PERBANDINGAN MODEL JARINGAN SYARAF TIRUAN DENGAN ALGORITMA *LEVENBERG-MARQUADT* DAN *POWELL-BEALE CONJUGATE GRADIENT*PADA KECEPATAN ANGIN RATA-RATA DI KOTA SEMARANG

Dwi Ispriyanti[1] , Alan Prahutama[2], Tarno [3] , Budi Warsisto [4] , Hasbi Yasin[5] , Pandu Anggara[6]

[1,2,3,4,5]Staff Pengajar Departemen Statistika FSM Universitas Diponegoro

[6]Mahasiswa Departemen Statistika FSM Universitas Diponegoro

Email: dwiispriyanti@yahoo.com

ABSTRACT

Wind is one of the most important weather components. Wind is defined as the dynamics of horizontal air mass displacement measured in two parameters, namely speed and direction. Wind speed and direction depend on the air pressure conditions around the place. High wind speed intensity can cause high sea water waves. To estimate wind speed intensity required a study of wind speed prediction. One of method that can be used is Artificial Neural Network (ANN). In ANN there are several models, one of which is backpropagation. Thepurpose of this researchis to compare between backpropagation model with Levenberg-Marquadt and Powell-Beale Conjugate Gradient algorithms. The results of this researchshowing that Powell-Beale Conjugate Gradient better than Levenberg-Marquadtalgorithms. The best model architecture obtained is a network with two input layer neurons, six hidden layer neurons, and one output layer neuron. The activation function used are the logistic sigmoid in the hidden layer and linear in the output layer. MAPE value based on the chosen model is 0,0136% in training process and 0,0088% in testing process.

Keywords: Wind, Artificial Neural Network (ANN), Backpropagation,Levenberg-Marquadt, Powell-Beale Conjugate Gradient, Neuron, MAPE

PENDAHULUAN

Angin merupakan salah satu komponen cuaca dalam kehidupan sehari-hari. Menurut [6], angin diartikan sebagai dinamika perpindahan massa udara secara mendatar (horizontal) yang diukur dalam dua parameter, yaitu kecepatan dan arah.Kecepatan dan arah angin bergantung pada kondisi tekanan udara di sekitar tempat tersebut.

Menurut [4], kecepatan angin yang melebihi 40 km/jam dapat menyebabkan bencana, seperti nelayan yang tidak dapat melaut karena gelombang air laut yang tinggi.Besar kecilnya kecepatan angin di kota Semarang menjadi hal yang perlu dipertimbangkan mengingat kota Semarang berada di daerah pesisir Laut Jawa. Oleh karena itu, perlu dilakukan suatu penelitian mengenai kecepatan angin. Penelitian mengenai kecepatan angin dapat menggunakan *Artificial Neural Network* (ANN).

Artificial Neural Network (ANN) merupakan sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf pada makhluk hidup [1]. Metode *Neural Network* (NN) memiliki beberapa model, salah satunya adalah model *backpropagation*. Model *backpropagation* bertujuan untuk melatih suatu jaringan sehingga diperoleh bobot-bobot optimal yang meminimumkan *error*. Istilah

backpropagation berhubungan dengan metode untuk perhitungan gradien fungsi *error* yang berkaitan dengan bobot-bobot untuk suatu jaringan *feedforward*. Pelatihan menggunakan *backpropagation* meliputi tiga tahap, yaitu umpan maju (*feedforward*), perhitungan *error*, dan penyesuaian bobot [7].

Pada penelitian ini, proses pelatihan untuk menentukan bobot-bobot koneksi pada jaringan pelatihan*backpropagation* digunakan algoritma *Levenberg-Marquadt* dan *Powell-Beale Conjugate Gradient* dengan bantuan program Matlab R2009a.

METODE PENELITIAN

Sumber Data dan Variabel Penelitian

Data yang digunakan dalam penelitian ini adalah data sekunder yang bersumber dari Badan Meteorologi Klimatologi dan Geofisika Semarang. Data tersebut merupakan data kecepatan angin rata-rata diKota Semarang dari periode 1 Januari 2016 sampai dengan 31 Agustus 2017 sejumlah 609 data.

