

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pada bab ini akan dijelaskan mengenai teori dasar dan literatur yang menjadi dasar dalam penyelesaian masalah penelitian ini. Berbagai sumber yang digunakan baik berupa buku, artikel, jurnal, maupun media internet yang digunakan untuk mendukung teori penyelesaian tugas akhir ini. Adapun pembahasan teori mencakup berbagai jenis data, teori dan metode perkiraan.

Banyak metode perkiraan yang dapat digunakan, oleh karena itu diperlukan penyesuaian terhadap jenis data yang digunakan dan jenis perkiraan yang akan dilakukan. Penelitian mengenai perkiraan kebutuhan energi listrik dimasa depan sudah pernah dibuat dengan studi kasus beberapa kota di Indonesia. Berikut akan dipaparkan beberapa penelitian yang berkaitan dan dijadikan sebagai sumber referensi dalam tugas akhir ini:

Anifah (2016) penelitian yang dilakukan mengenai perkiraan beban listrik jangka panjang di Provinsi D.I Yogyakarta menggunakan *neural network backpropagation*. Didapatkan bahwa perkiraan beban listrik Provinsi D.I Yogyakarta menghasilkan rata-rata kelajuan data sebesar 8,1007% setiap tahunnya. Komparasi hasil perkiraan beban listrik menggunakan *neural network backpropagation* dengan hasil perkiraan beban listrik RUPTL PT.PLN 2015 – 2024 menghasilkan rata-rata persentase perbedaan data sebesar 9,882%.

Madeli, Syukran (2018) memprediksi beban Puncak di kota Jambi menggunakan perhitungan manual menghasilkan persentase total kenaikan tiap tahunnya adalah sekitar 4.94%. Sedangkan perhitungan menggunakan metode Jaringan Syaraf Tiruan (JST) menghasilkan MSE sebesar 8.9481e-10, hasil ini menunjukkan model jaringan yang dibangun cukup bagus untuk diterapkan pada peramalahn beban puncak. Hasil prediksi yang dilakukan untuk kurun waktu 9 tahun kedepan dari tahun 2017 – 2025 menunjukkan adanya tren kenaikan beban

puncak setiap tahunnya. Pada tahun 2025, prediksi beban puncak yang dihasilkan adalah 855.8628 MW.

Syahfitra, Febrian D (2018) melakukan penelitian memprediksi beban puncak di gardu induk bumiayu dan menunjukkan beban puncak transformator1 menghasilkan MSE pelatihan sebesar $5,7446e-08$ dan MSE pengujian sebesar $5,3671e-04$. Hasil ini menunjukkan model jaringan yang dibangun cukup bagus untuk diterapkan pada perkiraan beban puncak transformator1. Hasil prediksi yang telah dilakukan untuk kurun waktu dari tahun 2018 sampai tahun 2027 menunjukkan adanya tren kenaikan beban puncak setiap tahunnya. Pada tahun 2027, prediksi beban puncak transformato yang dihasilkan adalah 33,39 MW. Angka tersebut merupakan 55,65 % dari kapasitas transformator1, dan dapat dikategorikan sebagai standar ringan beban transformator.

2.2 Dasar Teori

2.2.1 Gardu Induk

Gardu Induk merupakan suatu instalasi yang terdiri dari sekumpulan peralatan listrik yang disusun menurut pola tertentu dengan pertimbangan teknis, ekonomis serta keindahan.

Fungsi dari Gardu Induk adalah sebagai berikut :

- a. Mentransformasikan tenaga listrik tegangan tinggi yang satu ketegangan yang lainnya atau tegangan menengah.
- b. Pengukuran pengawasan operasi serta pengaturan pengamanan dari system tenaga listrik.
- c. Pengaturan daya ke gardu-gardu lainnya melalui tegangan tinggi dan gardu distribusi melalui feeder tegangan menengah.

Pada dasarnya gardu induk terdiri dari saluran masuk dan dilengkapi dengan transformator daya, perlatan ukur, peralatan penghubung dan lainnya yang saling menunjang

2.2.1.1 Jenis Gardu Induk Menurut Tegangan

Berdasarkan tegangan, gardu induk dapat dibedakan menjadi 2 macam yaitu:

a. Gardu induk transmisi

Yaitu gardu induk yang mendapat daya dari saluran transmisi untuk kemudian menyalurkannya ke daerah beban (industri, kota, dan sebagainya). Gardu induk transmisi yang ada di PLN adalah tegangan tinggi 150 KV dan tegangan tinggi 70 KV.

b. Gardu distribusi

Yaitu gardu induk yang menerima tenaga dari gardu induk transmisi dengan menurunkan tegangannya melalui transformator tenaga menjadi tegangan menengah (20 KV, 12 KV atau 6 KV) untuk kemudian tegangan tersebut diturunkan kembali menjadi tegangan rendah (127/220 V) atau (220/380 V) sesuai dengan kebutuhan.

