberapa data overhead untuk melakukan tugasnya, *payload* kadang-kadang dianggap mencakup bagian data overhead yang ditangani lapisan ini. Namun, dalam penggunaan yang lebih umum, *payload* adalah bit yang dikirimkan ke pengguna akhir di tempat tujuan.

### 2.3 Jaringan Saraf Tiruan

Jaringan saraf tiruan (JST) adalah suatu metode dalam ruang lingkup *Artificial Intelligence* (AI) yang memiliki arsitektur kerja menyerupai dengan otak manusia sehingga dapat menyelesaikan masalah yang kompleks dan memiliki pola – pola yang sulit dipahami. Pembangunan arsitektur jaringan saraf tiruan meniru prinsip kerja dari saraf biologi manusia yang dikembangkan menggunakan model matematis yang digeneralisasikan berdasarkan saraf dan kesadaran manusia (Amina dan Irawan, 2013).

### 2.4 Algoritme Backpropagation

*Backpropagation* adalah salah satu metode dari jaringan saraf tiruan yang dapat digunakan untuk masalah prediksi. Pada proses pelatihan dalam metode *backpropagation*, bobot yang digunakan untuk penghubung antar lapisan akan selalu dilakukan *update* agar mendapatkan bobot yang sesuai untuk permasalahan yang dipakai.

Pada proses metode *backpropagation* terdapat 3 fase (Siang, 2005), yaitu:

#### a. Fase 1 *Feedforward* (Propagasi maju)

Pada fase propagasi maju, akan dilakukan perhitungan dari layar masukkan menuju layar tersembunyi dan dilanjutkan ke layar keluaran dengan menggunakan fungsi aktivasi yang sudah ditentukan.

#### b. Fase 2 *Backpropagation* (Propagasi mundur)

Fase propagasi mundur merupakan fase perhitungan *error* yang dihasilkan dari perhitungan propagasi maju. Nilai keluaran yang dihasilkan oleh propagasi maju akan dibandingkan dengan nilai target yang sudah ditentukan, kemudian semua yang berhubungan dengan layar keluaran akan dilakukan perhitungan *error*.

#### c. Fase 3 *Weight Update* (Perubahan bobot)

Pada fase perubahan bobot, bobot yang dihasilkan secara acak pada saat inisialisasi bobot, akan dilakukan perubahan pada fase 3 berdasarkan dengan nilai *error* yang dihasilkan pada saat propagasi mundur.

Proses perhitungan yang melewati propagasi maju, propagasi mundur dan perubahan bobot akan dilakukan secara berulang-ulang sampai nilai *error* sudah mencapai nilai yang sudah ditentukan atau nilai iterasi sudah mencapai pada jumlah iterasi yang ditentukan.

1. Melakukan inisialisasi bobot secara acak dengan rentang nilai antara 0 sampai dengan 1. Menentukan nilai *learning rate* dan iterasi.
2. Melakukan langkah 2 – 8 jika belum mencapai *stop condition*
3. Melakukan langkah 3 – 8 untuk perhitungan setiap data latih

#### Fase 1 : *Feedforward*

4. Setiap *input*  $x_i$  menerima sinyal dan meneruskan ke setiap unit di layar selanjutnya
5. Menghitung nilai  $z_{netj}$  di setiap layar tersembunyi dengan menggunakan persamaan (1)

$$z_{netj} = v_{0j} + \sum_{i=0}^n x_i \cdot v_{ij} \quad (1)$$

Dengan menggunakan fungsi sigmoid yang ada pada persamaan (2) akan didapatkan nilai  $z_j$

$$z_j = f(z_{net_j}) = \frac{1}{1 + \exp(-z_{net_j})} \quad (2)$$

Keterangan:

$z_{net_j}$  = hasil perhitungan antara input  $x_i$  dengan bobot  $v_{ij}$

$v_{0j}$  = bobot bias layar masukkan dengan layar tersembunyi

$x_i$  = sinyal masukkan ke - I ( $i=1,2,\dots$ )

$v_{ij}$  = bobot antara sinyal  $x_i$  dengan  $z_{net_j}$

$z_j$  = hasil dari  $z_{net_j}$  yang sudah dilakukan aktivasi

$i$  = Indeks pada unit layar masukkan

$j$  = Indeks pada unit layar tersembunyi

- Menghitung setiap nilai  $y_{net_k}$  yang ada pada layar *output* ( $k = 1, 2 \dots n$ )

$$y_{net_k} = w_{0k} + \sum_{j=1}^p z_j \cdot w_{jk} \quad (3)$$

Agar mendapatkan nilai  $y_k$  maka digunakan fungsi aktivasi sigmoid yang ditunjukkan pada persamaan 4

$$y_k = f(y_{net_k}) = \frac{1}{1 + \exp(-y_{net_k})} \quad (4)$$

Keterangan:

