

Sistem Informasi Peramalan Beban Listrik Jangka Panjang di Kabupaten Jember Menggunakan JST *Backpropagation*

Dodi Setiabudi

Abstrak—Kebutuhan energi listrik sebagai salah satu infrastruktur penting sangat diutamakan. Pertumbuhan ekonomi pada suatu daerah menyebabkan peningkatan konsumsi energi listrik. Pemenuhan kebutuhan energi listrik pada suatu daerah diperlukan perencanaan sistem tenaga listrik yang tepat.

Penelitian ini mengembangkan suatu model system informasi peramalan beban puncak listrik jangka panjang menggunakan metode Jaringan syaraf tiruan (JST) *backpropagation*. Hasil penelitian menunjukkan bahwa peramalan beban puncak listrik menggunakan JST *backpropagation* memiliki rata-rata *Mean Absolute Percentage Error (MAPE)* sebesar 17,09%,

Kata Kunci—Peramalan beban puncak listrik jangka panjang, Sistem Informasi, Jaringan Syaraf Tiruan, *Backpropagation*.

I. PENDAHULUAN

PENAMBAHAN panjang jaringan transmisi tenaga listrik mengalami peningkatan sebesar 11% selama periode 5 tahun sejak tahun 2004 [3], perkembangan sektor industri dan pariwisata diramalkan mengalami pertumbuhan beban dengan pesat. Penyediaan tenaga listrik harus menjadi prioritas dalam pembangunan dan dalam prosesnya harus dikembangkan dengan prinsip-prinsip keterandalan. Dengan menggunakan SPK yang baik diharapkan membantu pihak PLN sebagai pengguna SPK untuk memperkirakan, mengalokasikan kapasitas kebutuhan beban listrik untuk periode kedepan dan perencanaan pemilihan lokasi penempatan trafo distribusi pada pengembangan jaringan baru yang tepat merujuk pada hasil peramalan beban listrik jangka panjang.

Peramalan beban listrik adalah penopang dasar untuk operasi manajemen dan kontrol *real-time* sistem operasi listrik [7]. Sebuah perencanaan sistem tenaga listrik dibuat berbasis hasil estimasi beban listrik.

Metode prakiraan beban listrik jangka panjang dengan rentang 1 sampai 10 tahun [1] selama ini yang digunakan PLN adalah dengan cara konvensional berbasis metode statistik, yaitu metode koefisien beban yang dilakukan dengan mempelajari data historis beban listrik terdahulu.

JST adalah algoritma penyelesaian masalah komputasi yang prinsip kerjanya menirukan jaringan syaraf manusia. Salah satu jenis dari metode JST yaitu *Backpropagation*. Metode ini dipilih karena saat *output* tidak sama dengan target yang diharapkan maka *output* akan disebarkan mundur (*backward*) pada *hidden layer* untuk diteruskan ke unit pada *input layer*, sehingga akan ada umpan balik untuk memvalidasi hasil keluaran JST [3]. Pada penelitian ini Metode *JST Backpropagation* digunakan untuk subsistem peramalan beban puncak listrik jangka panjang.

Menjawab berbagai permasalahan dalam hal perencanaan sistem tenaga listrik yang tepat maka dibutuhkan suatu system informasi peramalan beban listrik jangka panjang di kabupaten Jember.

II. DASAR TEORI

A. Peramalan beban listrik jangka panjang

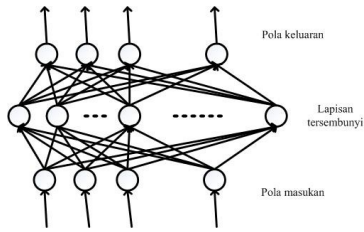
Secara umum, beban yang dilayani oleh sistem distribusi listrik ini dibagi dalam beberapa sektor yaitu sektor perumahan, sektor industri, sektor komersial, dan sektor publik atau umum. Masing – masing sektor mempunyai karakteristik – karakteristik yang berbeda. Ramalan dalam bidang tenaga listrik pada dasarnya merupakan ramalan kebutuhan energi listrik (watt jam) dan ramalan beban tenaga listrik (watt). Keduanya disebut dengan istilah *load (demand) forecasting*. Hasil peramalan ini dipergunakan untuk membuat rencana pemenuhan kebutuhan maupun pengembangan penyediaan tenaga listrik setiap saat secara baik dan kontinyu. Peramalan beban listrik jangka panjang adalah untuk jangka waktu di atas satu tahun [1].

