

Prediksi Jumlah Permintaan Koran Menggunakan Metode Jaringan Syaraf Tiruan *Backpropagation*

Nabilla Putri Sakinah¹, Imam Cholissodin², Agus Wahyu Widodo³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹nabillaputri008@gmail.com, ²imamcs@ub.ac.id, ³a_wahyu_w@ub.ac.id

Abstrak

Di era globalisasi seperti saat ini, kebutuhan masyarakat akan informasi semakin tahun semakin meningkat. Hal ini dapat dilihat dari perilaku masyarakat dalam merespon berbagai peristiwa yang terjadi, baik global maupun nasional. Banyaknya media yang tersedia memberikan kemudahan bagi masyarakat untuk mendapatkan informasi secara aktual. Media yang ada salah satunya adalah media cetak. Media cetak dianggap media yang memiliki aktualitas dan akurasi yang dapat dipercaya. Salah satu contoh media cetak yaitu salah satunya surat kabar atau koran. Koran berfungsi sebagai sarana informasi dan edukasi yang sampai sekarang masih digunakan dari berbagai kalangan masyarakat. Banyak metode peramalan yang telah digunakan untuk meramalkan data seperti Jaringan Syaraf Tiruan *Backpropagation* yang terbukti pada beberapa kasus peramalan dapat memberikan hasil peramalan yang baik, seperti pada kasus peramalan konsumsi air, konsumsi curah hujan, nilai tukar Dollar dan peramalan beban listrik. Sesuai dengan pengujian yang dilakukan dengan menggunakan data jumlah penjualan koran Radar Madura tahun 2015 menghasilkan jumlah iterasi terbaik adalah sebesar 200, dan *learning rate* terbaik adalah sebesar 0.6, dan pengujian data latih dan data uji menghasilkan nilai data latih terbaik yaitu 100 dan data uji 10. Dan menghasilkan nilai *error rate* terkecil yaitu 0.0162.

Kata kunci: peramalan koran, jaringan syaraf tiruan, *backpropagation*, media cetak, koran, *error rate*

Abstract

In the era of globalization, community needs for information is increasing by years. This can be seen from the behavior of the community in responding to everything, both global and national. The quantity of media provided convenience for the community to get information actually. Print media is one of media which has actuality and accuracy which can be trusted. One example of media is print media or newspaper. Newspaper is information tool and educational tool which until now still useful for every community. There are many forecasting methods that have been used to predict which is proven in some forecasting and providing the good result, for example forecasting of water consumption, rainfall consumption, the exchange rate of dollar and forecasting electrical load. In accordance with the tests conducted using the data sales of Radar Madura in 2015, resulted the best iterations is 200, and the value of learning rate is 0.6, and the test of training data and test data yields the best value of training data is 100 and test data 10. With error rate 0.0162.

Keywords: newspaper forecasting, artificial neural network, *backpropagation*, print media, newspaper, *error rate*

1. PENDAHULUAN

Media cetak dianggap media yang memiliki aktualitas dan akurasi yang dapat dipercaya. Salah satu contoh media cetak yaitu salah satunya surat kabar atau koran. Koran berfungsi sebagai sarana informasi dan edukasi yang sampai sekarang masih digunakan dari berbagai kalangan masyarakat. Berdasarkan hasil penelitian Roy Morgan International Research

pada tahun 2014, koran Jawa Pos menjadi media cetak di Indonesia yang paling diminati oleh pembaca berdasarkan survei yang dilakukan Roy Morgan di 17 provinsi, 22 kota besar dan 23 kota kecil. Dalam riset Roy Morgan, Jawa Pos saat ini dibaca rata-rata 1,4 juta orang sehingga mampu menjadi yang terbaik mengalahkan pesaingnya.

Jumlah permintaan koran selalu berubah dikarenakan banyak faktor internal maupun eksternal. Sehingga, ini merupakan salah satu

tantangan bagi perusahaan. Jumlah produksi koran sangatlah penting dikarenakan jika perusahaan tidak bisa memperkirakan jumlah produksi koran, maka akan terjadi resiko kerugian dikarenakan kelebihan jumlah produksi sehingga koran yang tidak terjual akan dijual separuh harga maupun kerugian dikarenakan kekurangan jumlah produksi.