$y_{net_k}$  = Hasil perhitungan antara nilai *hidden layer* dengan bobot  $w_{jk}$

$w_{0k}$  = Bobot bias pada *hidden layer*

$z_j$  = Nilai node *hidden layer* yang sudah diaktivasi

$w_{jk}$  = Bobot antara *hidden layer* dengan *output layer*

$y_k$  = Nilai *output layer* yang sudah diaktivasi

$k$  = Jumlah node di *output layer*

### Fase 2 : Backpropagation

- Menghitung nilai faktor  $\delta$  pada setiap indeks  $k$  berdasarkan *error* pada setiap *output layer*

$$\delta_k = (t_k - y_k) f'(y_{net_k}) = (t_k - y_k) y_k (1 - y_k) \quad (5)$$

Keterangan:

$\delta_k$  = Faktor perbaikan nilai bobot yang terhubung dengan *output layer*

$y_{net_k}$  : Nilai *output layer* sebelum diaktivasi

$t_k$  : Nilai target prediksi

$y_k$  : Nilai yang dihasilkan dari perhitungan *feedforward*

$k$  : Indeks *output layer*

$\delta_k$  digunakan untuk menghitung nilai perubahan bobot  $w_{jk}$  yang digunakan untuk meng-*update* bobot  $w_{jk}$  nanti.

Menghitung nilai perubahan bobot dengan menggunakan percepatan alpha (*learning rate*)

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

Keterangan:

$\Delta w_{jk}$  = Nilai perubahan bobot  $w_{jk}$

$\alpha$  = Nilai *learning rate*

$\delta_k$  = Faktor perbaikan bobot pada *output layer*

$z_j$  = Nilai *output layer* yang sudah diaktivasi

- Menghitung nilai  $\delta_{net_j}$  pada setiap indeks  $j$  berdasarkan *error* pada *hidden layer* ( $j = 1, 2, 3 \dots m$ )

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

Mengitung nilai  $\delta_j$

$$\delta_j = \delta_{net_j} \cdot f'(z_{net_j}) = \delta_{net_j} \cdot z_j (1 - z_j) \quad (8)$$

Menghitung nilai  $\Delta v_{ij}$  yang nantinya akan digunakan untuk melakukan *update* bobot  $v_{ij}$  yang baru

$$\Delta v_{ji} = \alpha \delta_j x_i \quad (9)$$

Keterangan:

$\delta_{net_j}$  = Faktor perbaikan pada nilai bobot di  $z_j$

$\delta_k$  = Faktor perbaikan nilai bobot pada *output layer*

$w_{jk}$  = Nilai bobot antara *hidden layer* dengan *output layer*

$\delta_j$  = Faktor perbaikan nilai bobot pada *hidden layer*

$z_j$  = Nilai sinyal pada *hidden layer*

$\Delta v_{ji}$  = Perubahan bobot antara *input layer* dengan *hidden layer*

$\alpha$  = Nilai *learning rate*

$x_i$  = Nilai sinyal pada *input layer*

### Fase 3 : Weight Update

- Menghitung semua perubahan bobot pada setiap *layer*. Persamaan untuk meng-*update* bobot antara *input layer* dengan *hidden layer*

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (10)$$

Keterangan:

$V_{ij}(\text{baru})$  = bobot baru antara *input layer* dengan *hidden layer*

$V_{ij}(\text{lama})$  = bobot lama antara *input layer* dengan *hidden layer*

$\Delta v_{ij}$  = Nilai perubahan bobot antara *input layer* dengan *hidden layer*

Persamaan untuk meng-*update* bobot antara *hidden layer* dengan *output layer*

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (11)$$

Keterangan:

$W_{jk}(\text{baru})$  = Bobot baru antara *hidden layer* dengan *output layer*

$W_{jk}(\text{lama})$  = Bobot lama antara *hidden layer* dengan *output layer*

$\Delta w_{jk}$  = Nilai perubahan bobot antara *hidden layer* dengan *output layer*

- Memeriksa kondisi berhenti sudah terpenuhi atau belum. Berikut ini beberapa kondisi berhenti yang digunakan, adalah:

- a. Menentukan kondisi berhenti berdasarkan nilai *error* yang sudah ditentukan di awal.
- b. Menentukan kondisi berhenti berdsarkan jumlah iterasi yang sudah ditentukan di awal.

