

## PERKIRAAN PENJUALAN BEBAN LISTRIK MENGGUNAKAN JARINGAN SYARAF TIRUAN RESILIENT BACKPROPAGATION (RPROP)

Apriliyah, Wayan Firdaus Mahmudy, Agus Wahyu Widodo

Program Studi Ilmu Komputer  
Fakultas MIPA Universitas Brawijaya, Malang

### ABSTRAK

Perkiraan atau prediksi dilakukan untuk memperkirakan perilaku data berdasarkan analisis dan pengolahan data historis (data *time series*). Banyak metode perkiraan yang telah dikembangkan untuk mendapatkan hasil perkiraan yang optimal. Jaringan syaraf tiruan (JST) adalah salah satu metode yang saat ini dikembangkan untuk mendapatkan hasil perkiraan yang mendekati dengan data sebenarnya. Dalam makalah ini dipaparkan implementasi jaringan syaraf tiruan *resilient backpropagation* (Rprop) untuk memprediksi penjualan beban listrik. Data yang digunakan dalam perkiraan adalah jumlah pelanggan, biaya beban, biaya pemakaian dan biaya kelebihan pemakaian dari unit Blimbing, Dinoyo, Kota dan Kebon Agung periode Januari 2003 sampai dengan Desember 2007. Hasil prediksi atau *persentase rata-rata error* antara data prediksi dengan data sebenarnya adalah 0.297% untuk unit Blimbing, 1.743% untuk unit Dinoyo, 0.597% untuk unit Kota dan 0.388% untuk unit Kebon Agung.

Kata kunci : jaringan syaraf tiruan, *resilient backpropagation*, prediksi

## SALE ELECTRICS LOAD FORECASTING USING ARTIFICIAL NEURAL NETWORK RESILIENT BACKPROPAGATION

### ABSTRACT

Forecasting is conducted to estimate data behavior based on historical data analysis and processing (time series data). Many estimation methods were developed to optimize the result. Artificial Neural Network (ANN) is one of the methods which are developed to get result of estimate near to the actual data. The research implemented artificial neural network Resilient Back propagation to predict sale of electric burden. Used data for estimating are amount of consumer, burden expenses, and usage expenses, excess of usage expenses from Unit of Blimbing, Dinoyo, Kota and Kebon Agung in January 2003 – December 2007. Training and testing ANN is conducted with modifying the number of neurons in the hidden layer and update value. Mean Square Error (MSE) for each unit is 0.00002832 for unit of Blimbing, 0.000197 for unit of Dinoyo, 0.00001836 for unit of Kota and 0.00000875 for unit of Kebon Agung. Difference of MSE because of used data for input is differing and weight ANN is obtained at random. Prediction result or percentage of error rates between predicted data and actual data is 0.297% for unit of Blimbing, 1.743% for unit of Dinoyo, 0.597% for unit of Kota and 0.388% for unit of Kebon Agung.

Keywords: Artificial Neural Network, *resilient back propagation*, Forecasting

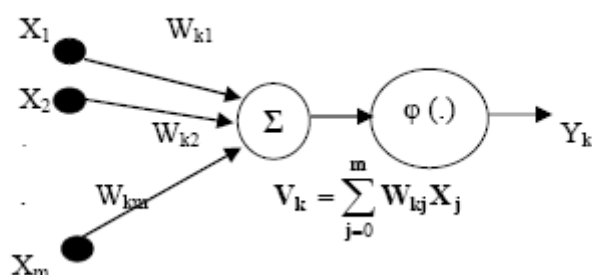
### 1. PENDAHULUAN

Perusahaan Listrik Negara (PLN) melakukan pengembangan pengoperasian daya listrik dan melakukan perencanaan dalam meningkatkan pelayanan kepada para pelanggan listrik. Perkiraan atau prediksi penjualan beban listrik dapat membantu PLN membuat keputusan dan membuat rencana pengembangan pengoperasian daya listrik. Jaringan Syaraf Tiruan (JST) adalah salah satu teknik yang digunakan untuk melakukan perkiraan. Dalam makalah ini akan dibahas salah satu algoritma yang digunakan untuk melakukan prediksi (*forecasting*) penjualan beban listrik yaitu algoritma *resilient backpropagation* (Rprop).

