

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 peramalan.

Banyak metode peramalan yang dapat digunakan, oleh karena itu diperlukan penyesuaian terhadap jenis data yang digunakan dan jenis peramalan yang akan dilakukan. Penelitian mengenai peramalan 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 peramalan beban listrik jangka panjang di Provinsi D.I Yogyakarta menggunakan *neural network backpropagation*. Didapatkan bahwa peramalan beban listrik Provinsi D.I Yogyakarta menghasilkan rata-rata kelajuan data sebesar 8,1007% setiap tahunnya. Komparasi hasil peramalan beban listrik menggunakan *neural network backpropagation* dengan hasil peramalan beban listrik RUPTL PT.PLN 2015 – 2024 menghasilkan rata-rata persentase perbedaan data sebesar 9,882%.

Fitriyah, Istardi (2011) memprediksi beban listrik Pulau Bali menggunakan metode jaringan syaraf tiruan. Diperoleh bahwa metode JST ini menggunakan beberapa layer tersembunyi dan target yang kemudian diperbandingkan dengan output serta menggunakan fungsi aktivasi model sigmoid biner dengan range antara 0 – 1. Dari hasil Analisa dengan MATLAB, metode backpropagasi menghasilkan prediksi beban listrik Pulau Bali sampai tahun 2035 mencapai 25431 G

Kafahri Arya Hamidie (2009) dengan judul penelitian Metode Koefisien Energi Untuk Peramalan Beban Jangka Pendek pada Jaringan Jawa Madura Bali. Dalam penelitian ini dilakukan Analisa nilai beban mingguan dan harian pada tahun 2009 menggunakan metode koefisien energi. Dari hasil Analisa didapat nilai error beban mingguan 2009 4,525% dan beban harian 2009 5,234%.

Yasin Mohamad (2013) dengan judul Analisa Perkiraan Energi Menggunakan Metode Koefisien Energi Studi Kasus PT. PLN (persero) Area Gorontalo. Dalam penelitian ini menganalisa perkiraan energi listrik jangka pendek yaitu pada tahun 2013 berdasarkan ROT (Rencana Operasi Tahunan). Metode yang digunakan dalam penelitian ini adalah menggunakan metode koefisien energi, dan sumber data yang digunakan adalah data historis tahun 2011 dan 2012.

Yayu Triwulan, Nasrun Hariyanto, Sabat Anwari (2013) dengan judul Peramalan Beban Puncak Listrik Jangka Pendek Menggunakan Metode Jaringan Syaraf Tiruan. Dalam penelitian ini menganalisa peramalan beban dengan metode Jaringan Syaraf Tiruan (JST) yang dijalankan dengan Backpropogation.

Arief Heru Kuncoro dan Rinaldy Dalimi (2005) dengan judul Aplikasi Jaringan Syaraf Tiruan Untuk Peramalan Beban Tenaga Listrik Jangka Panjang Pada Sistem Kelistrikan Di Indonesia. Dalam penelitian ini data yang dipergunakan adalah data yang sebenarnya yaitu dari tahun 1990 sampai tahun 2000. Hasil ramalan beban puncak pada akhir tahun studi (2025) dengan metode JST tidak berbeda jauh dengan ramalan dalam Rencana Umum Ketenagalistrikan Nasional (RUKN), yaitu masing-masing sebesar 85.584 MW dan 79.920 MW (terjadi perbedaan sekitar 6.6%).

Tamizharasi, Khatiresan dan Sreenivasan (2014) dengan judul *Energy Forecasting using Artificial Neural Network*, dihasilkan bahwa data input dan output aktual yang tercatat berpengaruh signifikan terhadap konsumsi energi yang digunakan dalam pelatihan, proses validasi dan pengujian. Prediksi telah dilakukan selama tahun 2005 – 2008, 2010, 2012, dan 2015 dengan hasil yang dihasilkan mendekati kenyataan yang sebenarnya, jauh lebih akurat daripada yang diperoleh dengan metode regresi linier.