B. Jaringan syaraf tiruan *backpropagation*

Jaringan syaraf tiruan adalah suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan syaraf biologi [2]. Ide dasarnya adalah mengadopsi cara kerja otak manusia yang memiliki ciri – ciri *pararel processing*, *processing element* dalam jumlah besar dan *faulttolerance*.

Dodi Setiabudi adalah Dosen Jurusan Teknik Elektro, Fakultas Teknik Universitas Jember, Jember, Indonesia (email: gbdoydist@gmail.com).

Jaringan propagasi balik terdiri atas tiga atau lebih lapisan atau *layer*[5]. Perbedaannya hanya pada jumlah lapisan tersembunyi yang dimiliki.



Gambar. 1. Lapisan Jaringan *Backpropagation*

Algoritma selengkapnya pelatihan jaringan *backpropagation* adalah sebagai berikut [2]:

1. Langkah 0 : Inisialisasi bobot-bobot (tetapkan dalam nilai acak kecil)
2. Langkah 1 : Bila syarat berhenti adalah salah, kerjakan langkah 2 sampai 9.
3. Langkah 2 : Untuk setiap pasangan pelatihan, kerjakan langkah 3 sampai 8.

Umpan maju :

4. Langkah 3 : Tiap unit masukan ($x_i, i = 1, \dots, n$) menerima isyarat masukan x_i dan diteruskan ke unit-unit tersembunyi.
5. Langkah 4 : Tiap unit tersembunyi ($z_j, j = 1, \dots, p$) menjumlahkan isyarat masukan berbobot,

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

dengan menerapkan fungsi aktivasi hitung :

$$Z_j = f(z_in_j) \quad (2)$$

dan kirim isyarat ini ke unit-unit keluaran.

6. Langkah 5 : Tiap unit keluaran ($y_k, k = 1, \dots, m$) menjumlahkan isyarat masukan berbobot,

$$y_in_j = w_{0k} + \sum_{i=1}^p z_i w_{jk} \quad (3)$$

dengan menerapkan fungsi aktivasi hitung,

$$y_k = f(y_in_k) \quad (4)$$

Perambatan Balik *Error* :

7. Langkah 6 : Tiap unit keluaran ($y_k, k = 1, \dots, m$) menerima pola sasaran berkaitan dengan pola pelatihan masukannya.

Hitung error informasi :

$$\delta_k = (t_k - y_k) f'(y_in_k) \quad (5)$$

Hitung koreksi bobot dan biasnya:

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

$$\Delta w_{0k} = \alpha \delta_k \quad (7)$$

8. Langkah 7 : Tiap unit tersembunyi ($z_j, j = 1, \dots, p$) menjumlahkan delta masukannya (dari unit-unit di lapisan atasnya).

$$\delta_in_j = \sum_{i=1}^m \delta_k w_{jk} \quad (8)$$

Hitung error informasinya :

$$\delta_j = \delta_in_j f'(z_in_j) \quad (9)$$

Hitung koreksi bobot dan biasnya :

$$\Delta v_{ij} = \alpha \delta_j x_i \quad (10)$$

Perbaharui bobot dan bias :

9. Langkah 8 : Tiap-tiap unit output ($Y_k, k=1,2,3,\dots,m$) memperbaiki bias dan bobotnya ($j=0,1,2,\dots,p$):

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

Tiap-tiap unit tersembunyi ($Z_j, j=1,2,3,\dots,p$) memperbaiki bias dan bobotnya ($i=0,1,2,\dots,n$)

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (12)$$

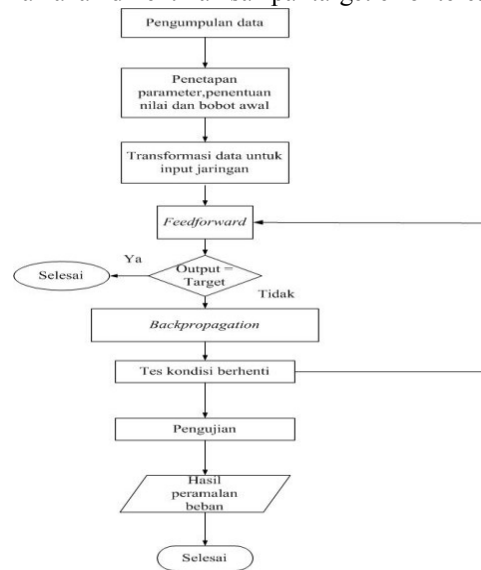
10. Langkah 9 : Uji syarat berhenti

III. METODE PENELITIAN

Penelitian ini mengembangkan model subsistem peramalan beban listrik jangka panjang di kabupaten. Data sekunder yang digunakan adalah data jumlah populasi penduduk, jumlah rumah tangga, jumlah pelanggan listrik tiap – tiap sektor dan penggunaan beban listrik sektor industri yang diperoleh dari PT. PLN APJ dan BPS kabupaten Jember.