### 1.1 Konsep Dasar Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) adalah suatu model yang mencoba meniru struktur dan cara kerja jaringan syaraf pada otak manusia. Otak manusia berisi berjuta-juta sel syaraf (*neurons*) yang bertugas untuk memproses informasi. Masing-masing *neuron* terhubung dengan *neuron* lain membentuk hubungan yang disebut *synapsis* (hubungan). Struktur JST terdiri dari *neuron*, bobot dan fungsi aktivasi. *Neuron* merupakan bagian penerima informasi dan meneruskan hasil olahan informasi. Kekuatan informasi yang masuk ke *neuron* ditandai dengan adanya bobot. Fungsi aktivasi adalah fungsi yang digunakan untuk menentukan besarnya *output*. Model *neuron* dapat dilihat pada Gambar 1.

JST dirancang agar memiliki kemampuan seperti otak manusia. Kemampuan otak manusia yaitu mampu memproses informasi, mengingat informasi, melakukan perhitungan. Beberapa permasalahan yang sering diselesaikan dengan JST adalah prediksi, klasifikasi, optimasi dan pengenalan pola. Berdasarkan kemampuan yang dimiliki, JST dapat digunakan untuk pembelajaran dan menggunakan hasil pembelajaran untuk menemukan solusi dari suatu permasalahan.



Gambar 1 Model *neuron*

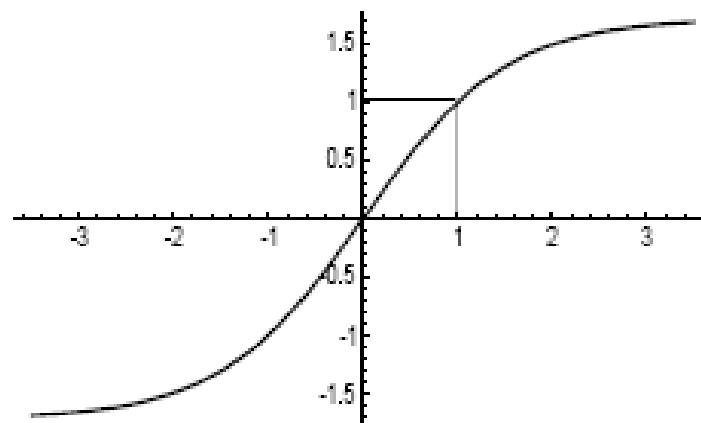
Sejumlah sinyal masukan yaitu  $X_1, X_2, \dots, X_m$  dikalikan dengan masing-masing bobot yang bersesuaian yaitu  $W_{k1}, W_{k2}, \dots, W_{km}$ . Dilakukan penjumlahan dari seluruh hasil perkalian menghasilkan nilai keluaran ( $V_k$ ). Nilai  $V_k$  diaktifkan dengan suatu fungsi aktivasi ( $\phi(\cdot)$ ) untuk menentukan sinyal *output* ( $Y_k$ ).

### 1.2 Fungsi Aktivasi JST

Fungsi aktivasi adalah fungsi yang digunakan untuk menentukan keluaran pada *neuron*. Fungsi aktivasi yang digunakan dalam JST sangat banyak. Fungsi aktivasi yang digunakan untuk mempercepat pembelajaran adalah fungsi *tangen hiperbolik* (Gambar 2). Fungsi ditunjukkan oleh Persamaan 1 dan fungsi turunan ditunjukkan oleh Persamaan 2.

$$f(x) = 1.7159 \tanh\left(\frac{2}{3}x\right) \quad (1)$$

$$f'(x) = 1.1439334 / \cosh^2(0.6666667x) \quad (2)$$



**Gambar 2** Fungsi  $f(x) = 1.7159 \text{ Tanh}(2/3 x)$

### 1.3 Pembelajaran JST

Pembelajaran JST merupakan proses pencarian konfigurasi bobot-bobot dalam tiap layer. Selama proses pembelajaran, terjadi perubahan pada bobot-bobot tiap *synapsis*. Berdasarkan pembelajarannya [4], JST dibagi menjadi:

#### 1. *Supervised*

Pada sistem pembelajaran ini, jaringan diberi masukan tertentu dan keluarannya (*target*) ditentukan. Target akan dibandingkan dengan *output* jaringan. Perbedaan antara target dan *output* disebut *error*. Apabila jaringan menghasilkan *error* yang masih besar, maka perlu dilakukan pembelajaran lagi sampai didapatkan *error* yang cukup kecil. Beberapa metode pembelajaran *supervised* antara lain *single layer perceptron*, *multi layer perceptron* dan *backpropagation*.

