

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Berdasarkan penelitian tentang proteksi transformator yang akan dibuat ini maka penulis mengacu pada beberapa referensi untuk mempertimbangkan permasalahan yang menjadi acuan dalam mendukung penyusunan penelitian ini, seperti dapat dilihat pada Tabel 2.1 berikut:

Tabel 2. 1 Daftar referensi tugas akhir

Nama Penulis	Tahun	Judul Penelitian	Hasil Penelitian
Kuncoro dan Dalimi	2005	Jaringan syaraf tiruan untuk peramalan beban tenag listrik jangka Panjang di Indonesia	Pada penelitian ini ramalan yang dilakukan tidak jauh berbeda dengan Rencana Umum Ketenagalistrikan Nasional (RUKN), yaitu masing-masing sebesar 85.584 MW dan 79.920 MW (perbedaannya sekitar 6,6%). Hasil simulasi JST untuk periode studi 2001-2025, memperlihatkan bahwa laju kenaikan beban tahunan rata-rata sekitar 7,1%. Ini merupakan suatu refleksi keadaan pertumbuhan ekonomi suatu negara yang baik dan stabil.
Fitriah dan Istardy	2011	Prediksi beban listrik pulau bali dengan menggunakan metode backpropagasi	Hasil pada penelitian ini bahwa metode backpropagasi bisa dilakukan untuk peramalan beban ketenagalistrikan. Metode ini menggunakan satu atau beberapa

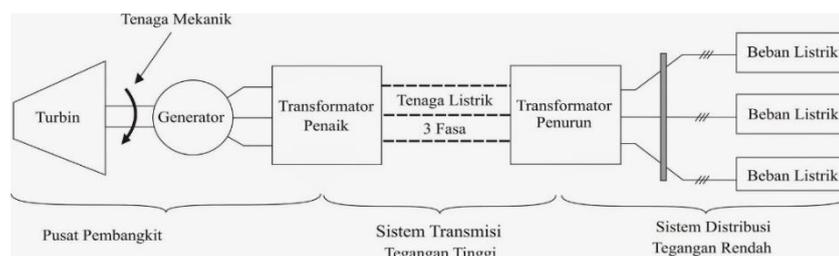
Nama Penulis	Tahun	Judul Penelitian	Hasil Penelitian
			layar tersembunyi dan juga target yang kemudian diperbandingkan dengan output. Fungsi aktivasi menggunakan sigmoid biner dengan range antara 0-1. Dari hasil analisa dengan MATLAB, bahwa metode backpropagasi bisa digunakan untuk memprediksi beban listrik Pulau Bali sampai tahun 2035 yang akan mempunyai beban energi listrik sebesar 25431 GWh.
Elias K. Bawan	2013	Estimasi Pembebanan Transformator Gardu Induk 150KV	Standar toleransi kelayakan kapasitas transformator Gardu Induk wirobrajan sebesar 85 % yaitu 50.89 MVA untuk fungsi eksponensial dan 48.07 MVA dengan fungsi polinomial dengan arus pembebanan 84.81% yaitu sebesar 230.72 Ampere tercapai pada tahun 2025.
Trywulan, Yuyu Dkk	2011	Peramalan beban puncak listrik jangka pendek menggunakan jaringan syaraf tiruan	Menggunakan peramalan beban listrik dengan jaringan syaraf tiruan <i>backpropagation</i> lebih baik dibandingkan metode koefisien beban PLN. Hasil yang didapatkan dari penelitian ini menggunakan jaringan syaraf tiruan <i>backpropagation</i> nilai rata-rata <i>error</i>

Nama Penulis	Tahun	Judul Penelitian	Hasil Penelitian
			0,12% dengan tingkat keakurasian 99,88%, sedangkan menggunakan metode koefisien beban PLN didapatkan nilai rata-rata <i>error</i> 1,85% dengan tingkat akurasi 98,15%.