Syeto, Fariza, dan Setiawardhana (2010) penelitian yang telah dilakukan mengenai peramalan beban listrik menggunakan jaringan syaraf tiruan metode kohonen. Dapat disimpulkan bahwa peramalan beban listrik menggunakan metode gabungan *backpropagation* dengan kohonn menghasilkan tingkat keakuratan peramalan lebih baik dibanding menggunakan metode gabungan *counterpropagation* dengan kohonen.

Harmawan (2013) dalam penelitian terhadap peramalan beban listrik harian Jawa Tengah dan DIY menggunakan metode SARIMA. Diperoleh bahwa metode SARIMA cocok digunakan untuk meramalkan besarnya beban listrik harian, terutama untuk hari kerja (senin-kamis) dan hari libur akhir pekan (sabtu dan minggu) dengan nilai *error* yang tergolong kecil yaitu pada hari senin, 23 April 2012 (nilai *error* 1.308%) dan hari minggu, 29 April 2012 (nilai *error* 0.688%).

Nugroho, F.A (2016) penelitian terhadap pertumbuhan beban transformator GI 150 KV Cilegon lama. Perhitungan perkiraan beban transformator menggunakan metode regresi linier diperoleh data bahwa pada tahun 2030 perkiraan beban trafo sebesar 59.24 MVA.

2.2 Dasar Teori

2.2.1 Peramalan

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. Ramalan Tunggal (point forecast)
2. Ramalan 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, peramalan dibagi menjadi 3 periode, sesuai dengan materi yang di ramalkan. Dalam peramalan beban listrik, periode peramalan dibagi menjadi 3, yaitu:

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

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

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

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

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

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

2.2.2 Cara-cara Peramalan Beban

Salah satu faktor yang sangat menentukan dalam membuat rencana operasi sistem tenaga listrik adalah peramalan 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 mengekstrapolasir grafik beban di masa lampau ke masa yang akan datang. Setelah dilakukan ekstrapolasi kemudian ditambahkan koreksi-koreksi terhadap hal-hal khusus, baik untuk peramalan 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.

2.2.3 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

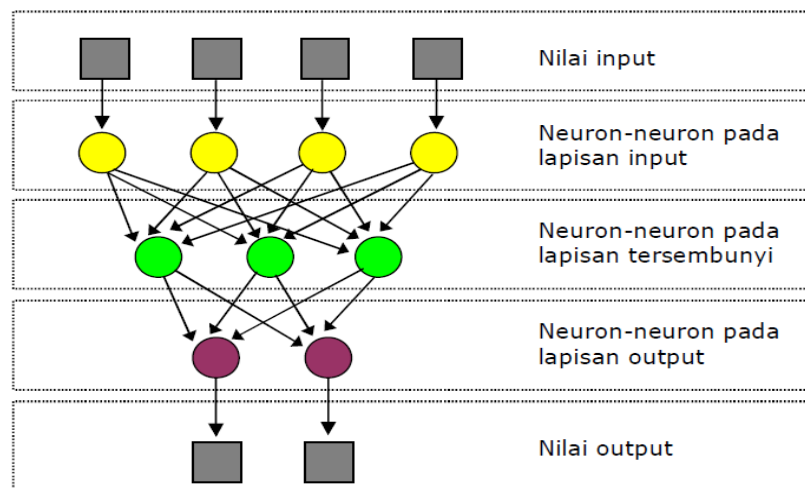
tiruan untuk dapat beradaptasi dengan lingkungan yang memberikan input kepadanya. Layaknya otak manusia dalam merespon kondisi lingkungan berbeda-beda, peranan JST dalam bidang penelitian dan pengembangan sangat penting di masa yang akan datang yang menuntut aspek otomatisasi dan aspek interaktif antara 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/algortma*).
3. Fungsi aktivasi.