**2.7 Mean Square Error**

*Mean Square Error* (MSE) merupakan salah satu perhitungan yang digunakan untuk mengetahui nilai *error* dari sebuah metode. *Mean Square Error* akan menghitung nilai hasil prediksi dengan cara mengurangkan nilai yang sebenarnya dengan nilai hasil prediksi. Selengkapnya dapat dilihat pada persamaan (1).

$$MSE = \frac{\sum(t-y_k)^2}{n} \quad (1)$$

Keterangan:

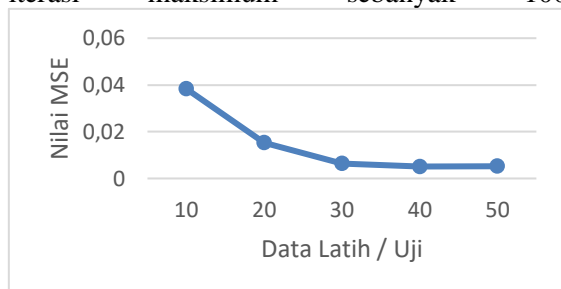
- t* : Nilai *actual*
- y<sub>k</sub>* : Nilai hasil prediksi
- k* : Indeks untuk unit *output*
- n* : Jumlah data latih

**3. PENGUJIAN DAN ANALISIS**

Peneliti akan melakukan pengujian sebanyak 4 kali, yaitu pengujian *learning rate*, pengujian iterasi, pengujian data dan pengujian *k-fold validation*. Setelah pengujian dilakukan maka peneliti melakukan analisa dari hasil yang dihasilkan dari pengujian.

**3.1 Pengujian Data Terhadap nilai MSE**

Pengujian data dilakukan agar dapat mengetahui jumlah data atau data ke berapa saja yang cocok dengan metode *backpropagation*. Parameter yang digunakan adalah data latih dari 10 data latih sampai 50 data latih serta data uji sebanyak 10. Parameter lain yang digunakan adalah nilai *learning rate* sebesar 0.1 dan jumlah iterasi maksimum sebanyak 100.



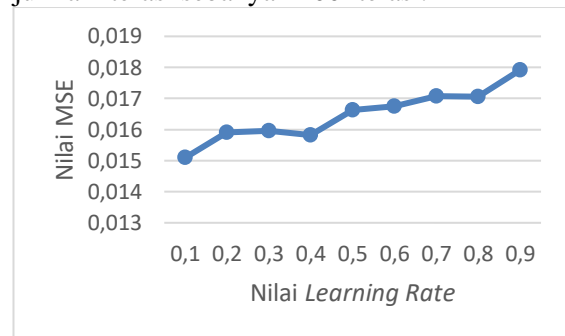
Gambar 2. Grafik Pengujian Data Terhadap Nilai MSE

Dari Gambar 2 dapat dilihat bahwa nilai MSE terbaik didapatkan pada jumlah data 40 dengan 10 data uji, dan pada jumlah data

sebanyak 50 mengalami penurunan nilai MSE namun tidak terlalu signifikan.

**3.2 Pengujian Learning Rate Terhadap Nilai MSE**

Pengujian nilai *learning rate* digunakan untuk menentukan nilai *learning rate* yang terbaik untuk metode *Backpropagation* yaitu dari nilai 0.1 sampai dengan nilai 0.9. Parameter yang digunakan adalah jumlah data latih sebanyak 55 data latih dengan 7 data uji dan jumlah iterasi sebanyak 100 iterasi.

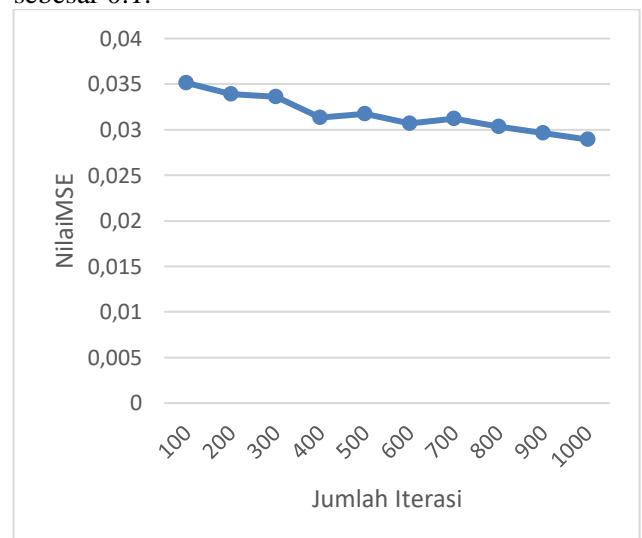


Gambar 3. Grafik Pengujian Learning Rate

Dari Gambar 3 terlihat dengan jelas bahwa semakin bsar nilai *learning rate* maka akan menghasilkan nilai MSE yang semakin besar. Jadi nilai *learning rate* terbaik terdapat pada nilai *learning rate* sebesar 0.1.