Sehingga diperlukan suatu sistem cerdas yang mampu memprediksi jumlah barang yang akan diproduksi. Sudah banyak metode yang digunakan untuk permasalahan prediksi atau peramalan. Salah satunya adalah Jaringan Syaraf Tiruan (JST) yang mengadopsi sistem pembelajaran pada otak manusia. Pada penelitian ini, penulis mengaplikasikan metode Jaringan Syaraf Tiruan *Backpropagation*. *Backpropagation* umumnya diterapkan pada perceptron berlapis banyak (*multilayer perceptrons*). Dalam prakteknya, banyaknya *hidden layer* paling banyak adalah tiga lapis. Dengan tiga lapis ini hampir semua permasalahan dalam dunia industri telah bisa diselesaikan. Meskipun dalam tahap training lambat, namun metode ini dapat memproses *output* secara cepat. Jaringan syaraf tiruan *backpropagation* telah digunakan untuk memprediksi curah hujan yang terjadi di daerah India oleh (Enireddy Vamisdhar, 2010) dan prediksi jumlah penggunaan listrik oleh (Sowjanya Param, 2015) yang menghasilkan *error rate* yang kecil.

2. TINJAUAN PUSTAKA

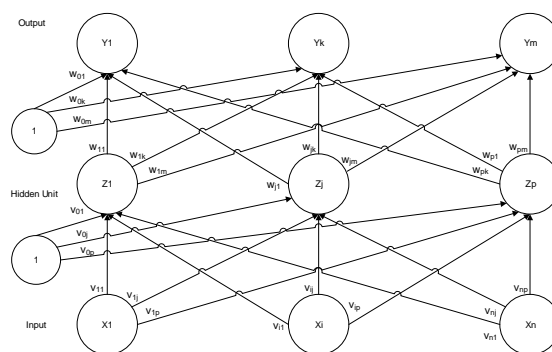
2.1. Koran

Koran atau surat kabar merupakan lembaran tercetak yang memuat laporan yang terjadi di masyarakat dengan ciri-ciri terbit secara periodik, bersifat umum, isinya termasa dan actual mengenai apa saja dan dimana saja di seluruh dunia untuk diketahui pembaca (Onong Uchjana Effendy, 2006).

Sedangkan menurut (Frank Jefkin, 2000) koran merupakan salah satu media yang merupakan rangkuman dari semua isi ebrita yang disajikan melalui media cetak yang termasuk penempatan komposisi layout. Dimana setiap bentuk layout harus mampu berbicara kepada pembaca lewat tampilan unsur rupa seperti: garis, bidang, bentuk warna, tipografi, ilustrasi sebagai bentuk kesatuan secara visual.

2.2. Jaringan Syaraf Tiruan *Backpropagation*

Jaringan saraf tiruan *backpropagation* pertama kali diperkenalkan oleh Rumelhart, Hinton dan William pada tahun 1986, kemudian Rumelhart dan Mc Clelland mengembangkannya pada tahun 1988. Algoritma ini termasuk dalam *supervised learning* dimana ciri dari metode ini adalah meminimalkan *error* pada *output* yang dihasilkan oleh jaringan. Algoritma *backpropagation* untuk neural network umumnya diterapkan pada jaringan berlapis banyak (*multilayer*). Algoritma ini paling tidak mempunyai bagian *input*, bagian *output* dan beberapa lapis yang berada di antara *input* dan *output*. Lapis di tengah ini, yang juga dikenal dengan lapis tersembunyi (*hidden layer*), bisa satu, dua, tiga dst. *Output* lapis terakhir dari *hidden layer* langsung dipakai sebagai *output* dari neural network. Gambar 1 merupakan arsitektur dari jaringan syaraf tiruan *backpropagation*.



Gambar 1. Arsitektur *Backpropagation*

Saat proses pelatihan (*training*), setiap *unit output* membandingkan aktivasinya (y_k) dengan nilai target (t_k) untuk menentukan besarnya *error*. Berdasarkan *error* ini, dihitung faktor σ_j juga dihitung pada *hidden unit* z_j , dimana faktor ini digunakan untuk memperbaiki bobot antara *hidden layer* dan *input layer*. Setelah semua faktor σ ditentukan, bobot untuk semua *layer* diperbarui.