2.2.1.2 Jenis Gardu Induk Menurut Fungsinya

Berdasarkan fungsinya, gardu induk dapat dibedakan menjadi 5 macam yaitu :

a. Gardu Induk Penaik Tegangan

Merupakan gardu induk yang berfungsi untuk menaikkan tegangan, yaitu tegangan pembangkit (generator) dinaikkan menjadi tegangan sistem. Gardu Induk ini berada di lokasi pembangkit tenaga listrik. Karena output voltage yang dihasilkan pembangkit listrik kecil dan harus disalurkan pada jarak yang jauh, maka dengan pertimbangan efisiensi, tegangannya dinaikkan menjadi tegangan ekstra tinggi atau tegangan tinggi.

b. Gardu Induk Penurun Tegangan

Merupakan gardu induk yang berfungsi untuk menurunkan tegangan, dari tegangan tinggi menjadi tegangan tinggi yang lebih rendah dan menengah atau

tegangan distribusi. Gardu Induk terletak di daerah pusat pusat beban, karena di gardu induk inilah pelanggan (beban) dilayani.

c. Gardu Induk Pengatur Tegangan

Pada umumnya gardu induk jenis ini terletak jauh dari pembangkit tenaga listrik. Karena listrik disalurkan sangat jauh, maka terjadi tegangan jatuh (*voltage drop*) transmisi yang cukup besar. Oleh karena diperlukan alat penaik tegangan, seperti *bank capacitor*, sehingga tegangan kembali dalam keadaan normal.

d. Gardu Induk Pengatur Beban

Berfungsi untuk mengatur beban. Pada gardu induk ini terpasang beban motor, yang pada saat tertentu menjadi pembangkit tenaga listrik, motor berubah menjadi generator dan suatu saat generator menjadi motor atau menjadi beban, dengan generator berubah menjadi motor yang memompakan air kembali ke kolam utama.

e. Gardu Distribusi

Gardu induk yang menyalurkan tenaga listrik dari tegangan sistem ke tegangan distribusi. Gardu induk ini terletak di dekat pusat-pusat beban

2.2.2 Metode Perkiraan

Dalam memperkirakan beban energi listrik dapat digunakan metode sebagai berikut :

a. Metode Analisis

Metode ini dibangun berdasarkan data dari analisis penggunaan terakhir tenaga listrik pada setiap konsumen pemakai. Perolehan data merupakan hasil survei ke lapangan. Pada umumnya data diperlukan ialah data yang memberi gambaran penggunaan peralatan listrik di masyarakat atau kemampuan masyarakat membeli peralatan listrik. Keuntungan metode ialah hasil perkiraan merupakan hasil simulasi dari penggunaan tenaga listrik di masyarakat, sederhana dan mengurangi masalah validitas

parameter model. Dan sebaliknya metode ini tidak tanggap terhadap perubahan parameter ekonomi, sebagai contoh pengaruh kenaikan tarif listrik, pendapatan (PDRB), dan sebagainya.

b. Metode Ekonometri

Suatu metode yang dibangun dengan mengikuti indikator-indikator ekonomi. Prakiraan beban ini didasarkan adanya hubungan antara penjualan energi listrik dan beban puncak dengan beberapa variabel ekonomi seperti pendapatan (PDRB), harga dan penggunaan peralatan listrik. Metode ekonometri ini cocok digunakan untuk suatu kasus, misalnya hanya berlaku untuk suatu daerah atau wilayah.

c. Metode Kecendrungan

Metode ini disebut juga metode trend yaitu metode yang dibuat berdasarkan kecendrungan hubungan data masa lalu tanpa memperhatikan penyebab atau hal-hal yang mempengaruhinya (pengaruh ekonomi, iklim, teknologi, dan lain-lain). Dari data masa lalu tersebut diformulasikan sebagai fungsi dari waktu dengan persamaan matematik oleh karena itu metode ini disebut metode time series. Metode ini biasanya digunakan pada prakiraan jangka pendek.

d. Metode Gabungan

Metode ini merupakan gabungan dari metode analisis, ekonometri, dan metode kecendrungan dimana masing-masing memiliki keunggulan dan kelemahan sendiri-sendiri. Metode ini dikembangkan berdasarkan keadaan sosioekonomi, penggunaan terakhir tenaga listrik disuatu daerah atau wilayah.

2.2.3 Prakiraan

Pada dasarnya ramalan merupakan suatu dugaan atau perkiraan atas terjadinya kejadian di waktu mendatang. Ramalan bisa bersifat kualitatif maupun kuantitatif. Ramalan kualitatif tidak berbentuk angka, misalnya besok akan turun hujan, tahun depan akan terjadi perang, hasil penjualan tahun depan akan meningkat, dan sebagainya. Sedangkan ramalan kuantitatif dinyatakan dalam

bentuk angka atau bilangan. Ramalan kuantitatif sendiri dibagi menjadi dua jenis, yaitu:

1. Perkiraan Tunggal (*point forecast*)
2. Perkiraan Selang (*interval forecast*)

Ramalan tunggal terdiri dari satu nilai saja, misalnya hasil produksi perusahaan ABC akan mencapai 1000 satuan, keuntungan penjualan bulan depan akan bernilai Rp. 250.000,-. Besar pemakaian daya tahun depan akan naik 5% dan sebagainya.

Ramalan selang terdiri atas beberapa nilai dalam satu interval yang dibatasi nilai batas bawah (ramalan rendah) dan batas atas (ramalan tinggi). Misalnya, hasil produksi perusahaan ABC akan mencapai 800 – 1200 satuan, keuntungan penjualan bulan depan akan bernilai Rp. 200.000 sampai dengan Rp.250.000, besarnya kenaikan konsumsi daya tahun depan berkisar antara 5 – 10%.