Variabel yang digunakan dalam penelitian ini adalah data kecepatan angin rata-rata diKota Semarang dari periode 1 Januari 2016 sampai dengan 31 Agustus 2017 sejumlah 609 data. Data tersebut dibagi menjadi dua,yaitu data *training* dan *testing*. Data tersebut diidentifikasi menggunakan plot *Partial Autocorrelation Function* (PACF) sebagai penentuan variabel *input*.

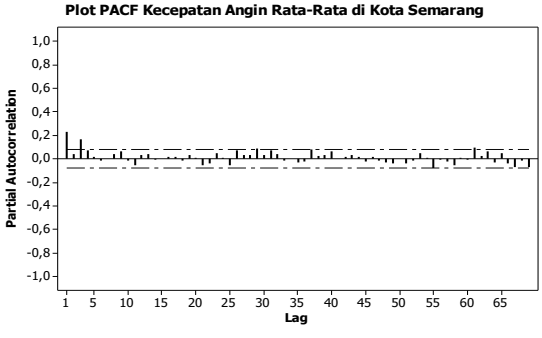
Tahapan Analisis Data

1. Menyiapkan data kecepatan angin rata-rata hariandi Kota Semarang.
2. Menentukan variabel *input* melalui plot PACF.
3. Membagi data menjadi dua, yaitu data *training* dan *testing*.
4. Menentukan jumlah neuron lapisan tersembunyi dari masing-masing algoritma *Levenberg-Marquadt* dan *Powell-Beale Conjugate Gradient*.
5. Membandingkan nilai MAPE *training* dan *testing* dari masing-masing algoritma kemudian dipilih yang terbaik (terkecil) antara algoritma *Levenberg-Marquadt* dan *Powell-Beale Conjugate Gradient*.
6. Menentukan bobot awal.
7. Melakukan inisialisasi parameter (*max epoch*, *learning rate*, *goal*).
8. Melakukan proses *training* dan *testing*.
9. Lakukan secara iteratif dengan pemilihan nilai MAPE yang terkecil dari masing-masing algoritma.
10. Menentukan bobot akhir.
11. Memperoleh model jaringan*backpropagation* terbaik.

HASIL DAN PEMBAHASAN

Penentuan Variabel *Input*

Penentuan variabel *input* dapat dilakukan dengan melihat *lag-lag* yang signifikan pada plot PACF dari data kecepatan angin rata-rata. Jika ada garis yang melewati selang kepercayaan pada plot PACF maka *lag* tersebut dianggap signifikan. Pada penelitian ini, lag yang signifikan adalah lag 1 dan lag 3 sehingga variabel input yang digunakan adalah x_t dipengaruhi oleh x_{t-1} dan x_{t-3}. Hal tersebut dapat ditunjukkan pada Gambar 1.



Gambar 1. Plot PACF Kecepatan Angin Rata-Rata

Pembagian Data

Data yang digunakan untuk pemodelan jaringan *backpropagation* menggunakan algoritma *Levenberg-*

Marquadt dan *Powell-Beale Conjugate Gradient* dibagi menjadi dua, yaitu data *training* dan *testing*. Pembagian data pada penelitian ini menggunakan 70% untuk data *training* dan 30% untuk data *testing*. Oleh karena itu, diperoleh data *training* sebanyak 424 data dan data *testing* sebanyak 182 data.

Penentuan Jumlah Neuron Lapisan Tersembunyi

1. Penentuan dengan Algoritma *Levenberg-Marquadt*

Untuk membangun jaringan *backpropagation* menggunakan algoritma *Levenberg-Marquadt*dapat menggunakan perintah *newff* pada Matlab sebagai berikut:

```
net=newff(minmax(P),[5
1],{'logsig'
'purelin'},'trainlm');
```

Pada perintah tersebut, diulang sebanyak 10 kali dengan cara mencoba neuron lapisan tersembunyi mulai dari 1 sampai dengan 10 neuron. Hasil perhitungannya sebagai berikut:

Tabel 1. Nilai MAPE dari Algoritma *Levenberg-Marquadt*

Neuron	MAPE training	MAPE testing
1	0,0161	0,0503
2	0,0146	0,044
3	0,0125	0,0618
4	0,0123	0,0567
5	**0,012**	**0,0305**
6	0,012	0,0259
7	0,0118	0,03
8	0,0117	0,0456
9	0,0114	0,0354
10	0,0114	0,035

Pada Tabel 1, diperoleh nilai MAPE *training* dan *testing* terkecil sebesar 0,012% dan 0,0305%. Nilai tersebut berada pada 5 neuron pada lapisan tersembunyi (*hidden layer*) dan 1 lapisan *output*. Fungsi aktivasi yang digunakan pada lapisan tersembunyi adalah *logsig* (*sigmoid biner*) dan lapisan output adalah *purelin* (*linear*).