#### 2. *Unsupervised*

Sistem pembelajaran *unsupervised* tidak memerlukan target keluaran sehingga jaringan akan *mengatur* sendiri keluarannya. Metode pembelajaran *unsupervised* antara lain metode *kohonen*, *hopfield*, *Radial Basis Function* dan lain sebagainya.

### 1.4 Algoritma Backpropagation

Algoritma *Backpropagation* merupakan algoritma pembelajaran yang terawasi dengan banyak lapisan (*multilayer*). Pada pembelajaran terawasi, terdapat target yang akan dibandingkan dengan keluaran jaringan. Ketika jaringan diberi sinyal masukan, sinyal ini akan menuju ke unit-unit pada lapisan tersembunyi kemudian diteruskan ke unit-unit pada lapisan keluaran. Apabila keluaran jaringan tidak sama dengan target, akan dilakukan langkah mundur pada lapisan tersembunyi diteruskan ke lapisan masukan (*input*).

Pelatihan sebuah jaringan *backpropagation* terdiri dari tiga langkah yaitu pelatihan pola, *input* secara *feedforward*, *backpropagation* dari kumpulan kesalahan dan penyesuaian bobot (Kristanto, 2004). Pelatihan dilakukan berulang-ulang dan berhenti jika telah mencapai batas iterasi maksimum yang ditentukan dan nilai *error* kurang dari *Mean Square Error* (MSE). Ketepatan algoritma *backpropagation* ditentukan dengan *Mean Square Error* (MSE). Semakin kecil nilai MSE maka dapat dianggap bahwa arsitektur jaringan semakin baik, demikian pula sebaliknya. MSE dihitung dengan Persamaan 3.

$$\text{MSE} = \frac{1}{N} \sum_{\mathbf{p}} (t_{\mathbf{p}} - y_{\mathbf{p}})^2 \quad (3)$$

### 1.5 Algoritma Resilient Backpropagation (Rprop)

Algoritma *Rprop* merupakan hasil pengembangan algoritma *backpropagation*. Perubahan bobot pada *backpropagation* dipengaruhi oleh *learning rate* (laju pembelajaran) dan tergantung dari kemiringan kurva *error* ( $\frac{\partial E}{\partial W_{ij}}$ ). Semakin kecil *learning rate* (laju pembelajaran), proses pembelajaran semakin lama. Sedangkan semakin besar *learning rate*, nilai bobot akan jauh dari bobot minimum. Untuk mengatasi hal tersebut, dikembangkan algoritma baru yang disebut Rprop. Algoritma ini menggunakan tanda (positif atau negatif) dari *gradient* untuk menunjukkan arah penyesuaian bobot. Sedangkan ukuran perubahan bobot ditentukan dengan nilai penyesuaian ( $\Delta_0$ ) [7].

Nilai penyesuaian mempunyai batas bawah ( $\Delta_{min}$ ) dan batas atas ( $\Delta_{max}$ ). Beberapa penelitian menggunakan  $\Delta_{min} = 10^{-6}$  dan  $\Delta_{max} = 50$ . Parameter *Rprop* yang lain adalah factor penurun ( $\eta^-$ ) dan faktor penaik ( $\eta^+$ ) atau dapat dikatakan sebagai laju pembelajaran seperti pada *backpropagation*. Nilai laju pembelajaran yang sering digunakan adalah 1.2 untuk nilai  $\eta^+$  dan 0.5 untuk nilai  $\eta^-$ . Aturan nilai penyesuaian[4]:

$$\Delta_{ij}^{(t)} = \begin{cases} \eta^+ \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} \frac{\partial E^{(t-1)}}{\partial w_{ij}} > 0 \\ \eta^- \Delta_{ij}^{(t-1)}, & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} \frac{\partial E^{(t-1)}}{\partial w_{ij}} < 0 \\ \Delta_{ij}, & \text{else} \end{cases} \quad (4)$$