Menurut jangka waktunya, perkiraan dibagi menjadi 3 periode, sesuai dengan materi yang di ramalkan. Dalam perkiraan beban listrik, periode perkiraan dibagi menjadi 3, yaitu:

1. Perkiraan Jangka Panjang (*Long-Term Forecasting*)

Merupakan perkiraan yang memperkirakan keadaan dalam waktu beberapa tahun ke depan. Tujuannya dalam adalah untuk dapat mempersiapkan ketersediaan unit pembangkitan, sistem transmisi, serta distribusi.

2. Peramlaan Jangka Menengah (*Mid-Term Forecasting*)

Merupakan perkiraan dalam jangka waktu bulanan atau mingguan. Tujuannya untuk mempersiapkan jadwal persiapan dan operasional sisi pembangkit.

3. Perkiraan Jangka Pendek (*Short-Term Forecasting*)

Merupakan perkiraan dalam jangka waktu harian hingga setiap jam. Biasa digunakan untuk studi perbandingan beban listrik perkiraan dengan actual (*realtime*).

2.2.4 Cara-cara Perkiraan Beban

Salah satu faktor yang sangat menentukan dalam membuat rencana operasi sistem tenaga listrik adalah perkiraan beban yang akan dialami oleh sistem tenaga listrik yang bersangkutan. Tidak ada rumus eksak untuk ini karena besarnya beban ditentukan oleh pemakai (konsumen) tenaga listrik yang secara bebas dapat menentukan pemakaiannya. Namun karena pada umumnya kebutuhan tenaga listrik seorang konsumen sifatnya periodik maka grafik pemakaian tenaga listrik atau yang disebut dengan grafik beban dari sistem tenaga listrik yang mempunyai sifat periodik. Oleh karena itu statistik beban dari masa lalu beserta analisisnya seperti di uraikan sangat diperlukan untuk memperkirakan beban di masa yang akan datang yang pada umumnya dilakukan dengan cara mengekstrapolasi grafik beban di masa lampau ke masa yang akan datang. Setelah dilakukan ekstrapolasi kemudian ditambahkan koreksi-koreksi terhadap hal-hal khusus, baik untuk perkiraan jangka panjang, jangka menengah, maupun jangka pendek.

Grafik beban perlahan-lahan berubah bentuknya baik kuantitatif maupun kualitatif. Perubahan ini disebabkan oleh:

1. Bertambahnya konsumen tenaga listrik
2. Bertambahnya konsumen tenaga listrik dari konsumen lama, misalnya karena ia membeli peralatan listrik dari konsumen lama, atau juga membeli peralatan listrik tambahan.
3. Suhu udara, kalau suhu udara tinggi maka pemakaian alat-alat penyejuk udara bertambah dan ini menambah pemakaian tenaga listrik.
4. Kegiatan ekonomi dalam masyarakat.
5. Kegiatan social dalam masyarakat.

Dari uraian di atas dapat dimengerti bahwa tidaklah mungkin ditemukan rumus yang eksak untuk menentukan besarnya beban, tetapi beban dapat diperkirakan besarnya berdasarkan pengalaman-pengalaman dan pengamatan di masa lalu kemudian di adakan perkiraan untuk masa yang akan datang.

2.2.5 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) merupakan sistem kecerdasan tiruan dengan kemampuan belajar dan menghimpun pengetahuan hasil pembelajaran dalam jaringan selnya (*neuron*) sehingga memungkinkan jaringan secara keseluruhan semakin cerdas merespon masukan/input yang diberikan. Kemampuan belajar dan mengakumulasi pengetahuan ini memungkinkan sistem jaringan syaraf tiruan untuk dapat beradaptasi dengan lingkungan yang memberikan input kepadanya. Layaknya otak manusia dalam merespon kondisi lingkungan berbeda-beda, peranan JST dalam bidang penelitian dan pengembangan sangat penting di masa yang akan datang yang menuntut aspek otomatisasi dan aspek interaktif antara alat dan manusia. (Muis 2017)

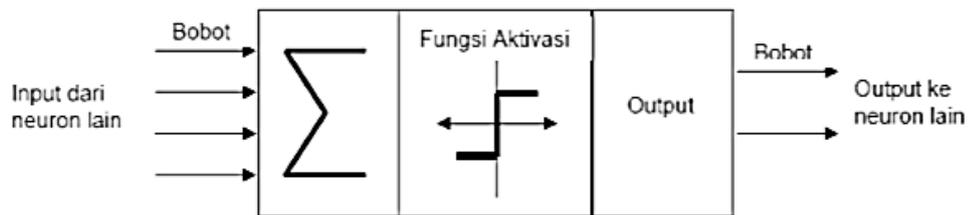
Menurut Jong Jek Siang (2009), sistem jaringan syaraf tiruan ditentukan oleh 3 hal, yaitu:

1. Pola hubungan antar *neuron* atau biasa disebut arsitektur jaringan.
2. Metode untuk menentukan bobot penghubung (disebut metode *training/leraning/algorithm*).
3. Fungsi aktivasi.