Pada intinya, pelatihan dengan metode *backpropagation* terdiri dari tiga langkah, yaitu:

1. Fase I: propagasi maju (*feedforward*)

Saat propagasi maju setiap *unit input* (x_i) akan menerima sinyal *input* dan akan menyebarkan sinyal tersebut pada tiap *hidden unit* (z_j). Setiap *hidden unit* kemudian akan menghitung aktivasinya dan mengirim sinyal ke (z_j) ke tiap *unit output* kemudian setiap *unit*

$output (Y_k y_k)$ juga akan menghitung aktivasinya (y_k) untuk menghasilkan respons terhadap $input$ yang diberikan jaringan.

2. Fase II: popagasi balik (*backpropagation*)

Setiap *unit output* ($y_k, k=1,2,3,\dots,m$) menerima suatu target (*output* yang diharapkan) yang akan dibandingkan dengan *output* yang dihasilkan. Faktor δ_k digunakan untuk menghitung koreksi *error* (Δw_{jk}) yang dipakai untuk memperbarui w_{jk} di *output layer* ke semua *hidden layer* yang terhubung langsung dengan *output layer* (y_k). Dengan cara yang sama, dihitung faktor koreksi *error* $\Delta v_{ij} (\delta_j)$ di setiap *unit* di *hidden layer* sebagai dasar perubahan bobot.

3. Perubahan Bobot

Setiap *unit output* ($y_k, k = 1,2,3, \dots, m$) akan memperbarui bias dan bobotnya dengan setiap *hidden unit*. Begitu juga dengan setiap *hidden unit* akan memperbarui bias dan bobotnya dengan setiap *unit-unit input*.

Ketiga fase tersebut diulang-ulang terus hingga kondisi penghentian dipenuhi. Umumnya kondisi penghentian yang sering dipakai adalah jumlah iterasi atau kesalahan. Iterasi akan dihentikan jika jumlah iterasi yang dilakukan sudah melebihi jumlah maksimum iterasi yang ditetapkan atau jika kesalahan yang terjadi sudah lebih kecil dari batas toleransi yang diijinkan.

Berikut adalah langkah-langkah *backpropagation*

Langkah 0 :

- Inisialisasi semua bobot dengan bilangan acak kecil
- Menetapkan maksimum epoch, target *error* dan *learning rate*
- Inisialisasi, epoch = 0
- Selama epoch < maksimum_epoch dan MSE < target_error, maka akan dikerjakan langkah-langkah berikut

Langkah 1 :

Jika *stopping condition* masih belum terpenuhi, lakukan Langkah 2-9

Langkah 2 :

Untuk setiap data *training*, lakukan langkah 3-8 Fase I : Propagasi maju (*feedforward*)

Langkah 3 :

Setiap *unit input* menerima sinyal *input* dan menyebarkan sinyal tersebut pada seluruh *unit* pada *hidden layer*

Langkah 4 :

Setiap *hidden unit* ($Z_j, j=1,2,3,\dots,p$) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot termasuk biasanya menggunakan rumus berikut

$$z_{net_j} = v_{j0} + \sum_{i=1}^n x_i v_{ji} \tag{1}$$

Dan memakai fungsi aktivasi *sigmoid* yang telah ditentukan untuk menghitung sinyal *output* dari *hidden unit* yang bersangkutan,

$$z_j = f(z_{net_j}) = \frac{2}{1+e^{-z_{net_j}}} - 1 \tag{2}$$

Lalu mengirim sinyal *output* ke seluruh *unit* pada *unit output*

Keterangan :

- z_{net_j} = sinyal *input* pada *hidden layer* ke - j
- v_{j0} = bias ke *hidden layer* ke - j
- v_{ji} = bobot antara *unit input layer* ke - i dan *hidden layer* ke - j
- x_i = *unit input layer* ke - i
- z_j = *unit input layer* ke - j
- i = urutan *unit input layer*
- j = urutan *unit hidden layer*
- p = jumlah maksimum *unit* pada *hidden layer*