Dimana:

- $\eta^-$  : *learning rate* penurun (faktor penurun)
- $\eta^+$  : *learning rate* penaik (faktor penaik)
- $0 < \eta^- < 1 < \eta^+$

$\frac{\partial E^t}{\partial W_{ij}}$  : kemiringan kurva *error* terhadap bobot (*gradient*) pada iterasi saat ini.

$\frac{\partial E^{t-1}}{\partial W_{ij}}$  : kemiringan kurva *error* terhadap bobot (*gradient*) pada iterasi sebelumnya.

Aturan penyesuaian bobot:

$$w_{ij}(t) = \begin{cases} -\Delta_{ij}^{(t)}, & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} > 0 \\ +\Delta_{ij}^{(t)}, & \text{if } \frac{\partial E^{(t)}}{\partial w_{ij}} < 0 \end{cases} \quad (5)$$

Algoritma Rprop:

1. Inisialisasi penyesuaian awal  $\Delta_{ij}(t) = \Delta_0 = 0$

$$\text{Gradient awal } \frac{\partial E}{\partial W_{ij}}(t-1) = 0$$

2. Lakukan langkah-langkah berikut sampai bobot konvergen

a. Hitung *gradient*  $\frac{\partial E}{\partial W}(t)$

b. Untuk semua bobot, hitung nilai penyesuaian

1. Jika  $\frac{\partial \mathbf{E}}{\partial \mathbf{W}}(\mathbf{t}) * \frac{\partial \mathbf{E}}{\partial \mathbf{W}_{ij}}(\mathbf{t}-1) \geq 0$  maka

$$\Delta_{ij}(\mathbf{t}) = \min(\Delta_{ij}(\mathbf{t}-1) * \eta^+, \Delta_{\max}) \tag{6}$$

2. Jika  $\frac{\partial \mathbf{E}}{\partial \mathbf{W}}(\mathbf{t}) * \frac{\partial \mathbf{E}}{\partial \mathbf{W}_{ij}}(\mathbf{t}-1) < 0$  maka

$$\Delta_{ij}(\mathbf{t}) = \max(\Delta_{ij}(\mathbf{t}-1) * \eta^-, \Delta_{\min}) \tag{7}$$

c. Hitung perubahan bobot

1. Jika  $\frac{\partial \mathbf{E}}{\partial \mathbf{W}}(\mathbf{t}) \geq 0$  maka

$$\mathbf{w}(\mathbf{t}+1) = \mathbf{w}(\mathbf{t}) - \Delta_{ij}(\mathbf{t}) \tag{8}$$

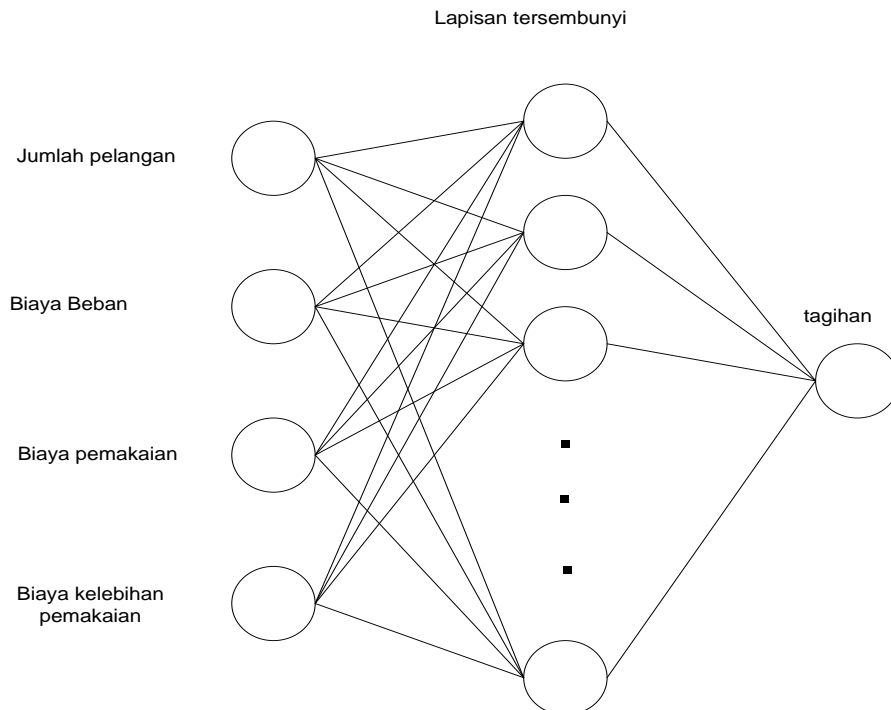
2. Jika  $\frac{\partial \mathbf{E}}{\partial \mathbf{W}}(\mathbf{t}) < 0$  maka

