

PREDIKSI BEBAN LISTRIK PADA PT.PLN (PERSERO) MENGUNAKAN REGRESI INTERVAL DENGAN NEURAL FUZZY

^[1]Prita Ayuningtyas, ^[2]Dedi Triyanto, ^[3]Tedy Rismawan
^[1,2,3]Jurusan Sistem Komputer, Fakultas MIPA Universitas Tanjungpura
Jl. Prof. Dr. Haji Hadari Nawawi,, Pontianak
Telp./Fax.: (0561) 577963
Email:
^[1]Ayuning.tyas@student.untan.ac.id, ^[2]dedi.triyanto@siskom.untan.ac.id,
^[3]tedyrismawan@siskom.untan.ac.id

Abstrak

Tenaga listrik merupakan salah satu kebutuhan dasar dalam kehidupan manusia khususnya pada era globalisasi dan modernisasi saat ini. Besarnya konsumsi listrik pada suatu waktu tidak dapat dihitung secara pasti, sehingga diperlukan prediksi beban listrik untuk menyelenggarakan usaha penyediaan daya listrik dalam jumlah merata. Prediksi beban listrik dapat dihitung dengan salah satu metode pada neural fuzzy yaitu regresi interval. Model regresi interval didasarkan pada jaringan backpropagation. Model ini menggunakan dua jaringan backpropagation yang dilatih secara terpisah. Satu jaringan digunakan untuk mencari batas atas interval sedangkan satu jaringan lainnya digunakan untuk mencari batas bawah interval. Aplikasi yang dibangun menerima input berupa tujuh buah data beban listrik kemudian memprosesnya menggunakan regresi interval dengan neural fuzzy yang kemudian memberikan informasi hasilnya berupa interval beban puncak listrik. Setelah melakukan tahap pengujian maka diperoleh hasil kinerja aplikasi mencapai 71,42%.

Kata Kunci : Energi Listrik, Regresi Interval, Neural fuzzy

1. PENDAHULUAN

Listrik adalah salah satu kebutuhan terpenting dalam kehidupan manusia, khususnya pada era globalisasi dan modernisasi saat ini. Listrik sangat diperlukan dalam kehidupan manusia baik untuk keperluan sehari-hari, dunia usaha, industri, pemerintahan, pendidikan, dan lainnya. Besarnya konsumsi listrik pada suatu waktu tidak dapat dihitung secara pasti. Jumlah konsumsi listrik yang tidak tentu dan tanpa diperkirakan terlebih dahulu dapat berpengaruh pada kesiapan dari unit pembangkit untuk menyediakan pasokan listrik kepada konsumen. Ketidak-seimbangan daya listrik antara sisi pembangkit dan sisi konsumen dapat mengakibatkan kerugian pada dua sisi tersebut. Oleh karena itu, prediksi memiliki peranan penting dalam pengoperasian dasar tenaga listrik.

Penelitian terhadap prediksi beban listrik menggunakan kecerdasan buatan telah banyak dilakukan. Diantaranya prediksi beban listrik di pulau Bali menggunakan Jaringan Syaraf Tiruan *Backpropagation* [1] serta peramalan beban listrik jangka pendek pada sistem kelistrikan Jawa Timur dan Bali menggunakan *fuzzy time series* [2]. Penggunaan metode jaringan syaraf tiruan dalam hal

prediksi beban listrik memiliki beberapa kelemahan yaitu hanya menghasilkan satu nilai keluaran saja, sedangkan penggunaan metode *fuzzy inference system* memiliki kelemahan yaitu diperlukannya suatu metode optimasi dalam menentukan parameter yang digunakan.

Sistem ini memprediksi nilai beban puncak listrik harian menggunakan metode regresi interval dengan neural fuzzy yang diharapkan dapat mengurangi kelemahan dari masing-masing metode tersebut bila digunakan secara terpisah sehingga prediksi yang dihasilkan diharapkan dapat lebih akurat. Metode ini merupakan penggabungan dari metode sistem fuzzy dengan jaringan syaraf tiruan. Model tersebut menggunakan 2 jaringan *backpropagation*. Satu jaringan digunakan untuk batas atas interval, sedangkan satu jaringan lainnya digunakan untuk batas bawah interval dari nilai beban puncak listrik harian. Regresi interval dengan *neural fuzzy* merupakan metode yang menggunakan jaringan syaraf *backpropagation* untuk pendekatan fungsi (regresi) dengan koefisien-koefisien yang direpresentasikan dengan bilangan *fuzzy*. Metode ini dapat menghasilkan dua nilai keluaran sehingga hasil prediksi berupa nilai interval yang dapat

memperkirakan antara nilai maksimum dan nilai minimum prediksi beban listrik.

2. LANDASAN TEORI

2.1 Logika Fuzzy

Fuzzy secara bahasa diartikan sebagai kabur atau samar-samar. Suatu nilai dapat bernilai besar atau salah secara bersamaan. Dalam *fuzzy* dikenal derajat keanggotaan yang memiliki rentang nilai 0 (nol) hingga 1(satu). Berbeda dengan himpunan tegas yang memiliki nilai 1 atau 0 (ya atau tidak). Logika *Fuzzy* merupakan suatu logika yang memiliki nilai kekaburan atau kesamaran (*fuzzyness*) antara benar atau salah. Dalam teori logika *fuzzy* suatu nilai bisa bernilai benar atau salah secara bersama. Namun berapa besar keberadaan dan kesalahan suatu tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* memiliki derajat ke-anggotaan dalam rentang 0 hingga 1. Berbeda dengan logika digital yang hanya memiliki dua nilai 1 atau 0. Logika *fuzzy* digunakan untuk menter-jemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*), misalkan besaran kecepatan laju kendaraan yang diekspresikan dengan pelan, agak cepat, cepat, dan sangat cepat. Logika *fuzzy* menunjukkan sejauh mana suatu nilai itu benar dan sejauh mana suatu nilai itu salah. Tidak seperti logika klasik/tegas, suatu nilai hanya mempunyai 2 kemungkinan yaitu merupakan suatu anggota himpunan atau tidak. Derajat keanggotaan 0 (nol) artinya nilai bukan merupakan anggota himpunan dan 1 (satu) berarti nilai tersebut adalah anggota himpunan[3].