Langkah 5 :

Setiap *unit output* ($y_k, k=1,2,3,\dots,m$) akan menjumlahkan sinyal-sinyal *input* yang sudah berbobot termasuk biasanya

$$y_{net_k} = w_{k0} + \sum_{j=1}^p z_j w_{kj} \tag{3}$$

Dan memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari *unit output* yang bersangkutan:

$$y_k = f(y_{net_k}) = \frac{1}{1+e^{-y_{net_k}}} \tag{4}$$

Keterangan :

- y_{net_k} = sinyal masukan *output* ke - k
- w_{k0} = bias ke *hidden layer* ke - k
- w_{kj} = *output* ke - k dan *hidden layer* ke - j
- z_j = aktivasi *hidden layer* ke - j

Fase II : Propagasi mundur

Langkah 6 :

Setiap *unit output* ($y_k, k=1,2,3,\dots,m$) menerima suatu target (*output* yang diharapkan) yang akan dibandingkan dengan *output* yang dihasilkan

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

Faktor δ_k digunakan untuk menghitung koreksi

error (Δw_{kj}) yang dipakai untuk memperbarui w_{kj} , dimana :

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

Faktor δ_k ini kemudian dikirim ke *layer* depannya

Keterangan :

δ_k = faktor koreksi *error* bobot w_{jk}

t_k = target *output* ke - k

y_k = aktivasi *output* ke - k

Δw_{kj} = nilai koreksi *error* bobot w_{kj}

z_j = aktivasi *hidden layer* ke - j

Langkah 7 :

Setiap *hidden unit* ($Z_j, 1, 2, 3, \dots, p$) menjumlah *input delta* (yang dikirim pada *layer* pada Langkah 6) yang sudah berbobot.

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

Kemudian hasilnya akan dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi eror δ_j , dimana :

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

Faktor δ_j digunakan untuk menghitung koreksi eror (Δv_{ji}) yang akan dipakai untuk memperbarui v_{ik} , dimana :

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

Keterangan :

δ_{net_j} = jumlah delta bobot *hidden layer* ke - j

δ_k = faktor koreksi *error* bobot w_{kj}

w_{kj} = bobot antara *output* ke - k dan *hidden layer* ke - j

δ_j = faktor koreksi bobot v_{ij}

z_j = aktivasi *hidden layer* ke - j

v_{ji} = nilai koreksi *error* bobot v_{ji}

α = laju percepatan (*learning rate*)

δ_i = faktor koreksi *error* bobot v_{ji}

x_i = unit *input* ke - i

Fase III : Perubahan Bobot

Langkah 8 :

Setiap *unit output* ($y_k, k=1, 2, 3, \dots, m$) akan memperbarui bias dan bobotnya dengan setiap *hidden unit*

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} \quad (10)$$

Begitu juga dengan setiap *hidden unit* akan memperbarui bias dan bobotnya dengan setiap *unit-unit input*.

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} \quad (11)$$

Keterangan :

$w_{kj}(\text{baru})$ = bobot baru dari unit *hidden layer* menuju unit *output layer*

$\Delta w_{kj}(\text{lama})$ = bobot lama dari unit *hidden layer* menuju unit *output layer*

$v_{ji}(\text{baru})$ = bobot baru dari unit *hidden layer* menuju unit *output layer*

$\Delta v_{ji}(\text{lama})$ = bobot lama dari unit *hidden layer* menuju unit *output layer*

Langkah 9 :

Memeriksa *stop condition*

Jika *stop condition* telah terpenuhi, maka pelatihan jaringan dapat dihentikan

2.3. Pemilihan Bobot dan Bias awal

Pemilihan bobot awal sangat mempengaruhi jaringan syaraf tiruan dalam mencapai minimum global terhadap nilai *error* dan kecepatan proses pelatihan menuju tingkat konvergensi. Apabila nilai bobot terlalu besar, maka *input* ke setiap lapisan tersembunyi atau lapisan *output* akan jatuh pada daerah yang memiliki turunan fungsi sigmoid yang sangat kecil. Sebaliknya, jika nilai bobot awal terlalu kecil akan menyebabkan pelatihan berjalan sangat lambat (Kusumadewi, 2004).