$$\mathbf{w}(\mathbf{t}+1) = \mathbf{w}(\mathbf{t}) + \Delta_{ij}(\mathbf{t}) \tag{9}$$

3. selesai

### 1.6 Perkiraan Penjualan Beban Listrik Menggunakan JST Rprop

Perkiraan penjualan beban listrik dengan menggunakan metode JST memerlukan *input* berupa data jumlah pelanggan, biaya beban, biaya pemakaian dan biaya kelebihan pemakaian. Sebagai *output* (keluaran) adalah tagihan listrik untuk bulan berikutnya. Arsitektur jaringan syaraf yang digunakan adalah 4 unit *neuron* pada lapisan *input*, 1 unit *neuron* pada lapisan *output*. Jumlah unit *neuron* pada lapisan *hidden* sebanyak n unit, ditentukan dengan melakukan modifikasi untuk mendapatkan arsitektur jaringan yang sesuai untuk melakukan perkiraan. Arsitektur JST perkiraan penjualan beban listrik dapat dilihat pada Gambar 3.



**Gambar 3.** Arsitektur JST perkiraan penjualan beban listrik

## 2. METODE

Tahapan dalam perkiraan penjualan beban listrik adalah:

### 1. Menentukan sumber data

Dalam penelitian ini data diambil dari PT PLN (PERSERO) Distribusi Jawa Timur APJ Malang. Data yang diperoleh adalah data yang berasal dari laporan penjualan tenaga listrik oleh Unit Pelayanan (UP) Blimbing, UP Dinoyo, UP Kota dan UP Kebon Agung. Data terdiri dari jumlah pelanggan, biaya beban, biaya pemakaian, biaya kelebihan pemakaian kVARh dan tagihan. Data dari masing-masing UP dalam skala bulan dimulai dari bulan Januari 2003 sampai dengan bulan Desember 2007.

### 2. Penentuan pola pelatihan

Pola pelatihannya adalah jumlah pelanggan ( $x_1$ ), biaya beban ( $x_2$ ), biaya pemakaian ( $x_3$ ) dan biaya kelebihan pemakaian kVARh ( $x_4$ ) pada tiap UP dengan targetnya ( $t$ ) adalah tagihan listrik.

### 3. Pelatihan JST Rprop

Data yang dipakai pada pelatihan ini sebesar 90 % dari data yang digunakan dalam penelitian ini yaitu data dari Januari 2003 sampai dengan Desember 2006. Jaringan terdiri dari satu lapisan *input* dengan empat *neuron*, satu lapisan *hidden* (tujuh sampai sepuluh *neuron*) dan satu lapisan *output* dengan satu *neuron*. Data dilatihkan ke jaringan hingga *error* minimum, jika kesalahan mencapai nilai minimum maka bobot akan disimpan dalam data penyimpanan (dataset).

### 4. Pengujian

Data yang dipakai untuk pengujian adalah 10% dari seluruh data yaitu data tahun 2007 (Januari sampai Desember). Perkiraan dilakukan pada masing-masing UP yaitu UP Blimbing, UP Dinoyo, UP Kota dan UP Kebon Agung.