2.2.6 Model Jaringan Syaraf Tiruan

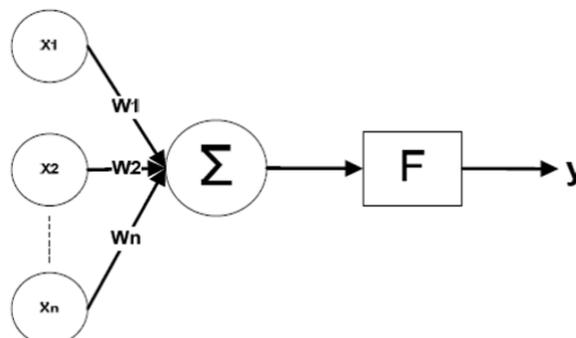
Seperti halnya sel pada otak manusia, jaringan syaraf juga terdiri dari beberapa *neuron*, dan terdapat hubungan antara *neuron-neuron* tersebut. Pada gambar dibawah. menunjukkan struktur neuron yang mana Neuron-neuron akan menjadi informasi yang diterima melalui sambungan keluarannya menuju ke *neuron-neuron* yang lain. Pada jaringan syaraf Tiruan hubungan ini dikenal dengan nama bobot. Informasi tersebut tersimpan pada suatu nilai tertentu pada bobot tersebut. keduanya atau mungkin lebih untuk mendapatkan redundansi data.

Ini diproses oleh suatu fungsi perambatan yang akan menjumlahkan nilai-nilai semua bobot yang akan datang. Hasil penjumlahan ini kemudian dibandingkan dengan suatu Informasi yang disebut dengan masukkan dikirim ke neuron dengan bobot kedatangan tertentu. Masukkan nilai ambang (*threshold*) tertentu melalui fungsi aktivasi setiap *neuron*.



Gambar 2.1 Struktur *Neuron JST*

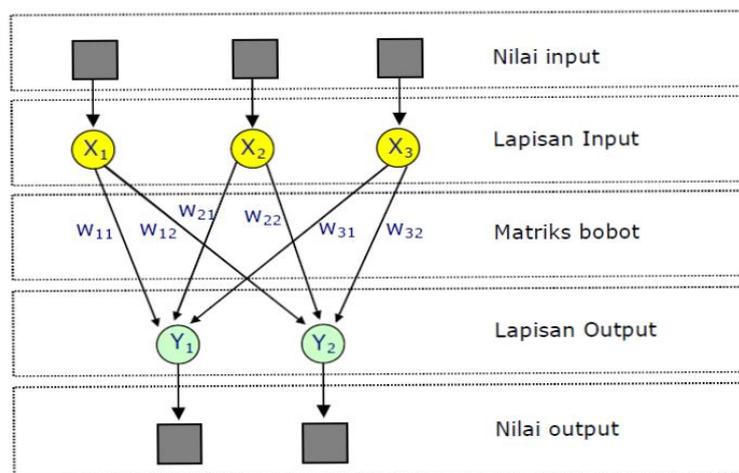
Pada jaringan sel syaraf, neuron-neuron akan dikumpulkan dalam lapisan – lapisan yang disebut dengan lapisan *neuron*. Biasanya neuron pada satu lapisan akan dihubungkan dengan lapisan sebelum atau sesudahnya terkecuali lapisan masukan dan lapisan keluaran. Informasi yang diberikan pada jaringan syaraf akan dirambatkan dari lapisan ke lapisan, melalui dari lapisan masukan sampai lapisan keluaran melalui lapisan tersembunyi. Algoritma pembelajaran menentukan informasi akan dirambatkan kearah mana, pada gambar dibawah ditunjukkan *neuron* jaringan syaraf dengan fungsi aktivasi F. sebuah neuron akan mengolah N masukan ($x_1, x_2, x_3, \dots, x_n$) yang masing-masing memiliki bobot $w_1, w_2, w_3, \dots, w_n$ dengan rumus : $y_{in} = \sum x_i w_i$



Gambar 2.2 Model Neuron Sederhana

2.2.7 Arsitektur Jaringan

Dalam JST, *neuron-neuron* akan dikumpulkan menjadi lapisan (*layer*) yang disebut *neuron layer*. Masing-masing *layer* akan dihubungkan satu sama lain, baik dengan *layer* sebelumnya maupun sesudahnya. Informasi akan dirambatkan dari satu *layer* ke *layer* berikutnya, mulai dari *input* sampai ke *output layer* melalui *hidden layer*. (Haidaroh, 2013)

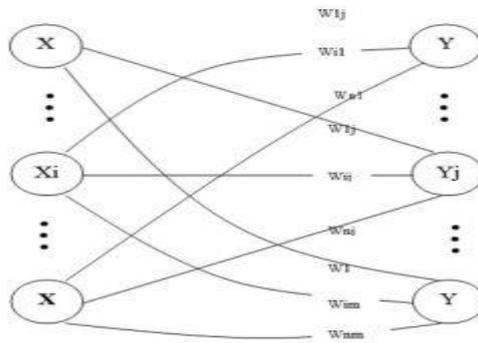


Gambar 2.3 Arsitektur Jaringan Syaraf Tiruan

Menurut Jong Jek Siang (2009), arsitektur jaringan syaraf tiruan yang sering digunakan antara lain :

a. Jaringan layar tunggal (*single layer network*)

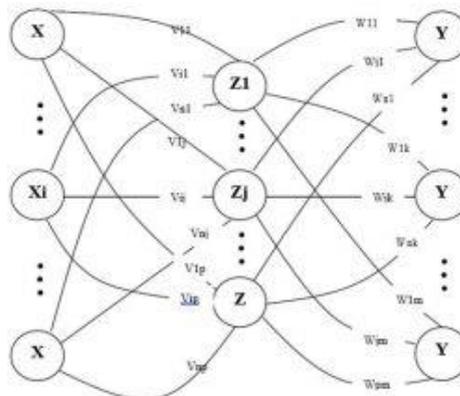
Dalam jaringan ini, sekumpulan *input neuron* dihubungkan langsung dengan sekumpulan outputnya, meskipun dengan bobot yang berbeda-beda. Tidak ada unit input yang dihubungkan dengan unit input lainnya. Demikian pula dengan *unit output*



Gambar 2.4 Arsitektur Jaringan Layar Tunggal

b. Jaringan layar jamak (*multi layer network*)

Dalam jaringan ini, selain unit input dan output terdapat juga layer yang tersembunyi. Jumlah layer tersembunyi dapat lebih dari satu. Samaseperti pada unit input dan output, unit-unit dalam satu layer tidak saling berhubung.