**3.3 Pengujian Iterasi Terhadap Nilai MSE**

Pengujian iterasi dilakukan dengan menggunakan sebanyak 30 data latih tanpa menggunakan data uji dan nilai *learning rate* sebesar 0.1.



Gambar 4. Grafik Pengujian Iterasi Terhadap Nilai MSE Latih

Dari Gambar 4 dapat dilihat bahwa iterasi

dengan nilai MSE terkecil jatuh pada jumlah iterasi maksimum sebesar 1000. Hal ini dikarenakan pada metode *Backpropagation* proses pelatihan akan terus memperbaiki bobot hingga iterasi yang ditentukan agar memperoleh bobot dengan nilai MSE terkecil.

### 3.4 Pengujian K-Fold Validation Terhadap Nilai MSE

Dari 58 total data yang ada dengan menggunakan pengujian *K-Fold Validation* maka dibagi menjadi 3 bagian yaitu 20 data, 20 data dan 18 data. Selengkapnya dapat dilihat pada Tabel 1.

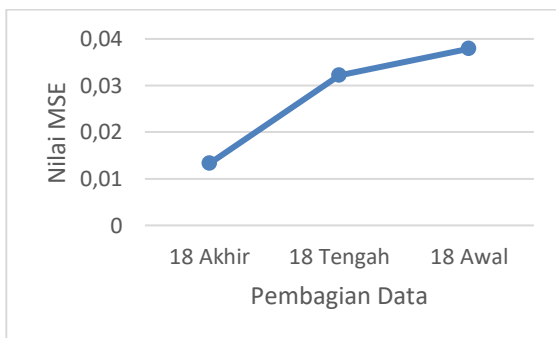
Tabel 1. Pembagian Data

	Jumlah		
akhir	20	20	18
tengah	20	18	20
awal	18	20	20

Keterangan :

Angka warna biru : data latih

Angka warna hijau : data uji

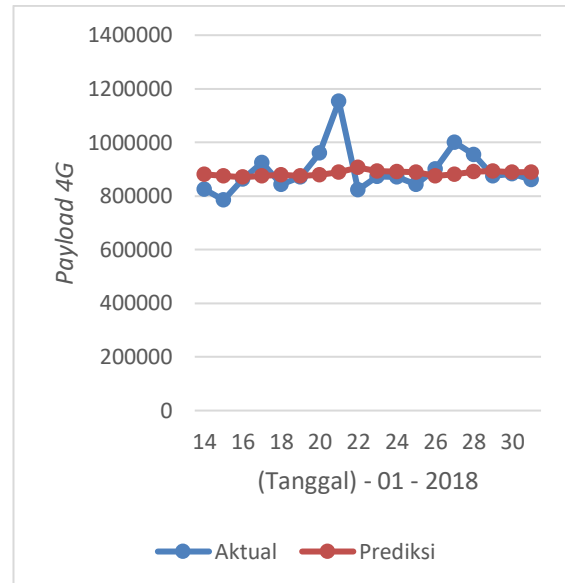


Gambar 5. Grafik Pengujian *K-Fold Validation* Terhadap Nilai MSE

Dari Gambar 5 dapat diketahui bahwa data terbaik terletak pada 40 data awal sebagai data latih sedangkan 18 data terakhir sebagai data uji dengan memperoleh nilai MSE terkecil.

### 3.5 Pengujian Hasil Prediksi

Dengan menggunakan nilai variable yang sudah didapatkan pada hasil pengujian sebelumnya, maka pengujian hasil prediksi dapat dilakukan. Pengujian hasil prediksi menggunakan nilai *learning rate* sebesar 0.1, iterasi maksimum sebanyak 1000, 40 data awal sebagai data latih dan 18 data akhir sebagai data uji.



Gambar 6. Grafik Pengujian Hasil Prediksi

Dari Gambar 6 dapat diketahui bahwa metode *Backpropagation* cukup baik dalam memprediksi *payload* 4G di Telkomsel Jember. Hal ini dibuktikan dengan nilai prediksi hampir mendekati dengan nilai aktual yang menghasilkan nilai MAPE (*Mean Absolute Percentage Error*) sebesar 6,0154830745999%.