### 5. Evaluasi

Pada langkah ini untuk mengevaluasi berapa nilai penyesuaian yang terbaik sehingga didapatkan kesalahan minimum. Evaluasi dilakukan dengan mencoba beberapa nilai penyesuaian yaitu mulai 0.01 sampai 0.9. Serta memodifikasi jumlah *neuron* pada lapisan *hidden* yaitu tujuh sampai sepuluh. Sedangkan nilai faktor penaik ( $\eta^+$ ) dan faktor penurun ( $\eta^-$ ) tetap yaitu masing-masing diset 1.2 dan 0.5. Uji coba dilakukan dengan mengubah jumlah *neuron* pada lapisan tersembunyi dan menggunakan nilai penyesuaian yang berbeda-beda. Uji coba dilakukan pada tiap-tiap unit sebanyak lima kali uji coba dan dari lima kali percobaan diambil nilai rata-ratanya.

## 3. HASIL DAN PEMBAHASAN

Keberhasilan JST Rprop diukur dengan *Mean Square Error*(MSE). Semakin kecil nilai MSE, maka kinerja JST semakin bagus. Tabel 1 adalah tabel MSE terkecil tiap unit setelah dilakukan uji coba sebanyak lima kali.

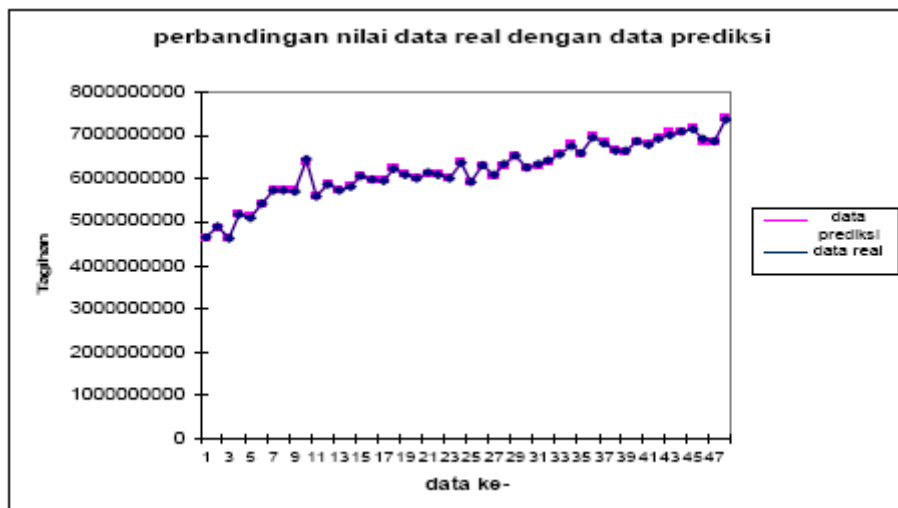
**Tabel 1.** MSE Terkecil

Unit	MSE pelatihan	MSE pengujian	Banyaknya <i>hidden neuron</i>	Nilai Penyesuaian
Blimbing	0,00001432	0,00057371	9	0,09
Dinoyo	0,00019442	0,0014768	9	0,09
Kota	0,00001793	0,00016896	8	0,03
Kebon Agung	0,00000947	0,00062405	9	0,01

Dari Tabel 1, didapatkan arsitektur JST Rprop. Untuk unit Blimbing, Dinoyo dan Kebon Agung adalah 4-9-1. Hal ini berarti terdapat 4 *neuron* pada *input layer*, 9 *neuron* pada *hidden layer* dan 1 *neuron* pada *output layer*. Sedangkan untuk unit Kota adalah 4-8-1. Nilai penyesuaian

yang digunakan juga berbeda-beda dan nilai ini diperoleh dengan melakukan uji coba terhadap beberapa nilai penyesuaian yaitu 0.01, 0.03, 0.05, 0.07, 0.09, 0.1. Berdasarkan penelitian, nilai ini sudah dapat mewakili untuk melakukan pelatihan.