Langkah-langkah dari inisialisasi Ngunyen-Widrow adalah sebagai berikut,

1. Inisialisasi semua bobot awal v_{ji} lama dengan bilangan acak dalam interval [-0.5, 0.5]

$$\|v_j\| = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{jn}^2} \quad (12)$$

2. Bobot yang digunakan untuk inisialisasi

$$v_{ji} = \frac{\beta v_{ji}(\text{lama})}{\|v_j\|} \quad (13)$$

3. Bias yang dipakai sebagai inisialisasi adalah v_{jo} yang merupakan bilangan acak antara - β dan β

Keterangan :

β = faktor skala sebesar $\beta =$

$$0.7 \sqrt{p}$$

v_{ji} = bobot dari unit *input* i menuju unit *hidden-j*

v_j = bobot pada *hidden layer*

2.4 Normalisasi

Metode normalisasi yang digunakan adalah min-max normalization, metode ini mengubah

data ke range baru lain yaitu antara 0 sampai 1.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (14)$$

Keterangan :

x' = hasil normalisasi

x = data asli

x_{max} = nilai maksimum dari semua data asli

x_{min} = nilai minimum dari semua data asli

2.5 Momentum

Pada *backpropagation* yang menggunakan momentum, perubahan bobot merupakan kombinasi dari kondisi gradient saat ini dan kondisi gradient sebelumnya. Simbol μ merupakan konstanta momentum dengan nilai $0 \leq \mu \leq 1$.

$$\Delta v_{ji}(t + 1) = (\alpha \delta_j x_i) + (\mu \alpha \delta_j x_i) \quad (15)$$

Dan

$$\Delta w_{kj}(t + 1) = (\alpha \delta_k x_j) + (\mu \alpha \delta_k x_j) \quad (16)$$

2.6 Mean Square Error (MSE)

Mean Squared Error (MSE) adalah rata-rata kesalahan meramal yang dikuadratkan. Jika terdapat sebanyak m *training* data, maka untuk menghitung *Mean Square Error* digunakan persamaan berikut:

$$MSE = \sum \frac{(t - y_k)^2}{N} \quad (16)$$

Keterangan :

t = nilai *output* target

y_k = nilai *output* jaringan

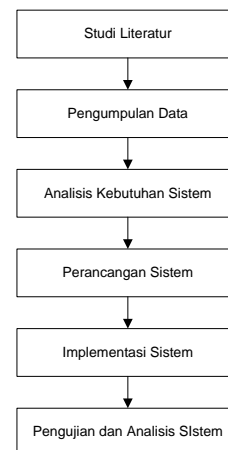
N = banyaknya data

3. METODOLOGI

Tahapan yang dilakukan dalam penelitian ini ditunjukkan pada Gambar 3. berupa diagram alir metodologi penelitian

3.1 Data Penelitian

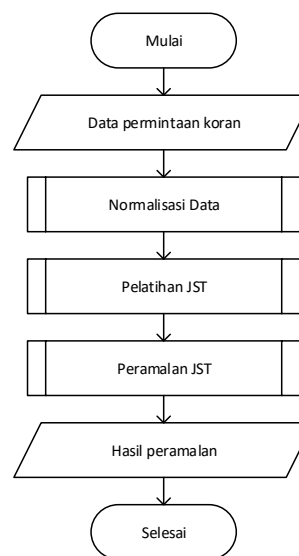
Pengumpulan data yang dilakukan adalah dengan cara observasi dan wawancara dengan narasumber bagian HRD yang ada di Bangkalan, Madura pada tanggal 23 April 2016. Observasi ini dilakukan untuk memperoleh informasi tentang jumlah permintaan koran yang telah ada minimal permintaan selama satu tahun. Data yang diambil digunakan untuk menghitung jumlah permintaan yang optimal per-harinya



Gambar 2. Diagram Alir Metodologi Penelitian

4. PERANCANGAN

Perancangan proses prediksi jumlah permintaan koran menggunakan metode *backpropagation* ditunjukkan pada Gambar 3 sebagai berikut



Gambar 3. Diagram Alir Perancangan

Langkah-langkah proses prediksi data permintaan koran menggunakan metode *backpropagation* adalah sebagai berikut:

1. Mengambil masukan dataset
2. Normalisasi dataset
3. Melakukan pelatihan jaringan syaraf tiruan *backpropagation*
4. Melakukan permalan jaringan syaraf tiruan *backpropagation*

Keluaran yang dihasilkan merupakan peramalan yang akan dihasilkan dari data uji

5. IMPLEMENTASI

Implementasi antarmuka yang digunakan

pada prediksi permintaan koran menggunakan metode *backpropagation* ditunjukkan apda Gambar 4



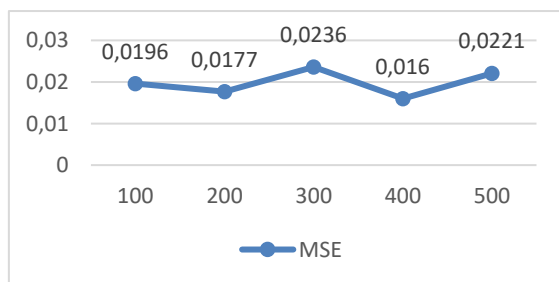
Gambar 4. Halaman Utama Sistem

6. PENGUJIAN

Pengujian yang dilakukan meliputi pengujian iterasi maksimum, pengujian *learning rate*, pengujian jumlah data latih dan data uji.

6.1 Pengujian Iterasi Maksimum

Pengujian iterasi maksimum digunakan untuk mengetahui jumlah iterasi maksimum. Pengujian dilakukan dengan mengubah nilai iterasi mulai 100, 200, 300, 400, dan 500. Nilai *learning rate* yang digunakan adalah 0,5. Tiap uji coba iterasi digunakan bobot awal yang berbeda untuk mencari iterasi maksimum. Pengujian yang dilakukan meliputi pengujian iterasi maksimum, *learning rate*, serta pengujian kombinasi data latih dan uji.

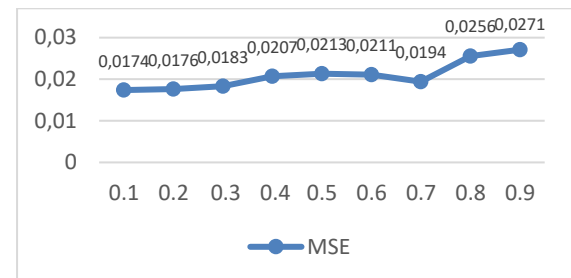


Gambar 5. Grafik Hasil Pengujian Iterasi Maksimum

Pengujian iterasi maksimum digunakan untuk mengetahui jumlah iterasi maksimum. Berdasarkan grafik hasil pengujian jumlah iterasi maksimum terhadap nilai MSE diperoleh iterasi terbaik pada iterasi 400. Dikarenakan semakin kecil tingkat MSE maka semakin baik jumlah iterasinya.

6.2 Pengujian *Learning Rate*

Iterasi maksimum yang digunakan adalah iterasi maksimum pada pengujian pengaruh jumlah iterasi maksimum yaitu 400. Pengujian dilakukan dengan cara mengubah nilai *learning rate* (α) dari 0,1 hingga 0,9. Tiap nilai *learning rate* dilakukan uji coba dengan bobot awal yang berbeda.

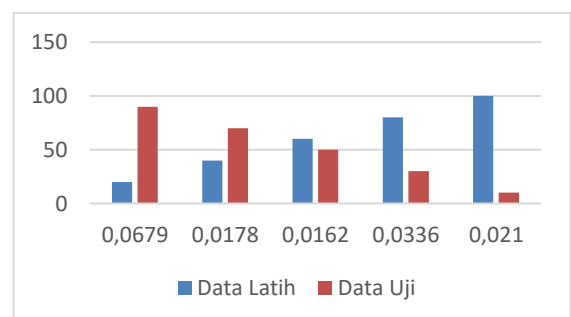


Gambar 6. Grafik Hasil Pengujian Learning Rate

Pengujian Nilai *learning rate* terhadap MSE digunakan untuk mengetahui *learning rate* terbaik yang dapat digunakan. Berdasarkan grafik hasil pengujian nilai *learning rate* terhadap nilai MSE diperoleh nilai terbaik pada *learning rate* 0.1. Dikarenakan semakin kecil tingkat MSE maka semakin baik nilai *learning rate*.