A. Peramalan beban listrik

Gambar 4 merupakan tahapan peramalan beban listrik jangka panjang menggunakan JST *backpropagation*. Algoritma ini mengadopsi pendekatan incremental dalam memperbarui bobot, yaitu bobot berubah segera setelah pola pelatihan dilakukan. Proses pelatihan akan dihentikan sampai target error tercapai.



Gambar. 2. Diagram alir peramalan beban listrik menggunakan metode JST *backpropagation*

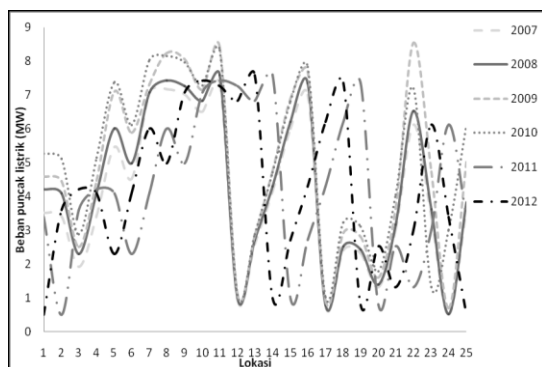
IV. HASIL DAN PEMBAHASAN

A. Pengumpulan data

Data historis diambil 6 tahun ke belakang tiap lokasi yang ada di kabupaten Jember dengan 6 variabel masukan yaitu Populasi Penduduk, Jumlah Rumah Tangga, Pelanggan Sektor Publik, Pelanggan Sektor Bisnis, Pelanggan Sektor Industri, Penggunaan Sektor Industri dan terdapat 1 variabel sebagai target yaitu Beban puncak Listrik. Dengan demikian ada 36 data masukan (6 x 6) sebagai historis untuk pelatihan per

lokasi. Jumlah total data masukan menjadi 36 x 25 kecamatan yaitu 900 data masukan. Data yang dikumpulkan adalah data historis dari tahun 2007 sampai tahun 2012.

Salah satu contoh dari keenam variabel adalah data target. Data target adalah data beban puncak listrik dalam satuan MW. Gambar 3 memperlihatkan data beban puncak listrik se-kabupaten Jember



Gambar. 3. Laju pertumbuhan beban puncak listrik Kabupaten Jember (PT.PLN APJ dan UPT Jember, 2013).

Normalisasi data adalah proses penskalaan data input sesuai dengan *range* fungsi aktivasi yang digunakan.

Dari data master diketahui data minimum=0,496049 dan data maksimum=40532157. *Backpropagation* menggunakan fungsi aktivasi sigmoid biner dengan interval $[0,1]$ di mana *Upperbound*=1 dan *Lowerbound*=0.

Normalisasi data untuk metode *Backpropagation* :

1. Cari nilai α

$$\alpha = \frac{\text{Upperbound} - \text{Lowerbound}}{\max(x) - \min(x)}$$

$$\alpha = \frac{1 - 0}{40532157 - 0,496049}$$

$$\alpha = 2,46718 \cdot 10^{-8}$$

2. Cari nilai β

$$\beta = \text{Upperbound} - (\alpha * (\max(x)))$$

$$\beta = 1 - (2,46718 \cdot 10^{-8} * 40532157)$$

$$\beta = -0,000012710726$$

3. Cari nilai x'

$$x' = \alpha * x + \beta$$

$$x' = 2,46718 \cdot 10^{-8} * 78420 + (-0,000012710726)$$

$$x' = 0,0019220519$$

B. Analisis metode *backpropagation*.

Arsitektur pelatihan JST yang digunakan dalam penelitian ini terdiri dari 3 layer, yaitu *input layer*, *hidden layer* dan *output layer*. Jumlah neuron lapisan masukan sebanyak 6 neuron, penentuan jumlah neuron berdasarkan parameter yang dijadikan faktor yang mempengaruhi peramalan beban listrik. Gambar 4 menunjukkan pola arsitektur jaringan.