2.2 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan adalah suatu representasi buatan dari otak manusia yang selalu mencoba untuk menstimulasikan proses pem-belajaran pada otak manusia tersebut. Istilah buatan disini digunakan karena jaringan syaraf ini di-implementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran [4].

Ada beberapa tipe jaringan syaraf, namun demikian hampir semuanya memiliki komponen-komponen yang sama. Seperti halnya otak manusia, jaringan syaraf juga terdiri dari beberapa neuron, dan ada hubungan antara neuron- neuron tersebut. Neuron-neuron tersebut akan menyampaikan informasi yang diterima melalui sambungan keluarnya menuju ke neuron-neuron yang lain. Pada jaringan syaraf, hubungan ini dikenal dengan nama bobot. Informasi tersebut disimpan pada suatu nilai tertentu pada bobot tersebut. Informasi (disebut dengan:*input*) akan dikirim ke neuron dengan bobot kedatangan tertentu. *Input* ini akan diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang datang. Hasil penjumlahan ini

kemudian akan dibandingkan dengan suatu nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap neuron. Apabila neuron tersebut diaktifkan, maka neuron tersebut akan mengirimkan *output* melalui bobot-bobot *output*nya ke semua neuron yang berhubungan dengannya [4].

Pada jaringan syaraf, neuron-neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (*neuron layers*). Biasanya neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan-lapisan sebelum dan sesudahnya (kecuali lapisan *input* dan lapisan *output*). Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan *input* sampai ke lapisan *output* melalui lapisan yang lainnya, yang sering dikenal dengan nama lapisan tersembunyi (*hidden layer*) [5].

Jaringan syaraf tiruan ditentukan oleh 3 hal:

1. Pola hubungan antar neuron (disebut arsitektur jaringan)
2. Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algorithm*)
3. *Output* menerima *input* dari neuron-neuron dengan bobot hubungan masing-masing. Ketiga impuls neuron yang ada dijumlahkan. Besarnya impuls yang diterima oleh *output* mengikuti fungsi aktivasi. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk merubah bobot.

2.3 Neural Fuzzy Sistem

Neural fuzzy Sistem (NFS) dirancang untuk merealisasikan proses penalaran *fuzzy*, dimana bobot-bobot yang terhubung pada jaringan berhubungan dengan parameter-parameter penalaran *fuzzy*. Dengan menggunakan algoritma pembelajaran *backpropogation*, NFS dapat meng-identifikasi aturan-aturan *fuzzy* dan melatih fungsi keanggotaan dari penalaran *fuzzy* tersebut. Biasanya, NFS memiliki neuron-neuron yang terpisah antara bagian anterseden, bagian operator, dan bagian konsekuen. NFS berupa jaringan dengan banyak lapisan yang digunakan untuk menentukan relsi masukan-keluaran pada sistem *fuzzy*. NFS dapat melatih aturan-aturan yang berbentuk linguistik atau fungsi keanggotaan. Ada 3 kemungkinan [6], yaitu:

1. Sistem diawali tanpa aturan, kemudian dibuat aturan, hingga masalah pembelajaran tersebut terselesaikan. Pembentukan aturan baru ini dipicu oleh ketidakcukupan aturan-aturan yang sudah ada pada basis pengetahuan.
2. Sistem dimulai dengan semua aturan, yang dapat dibuat dalam batas partisi variabel dan menghapus aturan-aturan yang tidak sesuai dalam basis pengetahuan berdasarkan evaluasi terhadap kinerjanya.

3. Sistem dimulai dengan suatu basis pengetahuan dengan sejumlah aturan tertentu, dan selama proses pembelajaran, aturan-aturan tersebut diganti dengan suatu proses estimasi.

Pada NFS, perancangan jaringan syaraf tiruan dipandu oleh formalisme logika *fuzzy*, dimana perancangan jaringan syaraf tiruan tersebut digunakan untuk mengimplementasikan logika *fuzzy* dan *fuzzy decision making*, dan juga digunakan untuk merepresentasikan fungsi keanggotaan yang merepresentasikan himpunan *fuzzy* [7].

2.4 Regresi interval

Model regresi interval merupakan model regresi yang menggunakan *neural fuzzy*. Misalkan diberikan pasangan *input-output* (x_k, d_k) , $k = 1, 2, \dots, p$ dengan $x_k = (x_{k1}, x_{k2}, \dots, x_{kn})$. Suatu model regresi *fuzzy* pada pola ke- k direpresentasikan sebagai [8]:

$$Y(x_k) = A_0 + A_1 x_{k1} + \dots + A_n x_{kn} \quad (1)$$

Dengan A_i adalah bilangan *fuzzy*. Oleh karena itu, nilai estimasi *output* $Y(x_k)$ juga merupakan bilangan *fuzzy*. Analisis regresi *fuzzy* dapat disederhanakan menjadi analisis regresi interval dimana model regresi interval nantinya akan dibentuk.

Konsep dasar dari analisis regresi interval yang didasarkan pada jaringan *backpropagation*, diperkenalkan oleh Ishibuchi dan Tanaka (1992). Model tersebut menggunakan 2 jaringan *backpropagation*. Satu jaringan digunakan untuk batas atas interval, sedangkan satu jaringan lainnya digunakan untuk batas bawah interval data. Kedua jaringan tersebut dilatih secara terpisah [9]. Misalkan $g^+(x_k)$ dan $g^-(x_k)$ adalah *output* dari kedua jaringan *backpropagation* tersebut (BPN^+ dan BPN^-) yang berhubungan dengan *input* vektor x_k , dimana setiap jaringan memiliki n neuron pada lapisan *input* dan 1 neuron pada lapisan *output*. Proses pembelajaran dilakukan terhadap kedua jaringan (BPN^+ dan BPN^-) untuk mendapatkan *output* jaringan $g^+(x_k)$ dan $g^-(x_k)$ yang berkaitan dengan kondisi sebagai berikut [8] :

$$g^-(x) \leq d_k \leq g^+(x), k = 1, 2, \dots, p \quad (2)$$

Pada proses pembelajaran BPN^+ , fungsi biaya yang digunakan adalah[5] :

$$E = \sum_{k=1}^p E_k = \frac{1}{2} \sum_{k=1}^p \alpha_k [d_k - g^+(x_k)]^2 \quad (3)$$