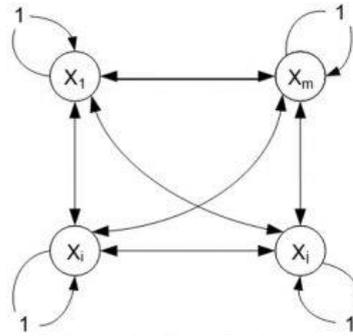


Gambar 2.5 Arsitektur Jaringan Layar Jamak

Jaringan ini didesain agar dapat menyelesaikan masalah yang lebih kompleks dengan layar tunggal, meskipun membutuhkan waktu yang lebih lama dalam proses pelatihnannya.

c. Jaringan Recurrent

Model jaringan ini mirip dengan jaringan layar tunggal maupun layar ganda, hanya saja ada neuron output yang memberikan sinyal pada unit input yang disebut feedback loop.



Gambar 2.6 Arsitektur Jaringan Recurrent

2.2.8 Fungsi Aktivasi

Menurut Jong Jek Siang (2009) fungsi aktivasi ialah hanya dipakai untuk menentukan keluaran suatu neuron. Berikut adalah fungsi aktivasi yang sering dipakai:

a. Fungsi *threshold* (batas ambang)

$$f(x) = \begin{cases} 1, & \text{Jika } x \geq a \\ 0, & \text{Jika } x \leq a \end{cases}$$

Untuk beberapa kasus, fungsi *threshold* yang dibuat tidak bernilai 0 atau 1, tapi bernilai 1 atau -1 (disebut dengan *threshold bipolar*)

$$f(x) = \begin{cases} 1, & \text{Jika } x \geq a \\ -1, & \text{Jika } x \leq a \end{cases}$$

b. Fungsi sigmoid

$$f(x) = \frac{1}{1+e^{-x}}$$

Fungsi sigmoid sering dipakai karena nilai fungsinya yang terletak antara 0 dan 1 serta dapat diturunkan dengan mudah.

c. Fungsi identitas

$$f(x) = x$$

Fungsi identitas sering dipakai apabila diinginkan keluaran jaringan berupa bilangan riil (bukan hanya pada range [0, 1] atau [1,-1])

2.2.9 Algoritma *Backpropagation*

Backpropagation merupakan algoritma pembelajaran yang terawasi dan biasanya digunakan oleh perceptron dengan banyak lapisan untuk mengubah

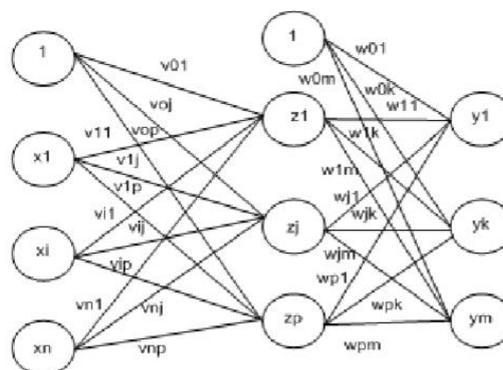
bobot-bobot yang terhubung dengan *neuron-neuron* yang ada pada lapisan tersembunyi. Algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobotnya ke arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. (Kusumadewi 2004)

Secara garis besar, mengapa algoritma ini disebut sebagai propagasi balik, karena ketika JST diberikan pola masukan sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit lapisan keluaran. Kemudian unit-unit lapis keluaran memberikan tanggapan yang disebut sebagai keluaran JST. Saat keluaran JST tidak sama dengan keluaran yang diharapkan maka keluaran akan disebarkan mundur (*backward*) pada lapis tersembunyi diteruskan ke unit pada lapisan masukan. Oleh karenanya maka mekanisme pelatihan tersebut dinamakan *backpropagation* (propagasi balik). (Purnomo dan Kurniawan 2006)

Menurut Jong Jek Siang (2009) standar algoritma *backpropagation* adalah :

a. Arsitektur *backpropagation*

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layer tersembunyi (*hidden layer*). Gambar 2.5 adalah arsitektur *backpropagation* dengan n buah masukan (ditambah sebuah bias), sebuah layer tersembunyi yang terdiri dari p unit (ditambah sebuah bias) serta m unit keluaran.