Berikut ini algoritma pelatihan metode *Backpropagation* yang terdiri dari tiga tahapan:

1. Inisialisasi bobot (ambil bilangan random yang cukup kecil)

2. Selama kondisi berhenti bernilai salah, kerjakan:

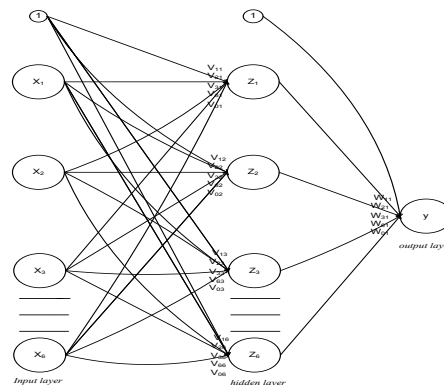
Tahap 1: Perambatan Maju (Feed Forward)

a. Setiap unit input (x_i , $i=1,2,3,\dots,n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan tersembunyi.

b. Setiap unit tersembunyi (z_j , $j=1,2,3,\dots,p$) menjumlahkan bobot sinyal input dengan persamaan (1). Dan menerapkan fungsi aktivasi untuk menghitung

sinyal output-nya dengan persamaan (2.) kemudian mengirimkan sinyal tersebut ke semua unit output.

c. Setiap unit output (y_k , $k=1,2,3,\dots,m$) menjumlahkan bobot sinyal input dengan persamaan (3). Dan menerapkan fungsi aktivasi untuk menghitung sinyal outputnya dengan persamaan (4).



Gambar. 4. Arsitektur pelatihan JST

Tahap 2: Perambatan Balik (*Back Forward*)

a. Setiap unit output (y_k , $k=1,2,3,\dots,m$) menerima pola target yang sesuai dengan pola input pembelajaran, kemudian hitung *error* dengan persamaan (5) kemudian kemudian hitung koreksi bobot dengan persamaan (6).

b. Setiap unit tersembunyi (z_j , $j=1,2,3,\dots,p$) menjumlahkan delta input-nya dengan persamaan (8)

Untuk menghitung informasi *error*, kalikan nilai ini dengan turunan dari fungsi aktifasinya dengan persamaan (9) kemudian hitung koreksi bobot dengan persamaan (10).

Setelah itu, hitung koreksi bias dengan persamaan (10)

Tahap 3: Perubahan Bobot dan Bias

a. Setiap unit output (y_k , $k=1,2,3,\dots,m$) dilakukan perubahan bobot dan bias ($j=0,1,2,\dots,p$) dengan persamaan (11)

Setiap unit tersembunyi (z_j , $j=1,2,3,\dots,p$) dilakukan perubahan bobot dan bias ($i=0,1,2,\dots,n$) dengan persamaan (12)

b. Tes kondisi berhenti.

c. Algoritma Pengujian Metode *Backpropagation*

1. Load bobot dan bias *input layer* (v_{ij}) dan *hidden layer* (w_{ij}) hasil pelatihan.

2. Setiap unit input (x_i , $i=1,2,3,\dots,n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua unit pada lapisan tersembunyi.

3. Setiap unit tersembunyi (z_j , $j=1,2,3,\dots,p$) menjumlahkan bobot sinyal input dengan persamaan (1) dan menerapkan fungsi aktivasi untuk menghitung sinyal output-nya dengan persamaan (2).

4. Setiap unit output (y_k , $k=1,2,3,\dots,m$) menjumlahkan bobot sinyal input dengan persamaan (3) dan menerapkan fungsi aktivasi untuk menghitung sinyal output-nya dengan persamaan (4).

Pada data epoch kedua, juga dilakukan langkah-langkah yang sama dengan menggunakan bobot baru atau nilai bobot akhir hasil pengolahan data pada epoch pertama. Proses ini dilakukan secara berulang-ulang sampai pada epoch yang ditentukan (1000) atau target error yang ditentukan (0,01).

Setelah proses pelatihan selesai, *Backpropagation* dapat dipakai untuk pengujian. Berikut ini pengujian algoritma metode *Backpropagation*:

1. Operasi pada Hidden Layer dengan persamaan (1)

$$z_{in1} = 0.1 + (0.01 * 0.0019220519) + (0.01 * 0.0002504634) + (0.01 * 0.0000122913) + (0.01 * 0.0000048651) + (0.01 * 0.0000125873) + (0.01 * 0.3905400823)$$

$$z_{in1} = 0.103926829$$

2. Fungsi aktifasi pada hidden layer dengan persamaan (2)

$$Z_j = f(z_{inj})$$

$$Z_1 = \frac{1}{1 + e^{-0.103926289}} = 0.525958415$$

3. Operasi pada output layer dengan persamaan (3)

$$y_{ink} = w_0 + w_1 * z_1 + w_2 * z_2 * w_3 * z_3$$

$$y_{ink} = 0.3 + (0.2 * 0.5347602100) + (0.5 * 0.5498896453) + (0.1 * 0.5639560100) = -1.00000020997$$