2. Penentuan dengan Algoritma *Powell-Beale Conjugate Gradient*

Untuk membangun jaringan *backpropagation* menggunakan algoritma *Powell-Beale Conjugate Gradient* dapat menggunakan perintah *newff* pada Matlab, yaitu:

```
net=newff(minmax(P),[6
1],{'logsig'
'purelin'},'traincgb');
```

Pada perintah tersebut, diulang sebanyak 10 kali dengan cara mencoba neuron lapisan tersembunyi mulai dari 1 sampai dengan 10 neuron. Hasil perhitungannya sebagai berikut:

Tabel 2. Nilai MAPE dari Algoritma *Powell-Beale Conjugate Gradient*

Neuron	MAPE training	MAPE testing
1	0,0161	0,0503
2	0,0157	0,0477
3	0,0125	0,0572
4	0,013	0,034
5	0,012	0,0766
6	**0,0136**	**0,0088**
7	0,0122	0,0456
8	0,012	0,0431
9	0,0119	0,0443
10	0,0116	0,0266

Pada Tabel 2, diperoleh nilai MAPE *training* dan *testing* terkecil sebesar 0,0136% dan 0,0088%. Nilai tersebut berada pada 6 neuron pada lapisan tersembunyi (*hidden layer*) dan 1 lapisan *output*. Fungsi aktivasi yang digunakan pada lapisan tersembunyi adalah *logsig* (*sigmoid biner*) dan lapisan output adalah *purelin* (*linear*).

Berdasarkan nilai MAPE yang dihasilkan, algoritma *Powell-Beale Conjugate Gradient* menghasilkan nilai yang lebih baik daripada *Levenberg-Marquadt* sehingga dipilih algoritma *Powell-Beale Conjugate Gradient* sebagai dasar pembentukan model jaringan *backpropagarion* dengan 6 neuron lapisan tersembunyi dan 1 lapisan *output* menggunakan fungsi aktivasi *sigmoid biner* dan *linear*.

a. Penentuan Bobot Awal dan Inisialisasi Parameter

Tahap awal dilakukan penentuan bobot awal dengan menggunakan *software* Matlab. Dalam menentukan bobot awal akan menghasilkan bobot dari lapisan *input*dan bias ke lapisan tersembunyi serta bobot dari lapisan tersembunyi dan bias ke lapisan *output* dengan perintah sebagai berikut:

```
Bobot_Awal_Input_Hidden=net.IW{1,1}
Bobot_Awal_Bias_Hidden=net.b{1,1}
Bobot_Awal_Hidden_Output=net.LW{2,1}
Bobot_Awal_Bias_Output=net.b{2,1}
```

Penentuan parameter pembelajaran yang digunakan dalam pelatihan jaringan dengan algortima*Powell-Beale Conjugate Gradient backpropagation* dengan `net.trainParam` untuk memperbaiki bobot dengan perintah sebagai berikut:

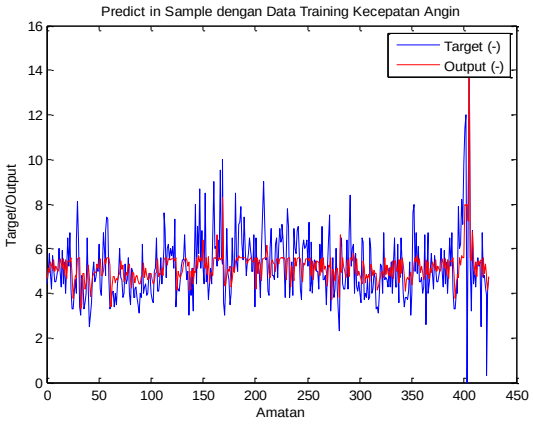
```
net.trainParam.epochs=1000;
net.trainParam.goal=1e-3;
net.trainParam.lr=0.1;
```

Jaringan yang telah terbentuk dapat dilatih dengan perintah sebagai berikut:

```
net=train(net,P,T);
```

b. Proses *Training*

Proses *training* pada pelatihan jaringan dilakukan dengan algoritma *Powell-Beale Conjugate Gradient backpropagation*. Proses tersebut menghasilkan nilai MAPE sebesar 0,0136%, artinya model jaringan yang terbentuk sangat baik.