2.2.4 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)



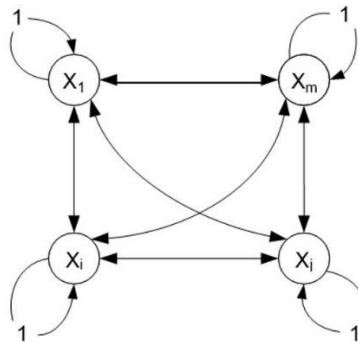
Gambar 2.1 Arsitektur Jaringan Syaraf Tiruan

(Sumber : Ir. Ahmad Haidaroh. Pengenalan Kecerdasan Buatan)

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

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 2.4 Arsitektur jaringan *Recurrent*

2.2.5 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])

2.2.6 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)

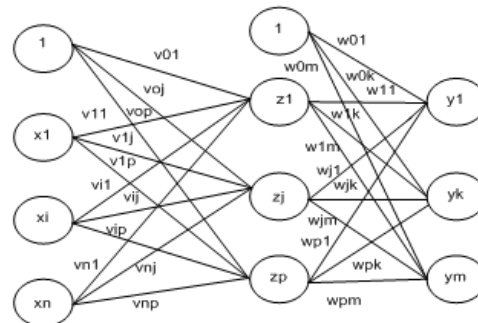
Secara garis besar, mengapa algoritma ini disebut sebagai propagasi balik, karena ketika JST diberikan pola masukan sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapisan tersembunyi untuk diteruskan ke unit lapisan keluaran. Kemudian unit-unit lapis keluaran memberikan tanggapan 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 2.5 Arsitektur *backpropagation*

(Sumber : <http://fauzi.me/page/5/>)

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

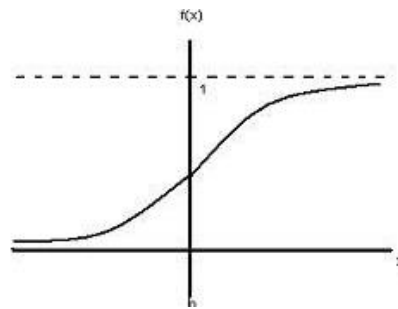
Fungsi aktivasi yang dipakai harus memenuhi beberapa syarat diantaranya: kontinu, terdiferensial dengan mudah dan merupakan fungsi yang tidak turun. Salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga yang sering dipakai ialah fungsi sigmoid biner yang memiliki range $(0, 1)$.

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

Dengan turunan

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

Berikut adalah gambar grafik fungsi sigmoid biner.



Gambar 2.6 Grafik fungsi sigmoid biner

(sumber: <https://sammypatikawa.wordpress.com/2013/01/15/fungsi-sigmoid-biner-untuk-jaringan-saraf-tiruan-backpropagation/>)

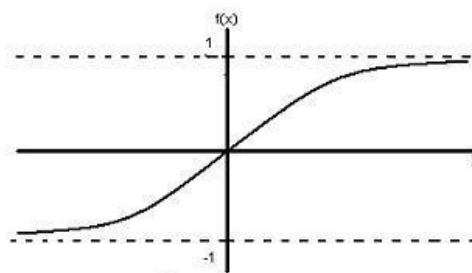
Fungsi lain yang sering dipakai adalah fungsi sigmoid bipolar yang bentuk dan fungsinya mirip dengan fungsi sigmoid biner, tapi dengan range $(-1, 1)$.

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

Dengan turunan

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

Berikut grafik fungsi dari sigmoid bipolar.



Gambar 2.7 Grafik fungsi sigmoid bipolar

(Sumber: <https://jalanwaktu.wordpress.com/jaringan-syaraf-tiruan/>)

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 ($j = 1, 2, \dots, p$)

$$z_{netj} = v_{j0} + \sum_{i=1}^n x_i v_{ji}$$

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

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

$$y_{netj} = w_{k0} + \sum_{j=1}^p z_j w_{kj}$$

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

Fase II : Propagasi mundur

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

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

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

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

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 :

$$\delta_j = \delta_{netj} f'(z_{netj}) = \delta_{netj} z_j (1 - z_j)$$

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

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

Fase III : Perubahan Bobot

Langkah 8 : hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran:

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

Perubahan bobot garis yang menuju ke unit tersembunyi:

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

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.

2.2.7. 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-Beale*

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* (traingda) dan algoritma *gradient descent with momentum* (traingdm).

2.2.8 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 ditahun t dapat diperoleh menggunakan persamaan berikut:

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

2.2.9 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)

Asumsi 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:

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