Gambar 2.7 Arsitektur *backpropagation*

V_{ij} merupakan bobot garis dari unit masukan x_i ke unit layer tersembunyi z_j (v_{0j} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit

layer tersembunyi z_j). W_{jk} merupakan bobot dari unit layer tersembunyi z_j ke unit keluaran y_k (W_{0k} merupakan bobot dari bias di layer tersembunyi z_j ke unit keluaran y_k).

b. Fungsi aktivasi

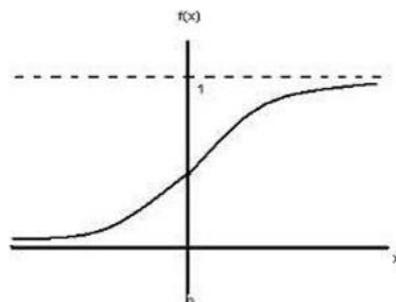
Fungsi aktivasi yang dipakai harus memenuhi beberapa syarat diantaranya: kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga yang sering dipakai ialah fungsi sigmoid biner yang memiliki range (0, 1).

$$f(x) = \frac{1}{1+e^{-x}}$$

Dengan turunan

$$f'(x) = f(x)(1 - f(x))$$

Berikut adalah gambar grafik fungsi sigmoid biner.



Gambar 2.8 Grafik fungsi sigmoid biner

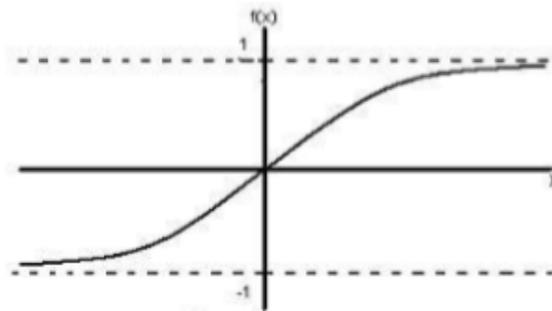
Fungsi lain yang sering dipakai adalah fungsi sigmoid bipolar yang bentuk dan fungsinya mirip dengan fungsi sigmoid biner, tapi dengan range (-1, 1).

$$f(x) = \frac{2}{1+e^{-x}} - 1$$

Dengan turunan

$$f'(x) = \frac{(1 + f(x))(1 - f(x))}{2}$$

Berikut grafik fungsi dari sigmoid bipolar.



Gambar 2.9 Grafik fungsi sigmoid bipolar

Fungsi sigmoid memiliki nilai maksimum =1, maka untuk pola yang targetnya >1 pola masukan dan keluaran harus terlebih dahulu di transformasikan sehingga semua polanya memiliki range yang sama seperti fungsi sigmoid yang dipakai. Alternatif lain yang bisa dipakai adalah menggunakan fungsi aktivasi sigmoid hanya pada layar yang bukan layar keluaran. Pada layar keluaran, fungsi aktivasi yang dipakai adalah fungsi identitas $f(x) = x$.

c. Pelatihan standar *backpropagation*

Pelatihan *backpropagation* meliputi 3 fase yaitu fase pertama adalah fase maju. Pola masukan dihitung maju dimulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan. Fase yang kedua ialah fase mundur, selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit di layar keluaran. Fase yang ketiga yaitu modifikasi bobot untuk menurunkan kesalahan yang terjadi.

Algoritma pelatihan untuk jaringan dengan satu layar tersembunyi (dengan fungsi aktivasi sigmoid biner) adalah sebagai berikut:

Langkah 0 : Inisialisasi semua bobot dengan bilangan acak kecil

Langkah 1 : Jika kondisi penghentian belum terpenuhi, lakukan langkah 2-9

Langkah 2 : Untuk setiap pasang data pelatihan, lakukan langkah 3-8

Fase I : Propagasi maju

Langkah 3 : Tiap unit masukan menerima sinyal dan meneruskannya ke unit tersembunyi di atasnya

Langkah 4 : Hitung semua keluaran di unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$Z_{netj} = v_j0 + \sum_{i=1}^n x_i v_{jt}$$

$$z_j = (z_{netj}) = \frac{1}{1 + e^{-Z_{netj}}}$$

Langkah 5 : Hitung keluaran jaringan di unit y_k ($k = 1, 2, \dots, m$)

$$y_{netj} = w_{k0} + \sum_{i=1}^n z_j w_{kj}$$

$$y_k = (y_{netk}) = \frac{1}{1 + e^{-Y_{netj}}}$$

Fase II : Propagasi mundur

Langkah 6 : Hitung faktor δ unit keluaran berdasarkan kesalahan di setiap unit keluaran y_k ($k = 1, 2, \dots, m$)

$$\delta_k = (t_k - y_k)'(y_{netk}) = (t_k - y_k)y_k(1 - y_k)$$

δ_k merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar dibawahnya. Hitung suku perubahan bobot w_{kj} (yang akan dipakai nanti untuk merubah bobot w_{kj}) dengan laju percepatan α

$$\Delta w_{kj} = \alpha \delta_k \quad ; = 1, 2, \dots, m; j = 0, 1, \dots, p$$

Langkah 7 : hitung faktor δ unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi z_j ($j = 1, 2, \dots, p$)

$$\delta_{netj} = \sum_{k=1}^m \delta_k w_{kj}$$

Faktor δ unit tersembunyi :

$$\delta_j = \delta_{netj} f' (Z_{netj}) = \delta_{netj} Z(1 - Z_j)$$

Hitung suku perubahan bobot V_{ji} (yang akan dipakai nanti untuk merubah bobot V_{ji})

$$\Delta v_{ji} = \alpha \delta_j \quad ; = 1,2,\dots,p; i = 0,1,\dots,n$$

Fase III : Perubahan Bobot

Langkah 8 : Hitung semua perubahan bobot. Perubahan bobot garis yang menuju ke unit keluaran:

$$w_{kj}(\text{baru}) = (\text{lama}) + \Delta \quad (k = 1,2,\dots,m; j = 0,1,\dots,p)$$

Perubahan bobot garis yang menuju ke unit tersembunyi:

$$v_{ji}(\text{baru}) = (\text{lama}) + \Delta \quad (j = 1,2,\dots,p; i = 0,1,\dots,n)$$

Setelah selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) yang dipakai untuk menentukan keluaran jaringan. Apabila fungsi aktivasi yang dipakai bukan sigmoid biner, maka langkah 4 dan 5 harus disesuaikan. Demikian juga turunannya pada langkah 6 dan 7.