6.3 Pengujian Jumlah data latih dan data uji

Setelah dilakukan uji coba diatas, didapatkan hasil yaitu jumlah iterasi dan *learning rate* terbaik yang akan digunakan sebagai parameter untuk pengujian jumlah data latih. Pengujian dilakukan dengan jumlah data latih 20, 40, 60, 80, 100 hari. Sedangkan jumlah data uji yang digunakan yaitu 90, 70, 50, 30, 10 hari dengan bobot awal yang berbeda.



Gambar 7. Grafik Pengujian Jumlah Data Latih

Berdasarkan Gambar 6.3 nilai MSE terkecil dihasilkan pada jumlah data latih 60 dan jumlah data uji 50 dengan nilai MSE 0.0162.

7. KESIMPULAN

Berdasarkan hasil penelitian tentang prediksi jumlah permintaan koran menggunakan metode jaringan syaraf tiruan backpropagation maka dapat diambil kesimpulan bahwa:

1. Untuk melakukan implementasi metode Backpropagation dalam prediksi jumlah permintaan koran diperlukan data latih yang akan digunakan sebagai sumber pelatihan yang selanjutnya diproses pada tahap pengujian dimana bobot awal harus ditentukan, kemudian menentukan iterasi maksimum, menentukan learning rate dan memperoleh nilai MSE.
2. Berdasarkan hasil pengujian, didapatkan jumlah iterasi terbaik adalah sebesar 200, dan learning rate terbaik adalah sebesar 0.6, dan pengujian data latih dan data uji menghasilkan nilai data latih terbaik yaitu 100 dan data uji 10. Sehingga menghasilkan nilai error rate terkecil yaitu 0.0162.

DAFTAR PUSTAKA

- Effendy O. U., 2006. Ilmu Komunikasi: Teori dan Praktek. [e-book]. Remaja Rosdakarya. Tersedia melalui: Perpustakaan Universitas Negeri Malang
<<http://library.um.ac.id/free-contents/index.php/buku/detail/ilmu-komunikasi-teori-dan-praktek-onong-uchjana-effendy-32610.html>> [Diakses 10 Februari 2016]
- Enireddy V., Varma K.V.S.R.P., Sankara Rao., Satapati R., 2010. *Prediction of Rainfall using Backpropagation Neural Network Model*, IJCSE International journal on computer science and engineering, 2(4),p.1
- Faris G., Alias A., 2012. *Prediction water quality index using backpropagation network algorithm case study: gombak river*, Journal of Engineering Science and Technology, 7(4), p.14
- Hadihardaja, I. K. & Sutikno S., 2012. Pemodelan Curah Hujan-Limpasan menggunakan *Artificial Neural Network (ANN)* dengan metode Backpropagation. Jurnal Teknik Sipil, Volume 12, pp. 249-258
- Heaton, J., 2008. *Introduction to Neural Networks for Java*. 2 ed. Chesterfield: Heaton Research, Inc.
- Heizer J., 2009. Principles of Operation Management. [online] Tersedia di : <<http://library.binus.ac.id/eColls/eThesisdoc/Bab2/2013-1-00192-MN%20Bab2001.pdf>> [Diakses 10 Februari 2016]
- Sownjaya, P. 2015. *Electricity demand prediction using artificial neural network framework*. North Dakota State University [online] Tersedia di : <<http://library.ndsu.edu/tools/dspace/load/?file=/repository/bitstream/handle/10365/25233/Electricity%20Demand%20Prediction%20Using%20Artificial%20Neural%20Network%20Framework.pdf?sequence=1>> [Diakses 10 Februari 2016]
- Samira C., Abdelouhab Z., Jilali A., 2010. *Identification and Prediction of Internet Traffic Using Artificial Neural Network*, *Journal Intelligent Learning Systems & Applications*, 2(2), p.7
- Widjanta B. & Widyaningsih A., 2007. Mengasah Kemampuan Ekonomi. Bandung: CV. Citra Praya