Dengan α_k diberikan sebagai berikut[5] :

$$\alpha_k = \begin{cases} 1; & d_k > g^+(x_k) \\ \alpha; & d_k \leq g^+(x_k) \end{cases} \quad (4)$$

Dengan α adalah bilangan positif yang cukup kecil pada interval (0,1). Nilai α ini dapat diberikan dengan menggunakan fungsi penurunan sebagai[5] :

$$\alpha(t) = \frac{1}{[1 + (\frac{t}{2000})^8]} \quad (5)$$

Dengan t adalah iterasi ke.

$$\Delta W_j = \eta \left(-\frac{\partial E_k}{\partial W_j} \right) = \eta \delta_k y_{kj} \quad (6)$$

$$\Delta W_{ji} = \eta \left(-\frac{\partial E_k}{\partial W_{ji}} \right) = \eta \delta_{kj} x_{ki} \quad (7)$$

Dengan w_j (bobot antara lapisan tersembunyi dengan lapisan *output*) dan w_{ji} (bobot antara lapisan *input* dengan lapisan tersembunyi) pada jaringan BPN^+ , didapat dari Aturan pembelajaran *backpropagation* dengan 3 lapisan (1 lapisan *input*, 1 lapisan tersembunyi, dan 1 lapisan *output*) dan fungsi aktivasi sigmoid.

$$\delta_k = \alpha_k (d_k - y_k) y_k (1 - y_k) \quad (8)$$

$$\delta_{kj} = y_{kj} (1 - y_{kj}) \delta_k w_j \quad (9)$$

$y_k = g^+(x_k)$ adalah *output* jaringan, dan y_{kj} adalah *output* neuron ke- j pada lapisan tersembunyi untuk *input* x_k .

Dengan cara yang sama, pembelajaran juga dilakukan pada jaringan BPN^- untuk mendapatkan *output* jaringan $g^-(x_k)$. Fungsi biaya yang digunakan untuk pembelajaran ini seperti terlihat pada persamaan (3), dengan α_k diberikan sebagai berikut[5]:

$$\alpha_k = \begin{cases} \alpha; & d_k \geq g^-(x_k) \\ 1; & d_k < g^-(x_k) \end{cases} \quad (10)$$

Dengan α adalah bilangan positif yang cukup kecil pada interval (0, 1).

Dengan menggunakan kedua algoritma pembelajaran tersebut, kita dapat menentukan 2 fungsi, $g^+(x)$ dan $g^-(x)$ dimana $g^-(x) \leq d_k \leq g^+(x)$, $k = 1, 2, \dots, p$. Dari sini, dapat diperoleh interval[5]

$$G(x) = [g^-(x), g^+(x)] \quad (11)$$

Karena $g^+(x)$ dan $g^-(x)$ diperoleh dari pembelajaran yang terpisah, maka sangat dimungkinkan $g^+(x) < g^-(x)$. Sehingga, jika hal tersebut terjadi, maka dapat dilakukan modifikasi pada interval tersebut sebagai berikut[5]:

$$h^-(x) \begin{cases} g^-; & g^-(x_k) \leq g^+(x) \\ \frac{1}{2}(g^-(x) + g^+(x)); & g^-(x_k) > g^+(x) \end{cases} \quad (12)$$

$$h^+(x) \begin{cases} g^+; & g^-(x_k) \leq g^+(x) \\ \frac{1}{2}(g^-(x) + g^+(x)); & g^-(x_k) > g^+(x) \end{cases} \quad (13)$$

Dengan demikian, interval yang terjadi adalah[5] :

$$G(x) = [h^-(x), h^+(x)] \quad (14)$$

2.5 MATLAB

MATLAB (Matrix Laboratory) adalah sebuah program untuk analisis dan komputasi numerik dan merupakan suatu bahasa pemrograman matematika lanjutan yang dibentuk dengan dasar pemikiran menggunakan sifat dan bentuk matriks. MATLAB merupakan sebuah bahasa dengan kemampuan tinggi untuk komputasi teknis. MATLAB menggabungkan komputasi, visualisasi, dan pemrograman dalam satu kesatuan yang mudah digunakan dimana masalah dan penyelesaian diekspresikan dalam notasi matematik yang sudah di kenal. Matlab dapat dipadang sebagai sebuah kalkulator dengan fitur yang lengkap. Kita pernah menggunakan kalkulator dengan degan fasilitas minimal, misalnya hanya terdapat fasilitas penambahan, pengurangan perkalian dan pembagian. Kalkulator yang lebih lengkap lagi adalah kalkulator scientific dimana fasilitas yang diberikan tidak hanya yang disebutkan di atas, melainkan sudah ada fungsi-fungsi trigonometri, bilangan kompleks, akar kuadrat dan logaritma. Matlab mirip dengan kalkulator tersebut, tetapi dengan fitur-fitur yang lengkap diantaranya dapat digunakan untuk memprogram, aplikasi berbasis GUI dan lengkap dengan toolbox yang dapat dimanfaatkan untuk memecahkan masalah sains dan teknik. Bahasa pemrograman pada MATLAB banyak digunakan untuk perhitungan numerik keteknikan, komputasi simbolik, visualisasi grafis, analisis data matematis, statistika, simulasi pemodelan, dan desain GUI (*graphical user interface*)[10].