Pada pelatihan jaringan syaraf tiruan yang terdapat pada aplikasi Matlab terdapat berbagai jenis Jaringan Syaraf Tiruan *Backpropagation* yaitu:

1. *Elman Backpropagation*

Jaringan *Elman* adalah jaringan berulang sederhana (SRN). Itu hanya jaringan umpan-maju dengan unit tambahan yang disebut *neuron konteks*. *Neuron konteks* menerima *input* dari *neuron* lapisan tersembunyi. Jaringan dilatih menggunakan algoritma *backpropagation*, dan tujuan pelatihan adalah untuk mempelajari fungsi sinus

2. *Feed-Forward Backpropagation*

Feed-Forward Neural Network adalah jenis arsitektur Jaringan *Neural* di mana koneksi tidak membentuk siklus (seperti dalam jaring berulang). Istilah "Umpan maju" juga digunakan ketika Anda memasukkan sesuatu di lapisan input dan bergerak dari *input* ke *hidden* dan dari *hidden* ke layer *output*.

3. *Cascade-forward Backpropagation*

Cascade-forward neural network adalah salah satu kelas dari *neural network* yang mirip dengan *feed-forward networks*, tetapi termasuk koneksi dari input dan

setiap lapisan sebelumnya ke lapisan berikut. Seperti halnya jaringan *feed-forward*, jaringan *Cascade-forward* dapat mempelajari hubungan input-output yang terbatas dengan cukup memberikan neuron tersembunyi. Keuntungan dari metode ini adalah mengakomodasi hubungan nonlinear antara input dan output dengan tidak menghilangkan hubungan linear antara keduanya.

2.2.10. Algoritma *Training* Jaringan pada MATLAB

MATLAB menyediakan fitur neural network yang memudahkan penggunaannya dalam membangun suatu jaringan syaraf tiruan. Dalam membangun suatu jaringan syaraf tiruan, perlu dilakukan tahap penelitian (*training*). Terdapat beberapa algoritma training jaringan yang terdapat pada MATLAB, antara lain :

1) Trainlm (Levenberg-Marquardt)

Trainlm adalah fungsi pelatihan jaringan yang memperbarui nilai bobot dan bias sesuai dengan pengoptimalan Levenberg-Marquardt. Trainlm biasanya merupakan algoritma *backpropagation* tercepat di *toolbox* untuk melatih jaringan syaraf tiruan berukuran sedang, namun algoritma ini memerlukan banyak penyimpanan daripada algoritma lainnya.

2) Trainbfgf (BFGS Quasi-Newton)

Trainbfgf adalah fungsi pelatihan jaringan yang memperbarui nilai bobot dan bias dengan metode Quasi-Newton BFGS. Algoritma ini membutuhkan lebih banyak perhitungan pada setiap literasi dan memerlukan penyimpanan lebih besar daripada metode gradien konjugasi. Walaupun biasanya konvergensi dalam iterasi lebih sedikit.

3) Trainrp (*Resilient Backpropagation*)

Trainrp adalah fungsi pelatihan jaringan yang memperbarui nilai bobot dan bias dengan metode *Resilient Backpropagation*. Tujuan dari algoritma pelatihan tangguh *Backpropagation* adalah untuk menghilangkan efek berbahaya dari besaran turunan parsial.

4) *Trainscg (Scaled Conjugate Gradient)*

Trainscg adalah fungsi pelatihan jaringan yang memperbarui nilai bobot dan bias dengan metode gradien konjugasi parsial.

5) *Traincgb (Conjugate Gradient with Powell/Beale Restarts)*

Traincgb adalah fungsi pelatihan jaringan yang memperbarui nilai bobot dan bias sesuai dengan *Backpropagation* gradien konjugasi dengan perulangan Powell-Belae

6) *Traincgf (Fletcher-Powell Conjugate Gradient)*

Traincgf fungsi pelatihan jaringan yang memperbarui nilai bobot dan bias sesuai dengan konvergensi *gradien backpropagation* dengan update Fletcher-Reeves. Algoritma gradien konjugasi biasanya jauh lebih cepat daripada variabel *Learning rate backpropagation*, dan kadang lebih cepat daripada *trainrp*, walaupun hasilnya bervariasi untuk beberapa kondisi. Algoritma gradien konjugasi hanya memerlukan sedikit penyimpanan daripada algoritma yang lebih sederhana. Oleh karena itu, algoritma ini bagus untuk jaringan dengan sejumlah besar bobot.

7) *Traincgp (Polak-Ribere Conjugate Gradient)*

Traincgp adalah fungsi pelatihan jaringan yang memperbarui nilai bobot dan bias sesuai dengan konvergensi gradien *backpropagation* dengan update Polak-Ribere.