Gambar 2. Prediksi Data Target dengan *Output* Proses *Training*

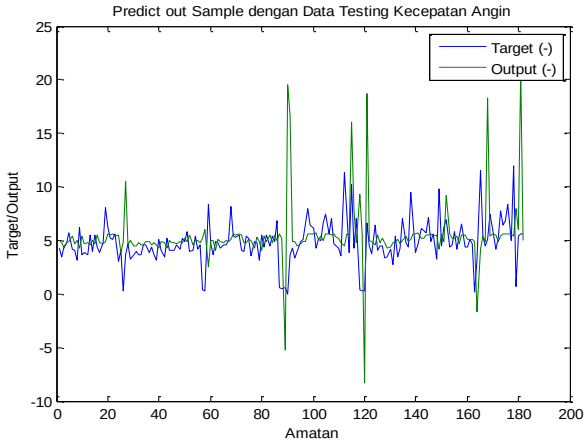
Berdasarkan Gambar 2, dapat dilihat bahwa pelatihan jaringan telah memberikan hasil prediksi yang cukup akurat yang ditunjukkan dengan kedekatan pola *output* yang disimbolkan garis merah dengan data target yang disimbolkan garis biru.

c. Penentuan Bobot Akhir

Setelah proses *training* berhenti maka akan diperoleh bobot akhir. Penentuan bobot akhir yang diperoleh dari pelatihan jaringan dilakukan dengan perintah sebagai berikut:

```
Bobot_Akhir_Input_Hidden=net.IW{1,1}
Bobot_Akhir_Bias_Hidden=net.b{1,1}
Bobot_Akhir_Hidden_Output=net.LW{2,1}
Bobot_Akhir_Bias_Output=net.b{2,1}
```

d. Proses *Testing*



Gambar 3. Prediksi Data Target dengan *Output* Proses *Testing*

Berdasarkan Gambar 2, dapat dilihat bahwa proses *testing* telah memberikan hasil prediksi yang cukup akurat yang ditunjukkan dengan kedekatan pola *output* yang disimbolkan garis hijau dengan data target yang disimbolkan garis biru dengan nilai MAPE sebesar 0,0088%.

e. Penentuan Model Jaringan Terbaik

Arsitektur model terbaik menggunakan jaringan *backpropagation* dengan algoritma *Powell-Beale Conjugate Gradient* yang dibangun dari

6 neuron pada lapisan tersembunyi, 2 lapisan *input* dari x_{t-1} dan x_{t-3},serta 1 lapisan *output*. Fungsi aktivasi yang digunakan pada lapisan tersembunyi adalah *sigmoid biner* dan lapisan output adalah *linear*, dengan model yang diperoleh sebagai berikut:

$$y_k = w_{k0} + \sum_{j=1}^{6} w_{kj}\left(\frac{1}{1+e^{-(v_{j0}+\sum_{i=1} x_i v_{ji})}}\right)$$

dengan

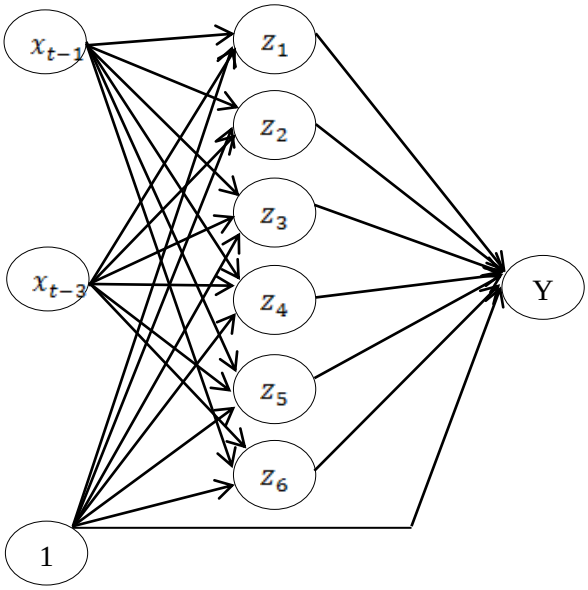
w_{k0} : bobot bias pada lapisan tersembunyi ke lapisan *output*