2.6 Beban Listrik

Secara umum beban yang dilayani oleh sistem distribusi elektrik ini dibagi dalam beberapa sektor yaitu sektor perumahan, sector industri, sektor komersial dan sektor usaha. Tiap sektor beban tersebut mempunyai karakteristik-karakteristik yang berbeda, sebab hal ini berkaitan dengan pola konsumsi energi. Sedangkan tipe-tipe beban menurut konsumen pemakainya pada umumnya dapat dikelompokkan dalam kategori berikut :

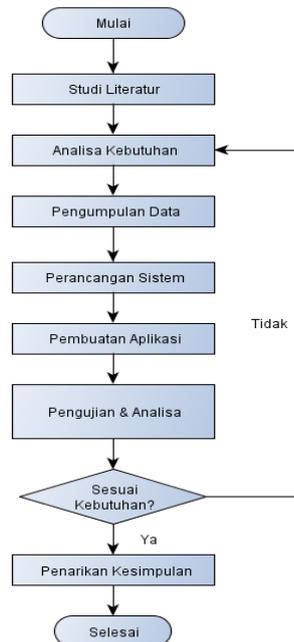
- Rumah Tangga (domestik/residen), terdiri dari beban-beban penerangan, kipas angin, alat-alat rumah tangga misalnya pemanas, lemari es, kompor listrik, dan lain-lain.

- Bisnis, terdiri atas beban penerangan dan alat listrik lainnya yang dipakai pada bangunan komersil atau perdagangan seperti toko, restoran, dan lain-lain.
- Umum/publik, terdiri dari pemakai selain ketiga golongan di atas misalnya gedung pemerintah, penerangan jalan umum, dan pemakai kepentingan sosial.
- Industri, terdiri dari industri kecil/rumah tangga hingga industri besar. Umumnya bebannya berupa beban untuk motor listrik.

Pada masing-masing konsumen di sektor tersebut. Karakteristik beban yang banyak disebut dengan pola pembebanan pada sektor perumahan ditujukan oleh adanya fluktuasi konsumsi energi elektrik yang cukup besar. Hal ini disebabkan konsumsi energi elektrik tersebut dominan pada malam hari. Sedang pada sektor industri fluktuasi konsumsi energi sepanjang hari akan hampir sama, sehingga perbandingan beban puncak terhadap beban rata-rata hampir mendekati satu. Beban pada sektor komersial dan usaha mempunyai karakteristik yang hampir sama, hanya pada sektor komersial akan mempunyai beban puncak yang lebih tinggi pada malam hari[11].

3. METODE PENELITIAN

Gambaran umum proses penelitian prediksi beban listrik menggunakan regresi interval dengan neural fuzzy dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

Pada Gambar 1 dapat dilihat bahwa penelitian ini dimulai dengan studi literatur atau mencari bahan tentang metode yang dipakai pada penelitian ini kemudian dilanjutkan dengan analisa kebutuhan yang diperlukan dalam pembuatan aplikasi ini. Selanjutnya dilakukan pengumpulan data yang

dialbil dari PT.PLN(PERSERO) area pengatur distribusi dan penyaluran kota Pontianak dan Singkawang. Tahap selanjutnya yaitu perancangan sistem yang akan dibuat. Setelah sistem selesai dirancang maka dilanjutkan ke tahap pembuatan sistem. Sistem yang sudah selesai dibuat kemudian akan diuji untuk mengetahui apakah sistem sudah memenuhi kebutuhan. Jika sistem sudah memenuhi kebutuhan yang ada maka bisa ditarik kesimpulan bahwa sistem sudah siap digunakan. Namun jika sistem masih memiliki kekurangan maka proses akan dikembalikan ke tahap analisa kebutuhan.

4. PERANCANGAN SISTEM

4.1 Penetapan Masukan

Masukan yang digunakan dalam aplikasi ini adalah berupa data beban puncak harian selama tujuh hari. Data historis yang digunakan untuk memprediksikan beban puncak harian jangka pendek menggunakan data beban puncak listrik harian dari PT.PLN (Persero) Area Pengaturan Distribusi dan Penyaluran (APDP) Bidang Operasi Sistem Pontianak-Singkawang dari bulan November 2013 sampai Februari 2014. 85 data akan dijadikan sebagai data pelatihan sedangkan 28 data akan dijadikan data pengujian.

4.2 Penetapan Keluaran

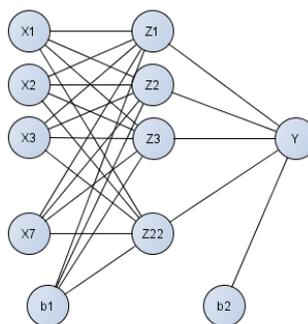
Keluaran yang akan dihasilkan pada penelitian ini adalah prediksi beban puncak harian untuk hari kedelapan. Pada aplikasi ini pengguna akan diminta untuk memasukkan terlebih dahulu data beban listrik selama tujuh hari. Setelah itu akan keluar prediksi beban listrik untuk hari kedelapan.

4.3 Arsitektur Jaringan

Arsitektur jaringan pada Jaringan syaraf tiruan memiliki beberapa lapisan, sesuai dengan jumlah neuron yang dibutuhkan. Arsitektur jaringan syaraf tiruan *backpropogation* yang digunakan yaitu *multilayer network*. *Multilayer network* menggunakan satu atau lebih lapisan tersembunyi yang terletak diantara lapisan *input* dan lapisan *output*. Parameter-parameter yang digunakan pada proses pelatihan jaringan syaraf tiruan untuk membentuk jaringan terbaik, yaitu: jumlah masukan sistem, jumlah lapisan tersembunyi, jumlah sel lapisan tersembunyi, jumlah keluaran sistem, target *error*, fungsi aktivasi, laju pembelajaran, momentum, dan maksimum iterasi. Untuk mendapatkan jaringan terbaik maka dilakukan metode *trial and error* (coba-coba) dalam menentukan parameter-parameter pelatihannya. Gambar 2 menunjukkan arsitektur jaringan yang digunakan.

Arsitektur jaringan *backpropogation* terdiri atas 7 unit (neuron) pada lapisan input, yaitu $X_1, X_2,$ hingga X_7 dan satu lapisan tersembunyi dengan 22 sel lapisan tersembunyi, yaitu $Z_1, Z_2,$ hingga Z_{22}

serta 1 unit pada lapisan output yaitu, Y. Pada Gambar 2, b_1 adalah bobot bias yang menuju ke neuron pertama hingga neuron ke duapuluh dua pada lapisan tersembunyi. Bobot bias yang menghubungkan lapisan tersembunyi dengan lapisan output yaitu b_2 . Fungsi aktivasi yang digunakan antara lapisan input dan lapisan tersembunyi serta antara lapisan tersembunyi dengan lapisan output adalah fungsi aktivasi sigmoid biner.