2.2 Landasan Teori

2.2.1 Sistem Tenaga Listrik

Secara umum, pengertian sistem tenaga listrik adalah sekumpulan pusat listrik dan Gardu Induk yang satu sama lain dihubungkan oleh sistem penyaluran sehingga merupakan satu kesatuan sistem. Secara garis besar sistem tenaga listrik memiliki 3 sub sistem, seperti dapat dilihat pada Gambar 2.1 berikut:



Gambar 2. 1 Sistem tenaga listrik

Sumber: <https://agusbudiana1.blogspot.com/2014/04/sistem-tenaga-listrik.html>

Sistem tenaga listrik memiliki 3 sub sistem dalam menyalurkan energi listrik dari sumber tegangan hingga ke konsumen:

1. Pusat pembangkit merupakan penyedia energi listrik yang kemudian disalurkan menuju sistem transmisi.
2. Sistem transmisi merupakan penyaluran energi listrik dari pusat pembangkit menuju sistem distribusi.
3. Sistem distribusi merupakan penyaluran energi listrik dari sistem transmisi menuju konsumen.

2.2.2 Gardu Induk

Gardu Induk merupakan suatu instalasi yang terdiri dari peralatan listrik yang merupakan pusat beban yang disusun menurut pola dan kegunaan dengan pertimbangan teknis, keindahan serta ekonomis. Ada beberapa fungsi dari Gardu Induk yaitu:

1. Mentransformasi tenaga listrik dari tegangan tinggi ke tegangan rendah begitu sebaliknya.
2. Pengukuran, pengawasan operasi serta pengaturan dari pengamanan dari sistem tenaga listrik.
3. menyalurkan energi listrik dan menerima energi listrik sesuai dengan ketentuan.

2.2.3 Jenis Gardu Induk

Gardu Induk dapat diklasifikasikan dalam beberapa sub bagian, yaitu:

1. Berdsarkan jenis pemasangan,
 - a. Gardu Induk pasangan dalam ialah Gardu Induk yang hampir semua komponennya dipasang dalam gedung. Kecuali transformator daya, pada umumnya dipasang di luar gedung. Gardu Induk ini biasanya berada di lingkungan padat penduduk.
 - b. Gardu Induk pasangan luar ialah Gardu Induk yang sebagian besar komponennya di pasang di luar gedung, kecuali komponen control, sitem proteksi dan sistem kendali serta komponen bantu lainnya.
 - c. Gardu Induk kombinasi ialah Gardu Induk yang komponennya ditempatkan di dalam gedung dan sebagian komponen lainya ditempatkan di luar gedung, denga memperhatikan keadaan dan kondisi.

2. Berdasarkan Tegangan

- a. Gardu Induk transmisi

Gardu Induk transmisi ialah Gardu Induk yang mendapat daya dari saluran transmisi untuk kemudian menyalurkannya ke daerah. Gardu Induk ini melayani tegangan tinggi 70 kV dan tegangan tinggi 150 kV

b. Gardu Induk distribusi

Gardu Induk distribusi ialah Gardu Induk yang menerima dan menyuplai daya dari Gardu Induk transmisi dengan menurunkan tegangannya menjadi tegangan menengah hingga ke tegangan rendah.

1. Berdasarkan Fungsi

a. Gardu Induk pengatur tegangan

Gardu Induk pengatur tegangan umumnya berada jauh dari pusat pembangkit sehingga terjadi tegangan jatuh transmisi cukup besar sehingga diperlukan alat penaik tegangan agar tegangan kembali normal

b. Gardu Induk penaik tegangan

Gardu Induk penaik tegangan yaitu Gardu Induk untuk menaikkan tegangan yang biasanya ada di dekat pembangkit listrik. Dengan mempertimbangkan efisiensi, tegangan dinaikkan menjadi tegangan ekstra tinggi atau tegangan tinggi

c. Gardu Induk penurun tegangan

Gardu Induk penurun tegangan untuk menurunkan tegangan, misalnya dari tegangan ekstra tinggi menjadi tegangan tinggi.

d. Gardu Induk pengatur beban

Gardu Induk pengatur beban yaitu untuk mengatur beban. Gardu Induk ini terpasang beban motor, yang suatu kejadian menjadi pembangkit tenaga listrik, dan pada suatu kejadian juga motor bisa menjadi beban

e. Gardu distribusi

Gardu distribusi yaitu untuk menyuplai energi listrik dari tegangan sistem ke tegangan distribusi.