Setelah didapatkan arsitektur, JST dapat digunakan untuk melakukan perkiraan. Berdasarkan Gambar 4 yaitu perbandingan antara nilai data sebenarnya dengan data keluaran JST, dapat dilihat bahwa data hasil prediksi dapat mengikuti pola meskipun masih terdapat selisih (*error*). Nilai *persentase error*, dapat dilihat pada Tabel 2. Dari Tabel 2 dapat dilihat bahwa *persentase error* yang dihasilkan oleh tiap-tiap unit sangat kecil. Hal ini menunjukkan bahwa JST Rprop telah dapat digunakan untuk melakukan perkiraan.



**Gambar 4.** Perbandingan data real dengan data hasil keluaran JST

**Tabel 2.** Persentase *error* tiap unit

Unit	% error
Blimbing	0,297
Dinoyo	1,743
Kota	0,597
Kebon Agung	0,399

#### 4. KESIMPULAN

1. Dari hasil uji coba 48 pola data pelatihan dengan jumlah *neuron* pada lapisan tersembunyi yang bervariasi, didapatkan arsitektur JST Rprop yang berbeda-beda. Dari Tabel 1, didapatkan arsitektur JST Rprop untuk unit Blimbing dan unit Dinoyo yaitu 4-9-1 (4 *neuron* lapisan *input*, 9 *neuron* lapisan tersembunyi dan 1 *neuron* lapisan *output*), unit Kota adalah 4-8-1 sedangkan untuk unit Kebon Agung adalah 4-7-1. Perbedaan arsitektur JST pada tiap-tiap unit karena bobot awal pelatihan diperoleh secara acak.
2. Setelah dilakukan pelatihan dengan 5000 iterasi (perulangan), didapatkan nilai MSE yang berbeda-beda. Nilai MSE yang berbeda-beda karena setiap kali pelatihan, bobot awal diperoleh secara acak. Dengan mengujicobakan beberapa nilai penyesuaian (0.01, 0.03, 0.05, 0.07, 0.09 dan 0.1), menghasilkan nilai MSE yang berbeda-beda dan dari tiap-tiap unit diambil nilai MSE terkecil sehingga didapatkan nilai penyesuaian untuk unit Blimbing dan unit Dinoyo adalah 0.09, untuk unit Kota adalah 0.03 dan 0.01 untuk unit Kebon Agung.

3. Hasil perkiraan (prediksi) penjualan beban listrik dinyatakan dengan *persentase* rata-rata *error* atau selisih antara data sebenarnya dan data prediksi. *Persentase* rata-rata *error* 0.297% untuk unit Blimbing, 1.743% untuk unit Dinoyo, 0.597% untuk unit Kota dan 0.388% untuk unit Kebon Agung. Hal ini menunjukkan bahwa JST *Rprop* mampu melakukan perkiraan mendekati data yang sebenarnya.

#### DAFTAR PUSTAKA

- [1] Akobir, Shahidi. 2007. *Mathematical apparatus of the RProp learning algorithm*. <http://www.basegroup.ru/neural/rprop.en.htm>. Tanggal Akses : 5 November 2007.
- [2] Anonymous. 2004. *Penggolongan Pelangan dan Tarif Harga Jual*. <http://www.indo.net.id/pln/htdocs/pengol.htm>. Tanggal Akses: 23 September 2007.
- [3] Kristanto, Andri. 2004. *Jaringan Syaraf Tiruan (Konsep Dasar, Algoritma dan Aplikasi)*. Gaya Media. Yogyakarta.
- [4] Kusumadewi, Sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Graha Ilmu. Yogyakarta.
- [5] Lecun, Yann dan Klaus Robert Muller. 1998. *Efficient Backprop*. IEEE volume 86 No 11.
- [7] Riedmiller, M. and Braun, H. 1992. *RPROP : A fast adaptive learning algorithm*. In Proceedings of the 1992 International Symposium on Computer and Information Sciences, Antalya, Turkey, pp.279-285.
- [8] Subiyanto. 2005. *Aplikasi Jaringan Syaraf Tiruan sebagai Metode Alternatif Prakiraan Beban Jangka Pendek*. [www.Elektroindonesia.com/elektro/ener29.html](http://www.Elektroindonesia.com/elektro/ener29.html). Tanggal Akses:13 Agustus 2007.