Gambar 2. Arsitektur dengan 1 lapisan tersembunyi dan 22 sel lapisan tersembunyi

Keterangan :

X_1-X_7 = Data masukan sistem

Z_1-Z_{22} = Lapisan tersembunyi

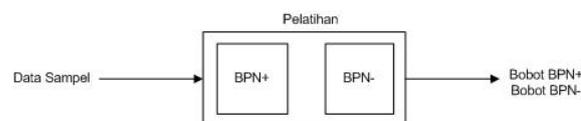
Y = Keluaran

b_1 =Bobot bias yang menghubungkan lapisan *input* dan lapisan tersembunyi

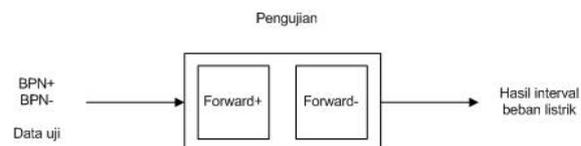
b_2 =Bobot bias yang menghubungkan lapisan tersembunyi dan lapisan *output*

4.4 Perancangan Prosedural

Perancangan alur proses penelitian digunakan untuk menggambarkan secara umum proses yang terjadi pada sistem. Gambar 3 menunjukkan alur proses pelatihan, sedangkan gambar 4 menunjukkan alur proses pengujian.



Gambar 3. Blok Diagram Pelatihan



Gambar 4. Blok Diagram Pengujian

Keterangan :

BPN^+ = Pelatihan menggunakan algoritma *Backpropogation* untuk mencari bobot pada batas atas interval

BPN^- = Pelatihan menggunakan algoritma *Backpropogation* untuk mencari bobot pada batas bawah interval

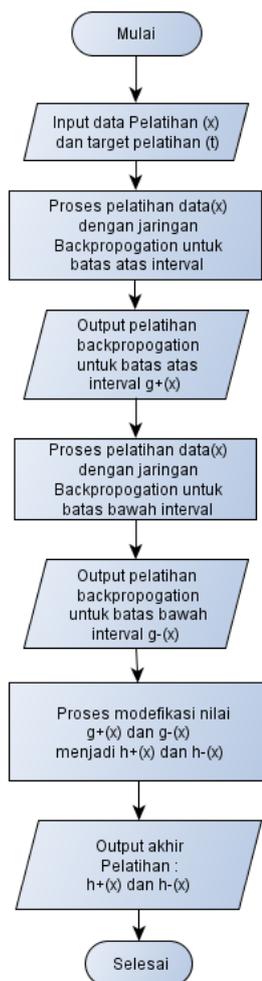
Forward⁺ = Pengujian menggunakan algoritma Feedforward untuk data uji dengan bobot BPN⁺

Forward⁻ = Pengujian menggunakan algoritma Feedforward untuk data uji dengan bobot BPN⁻

Data sampel digunakan sebagai data pelatihan. Data sampel tersebut kemudian akan dilatih menggunakan algoritma *backpropogation* untuk memperoleh nilai bobot-bobot yang nantinya akan digunakan pada tahap pengujian. Proses pelatihan dibagi menjadi dua, pelatihan pertama dilakukan untuk mencari nilai bobot batas atas interval dan pelatihan kedua dilakukan untuk mencari nilai batas bawah interval. Setelah bobot-bobot tersebut didapatkan, maka dilanjutkan ke tahap pengujian. Nilai bobot tersebut dan data uji akan digunakan pada proses pengujian dengan menggunakan algoritma *feedforward* untuk mendapatkan hasil akhir nilai beban listrik.

4.5 Tahap Pelatihan

Tahap pelatihan pada penelitian ini dapat dilihat pada Gambar 5.



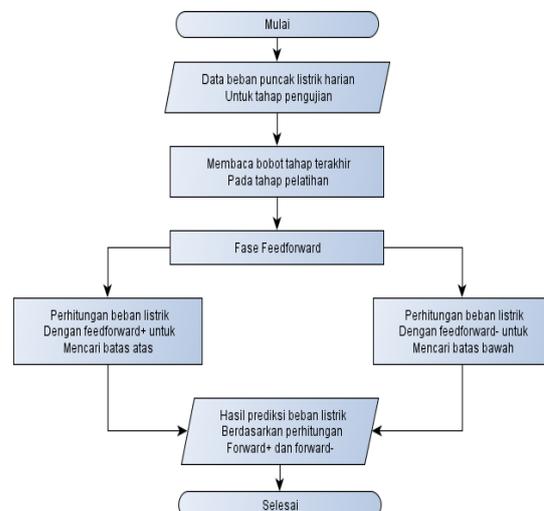
Gambar 5. Diagram alir tahap pelatihan

Pada aplikasi ini terjadi dua kali pelatihan. Tahap pelatihan pertama dilakukan untuk mencari batas atas interval dan tahap pelatihan kedua untuk mencari batas bawah interval. Hal pertama yang harus ditentukan untuk mencari batas atas adalah menginisialisai bobot, menetapkan konfigurasi dari jaringan syaraf tiruan yaitu maksimum epoch, target error dan *learning rate*. Selanjutnya masuk ketahap alur maju atau *feedforward*, kemudian dilanjutkan pada tahap *backpropagation*, dan menghitung MSE yang didapat, jika error yang didapat lebih besar dari target error yang diijinkan, maka proses pelatihan tersebut akan kembali pada tahap inialisasi bobot, dan seterusnya sampai didapat nilai error yang lebih kecil dari target error yang diijinkan. Jika proses pelatihan untuk mencari nilai batas atas interval sudah tercapai maka dilanjutkan dengan mencari nilai batas bawah interval. Proses akan langsung ke tahap alur maju atau *feedforward*, kemudian dilanjutkan pada tahap *backpropagation*, dan menghitung MSE yang didapat, jika error yang didapat lebih besar dari target error yang diijinkan, maka proses pelatihan tersebut akan kembali pada tahap inialisasi bobot dan seterusnya sampai didapat nilai error yang lebih kecil dari target error yang diijinkan. Jika nilai yang diinginkan telah didapat, bobot tersebut tersimpan untuk dimasukkan pada tahap pengujian.