## 4. KESIMPULAN

Setelah penelitian dilakukan peneliti dapat mengambil sebuah kesimpulan bahwa metode *Backpropagation* cukup baik untuk kasus prediksi *payload* 4G di Telkomsel Jember dengan menghasilkan nilai MAPE (*Mean Absolute Percentage Error*) sebesar 6,0154830745999%. Dengan menggunakan nilai *learning rate* sebesar 0.1, jumlah iterasi maksimum sebanyak 1000 dan 40 data awal sebagai data latih serta 18 data akhir sebagai data uji.

## 5. REFERENCES

- Amina, F. & Irawan, M. I., 2013. Prediksi Jumlah Penduduk Miskin Di Provinsi Kalimantan Selatan Menggunakan Jaringan Saraf Tiruan Backpropagation. *Seminar Nasional Pasca Sarjana XIII – ITS*.
- Ariyanti, S., 201. Pemanfaatan Digital Dividend untuk layanan Long Term Evolution Area (LTE). *Puslitbang Sumber Daya dan perangkat Pos dan Informatika*, 11(3), pp. 189-208.

- Diputra, M. I., 2018. Prediksi Tingkat Keuntungan Usaha Peternakan Itik Alabio Petelur menggunakan Jaringan Saraf Tiruan Backpropagation (Kasus di Kabupaten Hulu Sungai Utara Kalimantan Selatan). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(2), pp. 785-792.
- Elfajar, A. B., Setiawan, B. D. & Dewi, C., 2017. Peramalan Jumlah Kunjungan Wisatawan Kota Batu Menggunakan Metode Time Invariant Fuzzy Time Series. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Februari, 1(2), pp. 85-94.
- Elfajar, A. B., Setiawan, B. D. & Dewi, C., 2017. Peramalan Jumlah Kunjungan Wisatawan Kota Batu Menggunakan Metode Time Invariant Fuzzy Time Series. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 1(2), pp. 85-94.
- Haykin, S., 2009. *Neural Networks and Learning Machines*. United State of America: Pearson.
- Ika, R. C., 2018. Optimasi Fuzzy Time Series Menggunakan Algoritme Particle Swarm Optimization untuk Peramalan Nilai Pembayaran Penjaminan Kredit Macet. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(6), pp. 2364-2373.
- Rachman, A. S., 2018. Peramalan Produksi Gula Menggunakan Metode Jaringan Saraf Tiruan Backpropagation Pada PG Candi Baru Sidoarjo. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, April, 2(4), pp. 1683-1689.
- Raharyani, M. P., 2018. Implementasi Algoritme Support Vector Regression Pada Prediksi Jumlah Pengunjung Pariwisata. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, April, 2(4), pp. 1501-1509.
- Riadi, M., 2016. *Jaringan Saraf Tiruan* [Online] Available at: <https://www.kajianpustaka.com/2016/11/jaringan-saraf-tiruan-jst.html>
- Rindhani, M Najmi; Putri, Rekyan Regasari Mardi; Sutopo, 2017. Peramalan Dosis Pupuk Berdasarkan Karakteristik dan Lingkungan Tanaman Jeruk Siam Menggunakan Metode Backpropagation. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, November, 1(11), pp. 1214-1223.
- Siang, J. J., 2005. *Jaringan Saraf Tiruan & Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi.
- Sinaga, R. F. P., Setiawan, B. D. & Marji, 2018. Peramalan Produksi Kelapa Sawit Menggunakan Jaringan Saraf Tiruan Dengan Metode Backpropagation (Studi Kasus PT.Sandabi Indah Lestari). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2(11), pp. 4613 - 4620.
- Susanti, N., 2014. *PENERAPAN MODEL NEURAL NETWORK BACKPROPAGATION UNTUK PREDIKSI HARGA AYAM*. Jawa Tengah, s.n., pp. 325-332.
- Taylor, J., 2003. Short-Term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing. *Journal of Operational Research Society*, pp. 799-805.
- Telkomsel, 2017. *Sejarah Telkomsel* [Online] Available at: <https://www.telkomsel.com/about-us/our-story/our-history>
- Yohanes, E., Mahmudy, W. F. & Rahmi, A., 2015. PENENTUAN UPAH MINIMUM KOTA BERDASARKAN TINGKAT INFLASI MENGGUNAKAN BACKPROPAGATION NEURAL NETWORK (BPNN). *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 04, 2(1), pp. 34-40.