w_{kj} : bobot lapisan tersembunyi ke-*j* ke lapisan *output*

v_{j0} : bobot bias pada lapisan *input* ke lapisan tersembunyi ke-*j*

v_{ji} : bobot bias lapisan *input* ke lapisan tersembunyi ke-*j*

x_i : variabel *input* ke-*i*



Gambar 4. Arsitektur Model Jaringan Syaraf Tiruan

Prediksi kecepatan angin di kota Semarang dengan model *backpropagation*menggunakan bobot akhir yang diperoleh dari model terbaik. Model terbaik dibangun dari 6 neuron pada lapisan tersembunyi dengan 2 lapisan *input* dari x_{t-1} dan x_{t-3}, serta 1 lapisan *output*. Sebagai contoh, perhitungan luaran secara manual dilakukan pada data *testing* periode 1Maret 2017. Data *input* yang digunakan adalah x_{t-1} = 4,3 dan x_{t-3} = 4,2. Berikut adalah perhitungan prediksi pada data *testing* periode 1 Juli 2017 dengan x_{t-1} = 4,3 dan x_{t-3} = 4,2.

Operasi luaran lapisan *input* ke-*j* ke lapisan tersembunyi sebagai berikut:

$$z\_in_j = v_{j0} + \sum_{i=1} x_i v_{ji}$$

Untuk x_{t-1} = 4,3

$z\_in_1 = -9{,}8141 + 4{,}3 \times 0{,}0507 = -9{,}5961$

$z\_in_2 = -11{,}1220 + 4{,}3 \times (-5{,}2630) = -33{,}7529$

$z\_in_3 = -5{,}9500 + 4{,}3 \times (-1{,}8088) = -13{,}7278$

$z\_in_4 = -56{,}1543 + 4{,}3 \times 8{,}3637 = -20{,}1904$

$z\_in_5 = -7{,}3180 + 4{,}3 \times (-0{,}9114) = -11{,}2370$

$z\_in_6 = 2{,}3097 + 4{,}3 \times 4{,}2060 = 20{,}3955$

Untuk x_{t-3} = 0,3

$z\_in_1 = -9{,}8141 + 4{,}2 \times (-0{,}3333) = -11{,}2140$

$z\_in_2 = -11{,}1220 + 4{,}2 \times 0{,}3199 = -9{,}7784$

$z\_in_3 = -5{,}9500 + 4{,}2 \times 0{,}6806 = -3{,}0915$

$z\_in_4 = -56{,}1543 + 4{,}2 \times 19{,}0759 = 23{,}9645$

$z\_in_5 = -7{,}3180 + 4{,}2 \times 1{,}5581 = -0{,}7740$

$z\_in_6 = 2{,}3097 + 4{,}2 \times 5{,}8280 = 26{,}7873$

Dengan menggunakan fungsi aktivasi *sigmoid biner* pada lapisan tersembunyi diperoleh:

$$z_j = f(z\_in_j) = \frac{1}{1+e^{-z\_in_j}}$$

Untuk x_{t-1} = 4,3

$$z_1 = f(-9{,}5961) = \frac{1}{1+e^{9{,}5961}} = 0{,}0001$$

$$z_2 = f(-33{,}7529) = \frac{1}{1+e^{33{,}7529}} = 0{,}0000$$

$$z_3 = f(-13{,}7278) = \frac{1}{1+e^{13{,}7278}} = 0{,}0000$$

$$z_4 = f(-20{,}1904) = \frac{1}{1+e^{20{,}1904}} = 0{,}0000$$

$$z_5 = f(-11{,}2370) = \frac{1}{1+e^{11{,}2370}} = 0{,}0000$$

$$z_6 = f(20{,}3955) = \frac{1}{1+e^{-20{,}3955}} = 1{,}0000$$

Untuk x_{t-3} = 4,2

$$z_1 = f(-11{,}2140) = \frac{1}{1+e^{11{,}2140}} = 0{,}0000$$
$$z_2 = f(-9{,}7784) = \frac{1}{1+e^{9{,}7784}} = 0{,}0001$$
$$z_3 = f(-3{,}0915) = \frac{1}{1+e^{3{,}0915}} = 0{,}0435$$
$$z_4 = f(23{,}9645) = \frac{1}{1+e^{-23{,}9645}} = 1{,}0000$$
$$z_5 = f(-0{,}7740) = \frac{1}{1+e^{0{,}7740}} = 0{,}3156$$
$$z_6 = f(26{,}7873) = \frac{1}{1+e^{-26{,}7873}} = 1{,}0000$$

Dilanjutkan menggunakan operasi luaran lapisan tersembunyi ke-j ke lapisan *output*:

$$y_k = w_{k0} + \sum_{j=1}^{6} w_{kj}\left(\frac{1}{1+e^{-(v_{j0}+\sum_{i=1} x_i v_{ji})}}\right)$$
$$y_1 = -0{,}1281 + (0{,}6195 \times (0{,}0001 + 0) + 12{,}4994 \times (0 + 0{,}0001) + (-33{,}8019) \times (0 + 0{,}0435) + 1{,}5931 \times (0 + 1) + 12{,}4661 \times (0 + 0{,}3156) + 0{,}4151 \times (1 + 1)$$
$$y_1 = 4{,}7616$$

Dari perhitungan di atas dapat disimpulkan bahwa hasil prediksi kecepatan angin di kota Semarang menggunakan model *backpropagation* dengan algoritma *Powell-Beale Conjugate Gradient* yang dibangun dari 6 neuron pada lapisan tersembunyi, 2 lapisan *input* dari x_{t-1} dan x_{t-3}, serta 1 lapisan *output*serta fungsi aktivasi *sigmoid biner* dan *linear* menghasilkan nilai sebesar 4,7616.

KESIMPULAN

Berdasarkan analisis yang telah dilakukan, diperoleh kesimpulan sebagai berikut:

1. Pelatihan jaringan syaraf tiruan menggunakan *backpropagation* dengan algoritma *Levenberg-Marquadt*menghasilkan nilai MAPE *training* dan *testing* sebesar 0,012% dan 0,0305% sedangkan algoritma *Powell-Beale Conjugate Gradient* menghasilkan nilai MAPE *training* dan *testing* sebesar 0,0136% dan 0,0088%. Oleh karena itu, algoritma *Powell-Beale Conjugate Gradient* lebih baik daripada *Levenberg-Marquadt*.
2. Arsitektur model terbaik dibentuk menggunakan algoritma *Powell-Beale Conjugate Gradient* yang dibangun dari 6 neuron pada lapisan tersembunyi, 2 lapisan input dari x_{t-1} dan x_{t-3}, serta 1 lapisan *output* dengan fungsi aktivasi pada lapisan tersembunyi adalah *logistic sigmoid* dan lapisan *output* adalah *linear* sebagai berikut:

$$y_k = w_{k0} + \sum_{j=1}^{6} w_{kj}\left(\frac{1}{1+e^{-(v_{j0}+\sum_{i=1} x_i v_{ji})}}\right)$$

DAFTAR PUSTAKA

[1] Fausset, L. 1994. *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*. New Jersey: Prentice-Hall Inc.

[2] Haykin, S. 1999. *Neural Networks a Comprehensive Foundation*. New Jersey: Prentice-Hall Inc.

[3] Kusumadewi, S. 2004. *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & Excel Link*. Yogyakarta: Graha Ilmu.

[4] Nurvitasari, Y dan Irhamah. 2012. Pendekatan Fungsi Transfer Sebagai Input Adaptive Neuro-Fuzzy Inference System (ANFIS) dalam Peramalan Kecepatan Angin Rata-Rata Harian di Sumenep. *Jurnal Sains dan Seni* Institut Teknologi Sepuluh November Vol. 1 No. 1

[5] Siang, J.J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Penerbit Andi.

[6] Turyanti, A dan Effendy, S. 2006. *Meteorologi*. Bogor: Institut Pertanian Bogor

[7] Warsito, B. 2009. *Kapita Selekta Statistika Neural Neural Network. Semarang*:BP UNDIP.