4.4 Tahap Pengujian

Diagram alir untuk tahap pengujian ditunjukkan pada Gambar 6. Penjelasan sebagai berikut:

1. Input data beban listrik puncak harian
2. Proses pembacaan bobot terakhir pada tahap pelatihan
3. Proses perhitungan nilai interval beban listrik dengan menggunakan *feedforward* berdasarkan masukan data beban listrik dan bobot-bobot dari hasil pelatihan
4. Hasil prediksi nilai interval beban listrik



Gambar 6. Diagram alir tahap pengujian

5. PENGUJIAN DAN ANALISA

5.1 Pelatihan JST

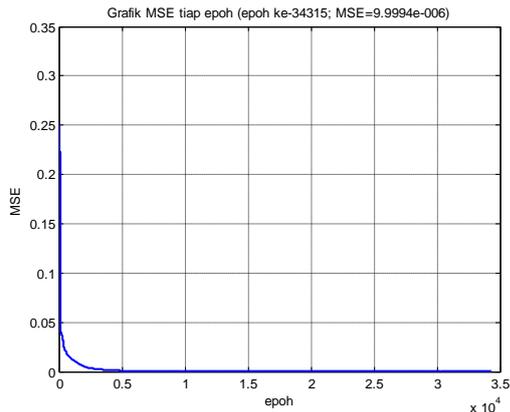
Pelatihan Jaringan Syaraf Tiruan menggunakan data beban listrik yang sudah dinormalisasikan. Ada 85 data yang bersifat *time series* yang akan dijadikan sebagai variabel masukan. Untuk parameter jaringan, dalam tahap ini dilakukan perubahan parameter untuk melihat parameter manakah yang menghasilkan nilai *error* paling kecil atau menghasilkan jaringan yang terbaik. Parameter yang diubah antara lain adalah jumlah sel lapisan tersembunyi, laju pembelajaran, momentum, dan maksimum iterasi. Setelah dilakukan 30 kali percobaan dengan mengubah sel lapisan tersembunyi mulai dari satu sel lapisan tersembunyi hingga tiga puluh sel lapisan tersembunyi. Setelah melakukan perubahan terhadap jumlah sel lapisan tersembunyi didapat hasil bahwa jaringan yang terbaik terbentuk menggunakan 22 sel lapisan tersembunyi. Laju pembelajaran juga merupakan salah satu parameter yang diubah untuk mencari jaringan terbaik pada proses pelatihan. Setelah dilakukan perubahan terhadap laju pembelajaran sebanyak tiga kali yang bernilai 0,2, 0,4 dan 0,5 didapat jaringan terbaik menggunakan laju pembelajaran sebesar 0,5. Semakin besar nilai laju pembelajaran serta toleransi kesalahan yang ada maka kecenderungannya akan semakin mempercepat proses pembelajaran tetapi semakin kecil akurasi hasil keluaran. Perubahan parameter selanjutnya yang dilakukan untuk mencari jaringan terbaik adalah melakukan perubahan nilai momentum. Ini dimaksudkan untuk menghindari perubahan bobot yang mencolok akibat adanya data yang sangat berbeda dengan yang lainnya. Jumlah neuron pada layar tersembunyi mempengaruhi kemampuan pengenalan pola dari jaringan. Pada fungsi aktivasi sigmoid biner jumlah neuron pada lapisan tersembunyi harus sebanding dengan besar momentumnya. Apabila jumlah neuron pada layar tersembunyi semakin banyak, maka agar mendapatkan hasil yang lebih optimal nilai momentumnya-pun harus diperbesar. Perbandingan besarnya momentum harus disesuaikan dengan kebutuhan jumlah neuron pada layar tersembunyi pada jaringan tersebut. Semakin banyak neuron pada layar tersembunyi kecepatan proses pembelajaran semakin lambat, hal ini terjadi karena jumlah bobot yang diproses semakin banyak. Untuk mengetahui pengaruh dari jumlah data latih terhadap hasil prediksi maka dilakukan dua kali pelatihan dengan menggunakan parameter-parameter terbaik yang telah didapat sebelumnya. Pelatihan pertama menggunakan 85 data pelatihan dan 28 data pengujian dengan arsitektur jaringan 1 lapisan tersembunyi dan 22 sel lapisan tersembunyi menghasilkan akurasi prediksi sebesar 71,42%. Pelatihan kedua menggunakan 115 data pelatihan dan 28 data pengujian dengan arsitektur 1 lapisan

tersembunyi dan 22 sel lapisan tersembunyi serta maksimum epoch sebanyak 15000 menghasilkan akurasi prediksi sebesar 53,57%. Pelatihan pertama memerlukan waktu kurang lebih 2 jam 20 menit untuk menyelesaikan kedua pelatihan jaringan sedangkan pelatihan kedua memerlukan waktu 3 jam 40 menit untuk menyelesaikan kedua pelatihan jaringan. Dari kedua pelatihan tersebut dapat diambil kesimpulan bahwa jumlah data latih berpengaruh terhadap lamanya waktu pelatihan namun tidak berpengaruh terhadap hasil akhir prediksi. Semakin banyak data latih maka waktu pelatihan akan semakin lama dan maksimum epoch juga semakin besar. Untuk mendapatkan jaringan terbaik maka dilakukan metode *trial and error* dalam menentukan nilai parameter-parameter tersebut. Tabel 1 menunjukkan parameter pelatihan untuk jaringan terbaik