Berdasarkan pola data yang diujikan (pola data pada tahun 2008 di lokasi Rambipuji) dapat diketahui nilai min=4.08198 dan max=22084813. Hasil pengujian berupa data yang diskalakan dengan notasi x' atau $y_{ink} = -1.00000020997$. Sedangkan berdasarkan fungsi identitas pada pengujian yang memiliki interval [-1,1] dapat diketahui $Upperbound = 1$ dan $Lowerbound = -1$. Sehingga perhitungan denormalisasinya adalah sebagai berikut:

1. Cari nilai α

$$\alpha = \frac{Upperbound - Lowerbound}{max(x) - min(x)}$$

$$\alpha = \frac{1 - (-1)}{22084813 - 4.08198}$$

$$\alpha = 9.056.10^{-8}$$

2. Cari nilai β

$$\beta = Upperbound - (\alpha * max(x))$$

$$\beta = 1 - (9.056.10^{-8} * 22084813)$$

$$\beta = -1.0000006528$$

3. Cari nilai x

$$x' = \alpha * x + \beta$$

$$x = \frac{x' - \beta}{\alpha}$$

$$x = \frac{-1.00000020997 - (-1.0000006528)}{9.056.10^{-8}}$$

$$x = 4.889880$$

Dari hasil denormalisasi dapat diketahui bahwa data hasil prediksi beban puncak adalah 4.889880 MW

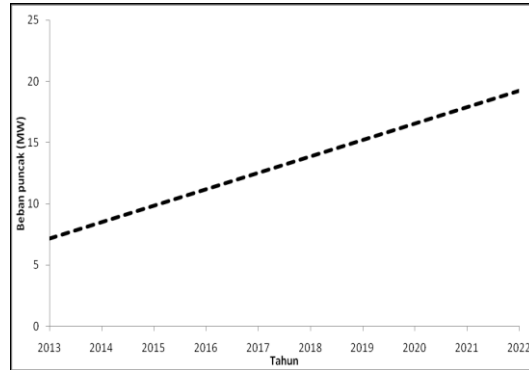
C. Peramalan beban listrik.

Setelah melakukan pelatihan dilanjutkan dengan tahap berikutnya yaitu prakiraan atau prediksi, dalam tahap ini dilakukan prakiraan untuk 10 tahun mendatang yaitu sampai dengan tahun 2022. Hasil prediksi pada gambar 6 dibawah diambil dari data prediksi lokasi Glantangan

Pada fitur antarmuka pengguna diberikan beberapa menu atau fitur-fitur untuk dapat meneruskan proses

sesuai dengan kebutuhan pengguna, yaitu fitur peramalan beban listrik dan fitur beban puncak maksimum

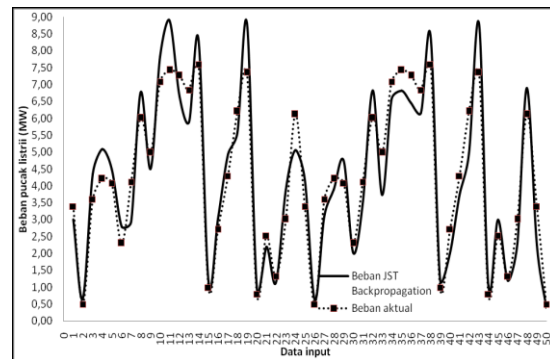
Pada tahap ini sistem jaringan saraf tiruan akan diperkenalkan berdasarkan pola-pola yang telah ditentukan. Proses perhitungannya tidak akan ditampilkan dalam program simulasi. Proses perhitungan tersebut akan bekerja dalam bentuk mode background, sehingga yang akan terlihat dalam tampilan form hanya jumlah iterasi yang telah dilakukan dan hasil dari pelatihan. Proses pembelajaran pada tahap training ini merupakan proses pembelajaran supervised. Artinya, program akan dituntun untuk mengenal pola-pola berdasarkan database



Gambar.5. Grafik prediksi beban puncak 10 tahun ke depan

D. Pengujian metode *backpropagation*

Data selama 2 tahun terakhir juga digunakan sebagai data untuk menguji validitas data antara data yang dihasilkan oleh PT.PLN APJ-UPT Jember dan data hasil perhitungan seperti yang ditunjukkan pada gambar 9 dibawah sehingga dapat diketahui nilai error yang dihasilkan.



Gambar. 6. Perbandingan beban puncak aktual dan JST backpropagation

Perhitungan akurasi Backpropagation terhadap target dihitung dengan metode Mean Absolute Percentage Error (MAPE) :

$$MAPE = \frac{100 \%}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right|$$

$$= \frac{100 \%}{50} \sum_{t=1}^{50} |854,74|$$

$$= 17.09 \%$$