8) *Trainoss (one-Step Secant)*

Trainoss adalah fungsi pelatihan jaringan yang memperbarui nilai bobot dan bias dengan metode *one-Step Secant*. Metode *one-Step Secant* adalah upaya untuk menjembatani kesenjangan antara logaritma *gradien* konjugasi dan algoritma kuasi-newton (*secant*). Algoritma ini membutuhkan lebih sedikit penyimpanan dan perhitungan per iterasi dibandingkan algoritma BFGS dan algoritma gradien konjugasi.

9) *Traingdx (Variable Learning Rate Backpropagation)*

Traingdx adalah fungsi pelatihan jaringan yang memperbarui nilai bobot dan bias berdasarkan momentum penurunan *gradien* dan *adaptive learning rate*. Algoritma ini merupakan penggabungan antara algoritma *gradient descent with*

adaptive learning (traindga) dan algoritma *gradient descent with momentum* (traingdm).

2.2.11 Pelatihan Jaringan Syaraf Tiruan Pada MATLAB

Pada pelatihan jaringan syaraf tiruan yang terdapat pada aplikasi Matlab terdapat istilah-istilah yang harus kita ketahui yaitu sebagai berikut:

1. *Epoch* = Bagian pelatihan yang menyatakan jumlah perulangan pembelajaran yang terjadi pada jaringan syaraf tiruan selama pelatihan
2. *Time* = Bagian pelatihan yang menyatakan waktu yang dibutuhkan MATLAB dalam melakukan pembelajaran pada jaringan syaraf tiruan
3. *Performance* = Bagian pelatihan yang menyatakan kualitas pembelajaran didalam MATLAB pada jaringan syaraf tiruan, makin dekat grafik *performance* dengan nilai nol maka kualitasnya semakin baik
4. *Gradient* = Bagian pelatihan yang menyatakan kemiringan antara satu literasi dengan literasi berikutnya, pelatihan akan berhenti jika kemiringan sudah tidak berubah
5. *Validation Check* = Bagian pelatihan yang menyatakan Menyatakan apakah proses pembelajaran mengarah pada tujuan yang cepat atau menyimpang.

2.2.12 Perhitungan Manual Perkiraan Beban

Dalam melakukan metode perhitungan manual ini terlebih dahulu perlu diketahui bahwa perhitungan yang dilakukan terbagi menjadi 2 (dua) tahap, yaitu tahap perhitungan persentase data beban puncak pada 7 (tujuh) tahun sebelumnya dan tahap perhitungan perkiraan untuk 9 tahun kedepan.

Untuk menghitung persentase kenaikan beban puncak pada tahun sebelumnya menggunakan rumus:

$$R_{t-1,t} = \frac{R_t - R_{t-1}}{R_{t-1}} \times 100\%$$

Dimana:

R = beban

t = tahun

R_t = tahun sekarang

R_{t-1} = tahun sebelumnya

Hasil dari rata-rata kenaikan jumlah beban puncak tersebut dapat diasumsikan untuk meramalkan pertumbuhan beban puncak menggunakan rumus:

$$Beban_t = (Beban_{t-1} \times (t-1,)) + Beban_{t-1}$$

2.2.13 Prakiraan Pertumbuhan Penduduk

Dalam membuat prakiraan pertumbuhan penduduk perlu diketahui asumsi nilai persentase pertumbuhan penduduk tiap tahunnya. Nilai persentase pertumbuhan penduduk dapat diperoleh menggunakan persamaan berikut:

$$R_{t-1,t} = \frac{Penduduk_t - Penduduk_{t-1}}{Penduduk_{t-1}} \times 100\%$$

Dimana :

$R_{(t-1,t)}$: presentase pertumbuhan penduduk (%)

$Penduduk_t$: penduduk tahun t (jiwa)

$Penduduk_{t-1}$: penduduk tahun sebelum t (jiwa)

Asumsi nilai persentase pertumbuhan penduduk dijadikan parameter dalam menghitung perkiraan jumlah penduduk di tahun-tahun mendatang. Perkiraan jumlah penduduk di tahun t dapat diperoleh menggunakan persamaan berikut:

$$Penduduk_t = (penduduk_{t-1} \times R_{(t-1,t)}) + Penduduk_{t-1}$$

2.2.14 Prakiraan Pertumbuhan PDRB

Dalam membuat prakiraan pertumbuhan PDRB (Produk Domestik Regional Bruto), perlu diketahui asumsi nilai presentase pertumbuhan PDRB tiap tahunnya. Nilai presentase pertumbuhan PDRB dapat diperoleh menggunakan persamaan berikut:

$$R_{t-1,t} = \frac{PDRB_t - PDRB_{t-1}}{PDRB_{t-1}} \times 100\%$$

Dimana :

$R(t-1,t)$: presentase pertumbuhan penduduk (%)

$PDRB_t$: PDRB t (jiwa)

$PDRB_{t-1}$: PDRB t (jiwa)

Asusmsi nilai persentase pertumbuhan PDRB dijadikan parameter dalam menghitung perkiraan jumlah PDRB di tahun-tahun mendatang. Perkiraan jumlah PDRB ditahun t dapat diperoleh menggunakan persamaan berikut:

$$PDRB_t = (PDRB_{t-1} \times R_{(t-1,t)}) + PDRB_{t-1}$$