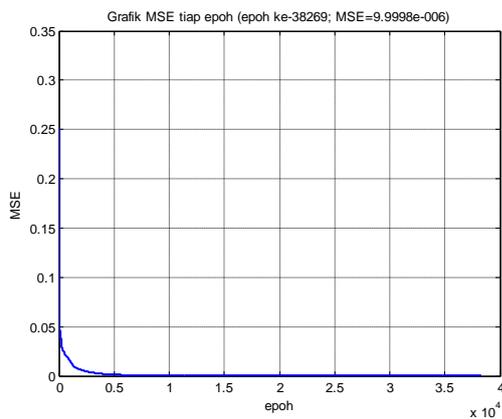
Tabel 1 parameter pelatihan untuk jaringan terbaik

Parameter	Nilai
Jumlah masukan sistem	7
Jumlah data latih	85
Jumlah lapisan tersembunyi	1
Jumlah sel lapisan tersembunyi	22
Jumlah keluaran sistem	1
Besar Galat	0,00001
Fungsi Aktivasi	<i>Sigmoid biner</i>
Learning rate	0,5
Maksimum Iterasi	50000
Momentum	0,9

Proses pelatihan jaringan untuk mencari BPN⁺ selesai dengan waktu 75 menit. Pelatihan Jaringan Syaraf Tiruan untuk mencari BPN⁺ dengan menggunakan arsitektur jaringan 1 lapisan tersembunyi dan 22 sel lapisan tersembunyi berhenti pada epoch ke 34315 dan mencapai target error 0,00001. Grafik proses pelatihan mencari BPN⁺ ditunjukkan pada Gambar 7 sedangkan untuk pelatihan mencari BPN⁻ Dengan waktu 65 menit pelatihan Jaringan Syaraf Tiruan untuk mencari BPN⁻ dengan menggunakan arsitektur jaringan 1 lapisan tersembunyi dan 22 sel lapisan tersembunyi berhenti pada epoch ke 38296 dan mencapai target error 0,00001. Grafik proses pelatihan mencari BPN⁺ ditunjukkan pada Gambar 8.



Gambar 7. Hasil pembelajaran BPN⁺



Gambar 8. Hasil pembelajaran BPN⁻

Setelah dilakukan pelatihan terhadap BPN⁺ dan BPN⁻ maka akan diperoleh bobot akhir bias dari lapisan-lapisan *input* ke lapisan tersembunyi, bobot akhir dari lapisan-lapisan *input* ke lapisan tersembunyi, bobot akhir bias dari lapisan-lapisan tersembunyi ke lapisan *output* dan bobot akhir dari lapisan-lapisan tersembunyi ke lapisan *output*. Bobot ini akan disimpan untuk digunakan pada proses pengujian.

5.1 Pengujian Aplikasi

Pada proses pengujian ini digunakan algoritma *feedforward*. Pengujian dilakukan untuk mengetahui apakah jaringan mampu mengenali pola pelatihan dari masukan yang telah diberikan. Pengujian menggunakan data-data yang tidak dilatih sebelumnya. Keluaran dari proses pengujian ini berupa prediksi data beban listrik untuk hari kedelapan. Data yang digunakan dalam proses ini menggunakan 28 buah data beban listrik yang akan dibaca sebagai pola masukan pada sistem. Masukan ini kemudian akan dihitung menggunakan bobot yang telah tersimpan pada proses pelatihan sebelumnya sehingga proses ini akan menghasilkan nilai hasil prediksi beban listrik. Hasil pengujian menggunakan arsitektur jaringan 1 lapisan tersembunyi dan 22 buah sel lapisan tersembunyi serta 28 data pengujian dapat dilihat pada Tabel 2.

Tabel 2. Hasil pengujian validitas sistem

Data ke-	Batas Bawah Interval	Target	Batas Atas Interval	Keterangan
1	208,93	207	224,83	F
2	211,78	206,7	231,33	F
3	207,41	215,3	232,55	T
4	211,63	221,3	228,80	T
5	210	218,1	224,40	T
6	209,05	225,2	213,59	F
7	207,14	224,6	219,15	F
8	205,03	221,7	222,50	T
9	204,55	221,2	222,25	T
10	204,79	226,4	231,62	T
11	217,94	227,4	232,81	T
12	211,17	229,1	234,40	T
13	217,16	230,4	234,15	T
14	208,98	227,7	233,61	T
15	205,87	204,1	231,71	F
16	220,61	215,6	232,99	F
17	204,81	226,7	235,02	T
18	212,43	230,9	236,01	T
19	208,68	236,1	235,61	F
20	208,73	223,1	235,40	T
21	204,77	230,3	223,64	F
22	204,23	219,5	223,18	T
23	205,33	215,2	234,85	T
24	204,64	229,6	239,39	T
25	226,91	227,7	236,04	T
26	209,86	231,1	235,83	T
27	218,46	224,6	233,88	T
28	205,47	224,8	234,35	T

Keterangan pada Tabel 2 menunjukkan T=true, terjadi apabila nilai target berada diantara nilai batas atas dan nilai batas bawah interval sedangkan F=false, terjadi apabila nilai target berada diatas nilai batas atas atau dibawah nilai batas bawah interval.

Contoh pada data no.1

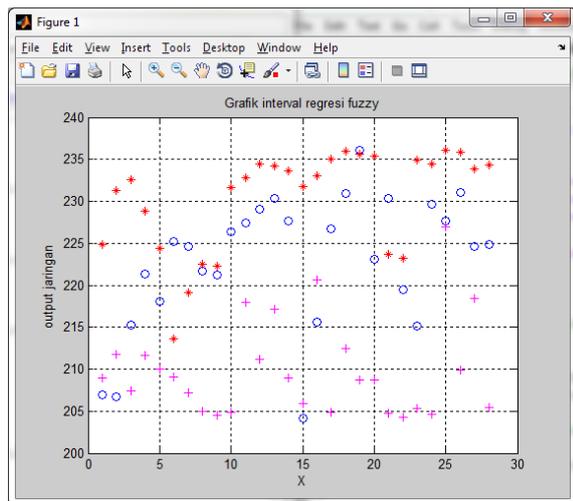
Data nomor 1

Batas bawah : 208,93

Batas atas : 224,83

Target : 207

Untuk melihat perbandingan nilai target dengan nilai batas atas dan nilai batas bawah interval dapat dilihat pada gambar 7



Gambar 7. Grafik hasil pengujian

Gambar 7 menunjukkan grafik hasil pengujian. Tanda (*) menunjukkan data hasil prediksi untuk batas atas interval sedangkan tanda (+) menunjukkan data hasil prediksi untuk batas bawah interval serta tanda (o) menunjukkan data target real. Dari hasil pengujian validitas sistem yang telah dilakukan diperoleh hasil bahwa jika nilai target yang berada diatas nilai batas atas atau nilai target yang berada dibawah nilai batas bawah interval maka nilai yang dihasilkan tersebut tidak valid. Besarnya kinerja sistem dapat dihitung menggunakan rumus:

Kinerja Sistem

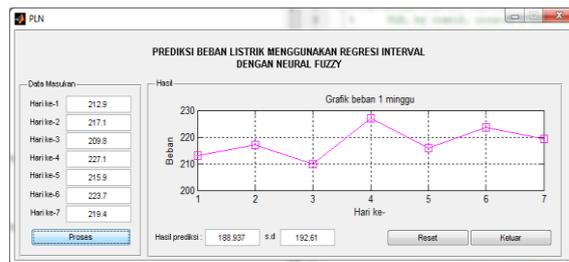
$$= \frac{\text{jumlah hasil pengujian yang benar}}{\text{jumlah data pengujian}} \times 100\%$$

Kinerja sistem = $20/28 \times 100\%$
= 71,42%

Maka setelah dilakukan perhitungan terhadap hasil pengujian diperoleh kinerja sistem sebesar 71,42%.

5.2 Program Aplikasi

Implementasi antarmuka dilakukan setelah proses pengujian mendapatkan hasil yang akurat atau hampir akurat. Rancangan antarmuka untuk prediksi beban listrik yang telah dibuat sebelumnya akan diimplementasikan ke dalam sebuah program aplikasi. Aplikasi ini dibuat menggunakan *Graphical User interface* (GUI) yang digunakan untuk menghubungkan antara pengguna dengan aplikasi yang telah dibuat. Aplikasi ini akan membantu pengguna untuk memasukkan data baru sehingga mempermudah dalam prediksi beban listrik.



Gambar 8 Tampilan Utama Aplikasi

Untuk melakukan prediksi menggunakan aplikasi ini pengguna terlebih dulu diminta untuk mengisi variabel-variabel masukan yang berupa data beban listrik selama tujuh hari. Kemudian pengguna dapat menekan tombol prediksi agar aplikasi dapat mulai memproses data dan akhirnya dapat menampilkan hasil prediksan beban listrik.

6. KESIMPULAN & SARAN

6.1 Kesimpulan

Berdasarkan hasil analisis dan pengujian terhadap Sistem prediksi beban listrik menggunakan regresi interval dengan neural fuzzy maka kesimpulan yang dapat diperoleh pada penelitian ini adalah sebagai berikut :

1. Model regresi interval dengan neural fuzzy dapat digunakan untuk menentukan prediksi beban listrik di kota Pontianak dan kota Singkawang menggunakan data historis sebagai variabel masukan, yang kemudian dinormalisasikan untuk memperoleh data masukan dan keluaran yang digunakan untuk perhitungan.
2. Informasi yang diberikan oleh sistem berupa prediksi beban puncak listrik harian untuk waktu tertentu.
3. Berdasarkan perubahan parameter yang dilakukan didapat jaringan terbaik menggunakan maksimum iterasi = 50000, laju pembelajaran = 0,5, momentum = 0,9, target error = 0,00001 dan jumlah sel lapisan tersembunyi = 22 menghasilkan tingkat akurasi 71,42%.

6.2 Saran

Hal-hal yang menjadi saran untuk pengembangan sistem prediksi beban listrik menggunakan regresi interval dengan neural fuzzy agar menjadi lebih baik kedepannya sebagai berikut :

1. Penelitian selanjutnya dapat menambahkan faktor-faktor eksternal yang ikut berperan dalam memprediksi pemakaian beban listrik.
2. Pada penelitian selanjutnya disarankan melakukan perbandingan metode dalam memprediksi beban listrik untuk menghasilkan prediksi yang lebih akurat.

3. Pengembangan sistem dapat dilakukan dengan menambah jangka waktu prediksi beban listrik menjadi prediksi beban listrik jangka panjang.

DAFTAR PUSTAKA

- [1] Fitriyah, Q., & Istardi, D. (2012). *Prediksi Beban Listrik Pulau Bali Dengan Menggunakan Metode Backpropagasi*.
- [2] Handoko, B. (2013). *Peramalan Beban Listrik Jangka Pendek pada Sistem Kelistrikan Jawa Timur dan Bali Menggunakan Fuzzy Time Series*.
- [3] Kusumadewi, S. (2003). *Artificial Intelligence Teknik dan Aplikasinya*. Yogyakarta: Graha Ilmu.
- [4] Fausett, L. (1994). *Fundamental of Neural Networks (Architectures, Algorithms, and Application)*. New Jersey: Prentice-Hall.
- [5] Kusumadewi, S., & Hartati, S. (2010). *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf*. Yogyakarta: Graha Ilmu.
- [6] Nauck, D., Klawoon, F., & Kruse, R. (1997). "Foundations of Neuro Fuzzy System", dalam Mitra, Sushmita; dan Hayashi, Yoici. 2000. "Neuro-Fuzzy Rule Generation: Survey in Soft Computing Framework". IEEE Transaction on Neural Network vol 11 no 3.
- [7] Pal, S.K, & Ghosh, A. (1996). *Neuro Fuzzy Computing for Image Processing and Pattern Recognition*. International Journal of System Science vol 27 no 12.
- [8] Lin, C.-T., & Lee, G. (1996). *Neural Fuzzy Systems*. London: Prentice-Hall.
- [9] Jang, J., Sun, C., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*. London: Prentice-Hall.
- [10] Siang, J. J. (2009). *Jaringan Syaraf Tiruan dan Programnya menggunakan Matlab*. Yogyakarta: Penerbit Andi.
- [11] Marsudi, D. (2006). *Pembangkitan Energi Listrik*. Jakarta: Erlangga.