2.2.3 Transformator

Transformator merupakan peralatan utama dalam sistem tenaga listrik yang digunakan sebagai menaikkan dan menurunkan besaran tegangan tanpa mengubah besaran frekuensi yang bekerja berdasarkan prinsip elektromagnetik. Dimana mula

arus listrik mengalir melalui lilitan kawat, yang membuat kawat tersebut seolah menjadi energi magnet.

Transformator adalah peralatan yang berfungsi untuk menurunkan atau menaikkan tegangan bolak-balik. Pada transformator terdapat 3 komponen utama yaitu kumparan sisi primer berfungsi sebagai masukan, kumparan sisi sekunder berfungsi sebagai keluaran dan inti besi berfungsi menghasilkan fluks magnet.

Kumparan tersebut mengelilingi inti besi dalam bentuk lilitan. Jika kumparan pada sisi primer trafo dihubungkan dengan suatu sumber tegangan bolak-balik sinusoidal, maka akan mengalir arus bolak-balik yang juga sinusoidal pada kumparan tersebut. Arus bolak-balik ini akan menimbulkan fluks magnetik yang sefasa dan juga sinusoidal di sekeliling kumparan. Akibat adanya inti trafo yang menghubungkan kumparan pada sisi primer dan kumparan pada sisi sekunder, maka fluks magnetik akan mengalir bersama pada inti trafo dari kumparan primer menuju kumparan sekunder sehingga akan membangkitkan tegangan induksi pada sisi sekunder trafo

2.2.4.1 Pembebanan Transformator

Pembebanan transformator didapatkan dari hasil prakiraan beban dibagi dengan kapasitas transformator, kapasitas transformator tergantung pada transformator yang dipakai Klasifikasi pembebanan dibagi menjadi empat, yaitu :

1. Beban Kecil

Pada kondisi beban kecil transformator terjadi ketika beban transformator kurang dari 60%. Pada kondisi ini, arus yang mengalir pada transformator kecil sehingga hanya mempengaruhi rugi tembaga dan tidak akan mempengaruhi rugi inti besi (rugi inti besi tetap), sehingga transformator bekerja tidak efisien.

2. Beban Optimal

Pada kondisi beban optimal transformator terjadi ketika beban transformator diantara 60% sampai dengan 80%. Pada kondisi ini, kerja transformator optimal

3. Beban Berat

Pada kondisi beban berat transformator terjadi ketika beban transformator diatas 80% sampai dengan 100%. Pada kondisi ini, kerja transformator berat sehingga menyebabkan panas yang berlebih pada transformator.

4. *Overload*

Pada kondisi beban Overload transformator terjadi ketika beban transformator diatas 100%. Pada kondisi ini, kerja transformator sangat berat sehingga menyebabkan transformator sangat panas bahkan mempunyai resiko terbakarnya transformator.

$$\text{Porsentase Pembebanan} = \frac{Y_x}{K_t} \times 100\% \dots \dots \dots (2.1)$$

Keterangan: Y_x = Pembebanan pada tahun x (MVA)

K_t = Kapasitas Transformator (MVA)

2.2.4.2 Capacity Balance Gardu Induk

Capacity balance Gardu Induk adalah cara mengetahui batas kapasitas transformator Gardu Induk dalam mendukung beban, yang dikaitkan peningkatan kebutuhan tenaga listrik berdasarkan prakiraan. Dengan capacity balance, dapat ditentukan tahun persiapan ekstensifikasi transformator baru dan pengadaan GI baru. Syarat-syarat Gardu Induk adalah:

1. Dalam satu Gardu Induk (GI) hanya diijinkan 3 (tiga) buah transformator
2. Kapasitas transformator tertinggi dalam setiap GI adalah 60 MVA
3. Pembebanan transformator tidak boleh melebihi 80% dari kapasitas transformator.
4. Bila beban transformator mendekati 80%, harus dipersiapkan :
 - a. Uprating, bila kapasitas transformator masih di bawah 60 MVA.
 - b. Ditambahkan transformator baru, bila kapasitas transformator sudah 60 MVA dan di GI tersebut jumlah transformator masih kurang dari 3 (tiga)
 - c. Pembangunan Gardu Induk baru dengan transformator baru

2.2.5 Prakiraan Beban

Prakiraan adalah suatu dugaan yang atas kejadian yang terjadi dimasa yang akan datang. Prakiraan ini terjadi karena adanya perbedaan waktu antara kesadaran akan peristiwa atas kejadian yang akan datang. Apabila jarak waktu terhadap kejadian tersebut sangat panjang maka suatu Prakiraan sangat dibutuhkan dalam suatu penentuan apapun, misalnya prakiraan prakiraan beban transformator.

Prakiraan dibedakan menjadi dua macam (Assauri,1984) yaitu :

1. Prakiraan Kualitatif

Prakiraan kualitatif merupakan prakiraan yang didasarkan atas data kumulatif pada masa lalu. Hasil prakiraan yang dibuat sangat tergantung pada orang yang menyusunnya. Hal ini penting sebab hasil prakiraan tersebut ditentukan berdasarkan intuisi, pendapat dan pengetahuan serta pengalaman penyusun.

2. Prakiraan Kuantitatif

Prakiraan kuantitatif merupakan prakiraan yang dibuat sangat tergantung pada metode yang digunakan dalam prakiraan tersebut. Baik tidaknya metode yang digunakan ditentukan oleh perbedan antara hasil prakiraan dengan kenyataan yang terjadi. Semakin kecil penyimpangan antara hasil prakiraan dan kenyataan maka metode prakiraan tersebut semakin baik.

2.2.5.1 Prakiraan Pertumbuhan Penduduk

Dalam prakiraan pertumbuhan penduduk perlu adanya nilai asumsi persentase pertumbuhan penduduk pada tiap tahunnya. Nilai pertumbuhan penduduk di dapatkan dengan persamaan berikut:

$$R_{(t-1,t)} = \frac{Penduduk_t - Penduduk_{t-1}}{Penduduk_{t-1}} \times 100 \% \dots\dots\dots(2.2)$$

Keterangan: $R_{(t-1,t)}$: Persentase pertumbuhan penduduk

$Penduduk_t$: penduduk tahun t (jiwa)

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

Asumsi persentase pertumbuhan penduduk dapat dijadikan sebagai parameter dalam menentukan jumlah penduduk di tahun yang akan datang. Prakiraan jumlah penduduk pada tahun t dijabarkan dalam persamaan berikut ini:

$$Penduduk_t = (Penduduk_{t-1} \times R_{t-1,t}) + Penduduk_{t-1} \dots\dots\dots(2.3)$$

2.2.5.2 Prakiraan Pertumbuhan Produk Domestik Regional Bruto (PDRB)

Dalam memperkirakan pertumbuhan Produk Domestik Regional Bruto (PDRB) dapat diasumsikan pada nilai presentase pertumbuhan PDRB setiap tahunnya dengan menggunakan persamaan berikut ini:

$$R_{(t-1,t)} = \frac{PDRB_t - PDRB_{t-1}}{PDRB_{t-1}} \times 100 \% \dots\dots\dots(2.4)$$

Keterangan: $R_{(t-1,t)}$: Persentase pertumbuhan PDRB

$PDRB_t$: PDRB tahun t (jiwa)

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

Asumsi persentase PDRB dapat dijadikan sebagai parameter dalam menentukan jumlah PDRB di tahun yang akan datang. Prakiraan jumlah PDRB pada tahun t dijabarkan dalam persamaan berikut ini:

$$PDRB_t = (PDRB_{t-1} \times R_{t-1,t}) + PDRB_{t-1} \dots\dots\dots(2.5)$$

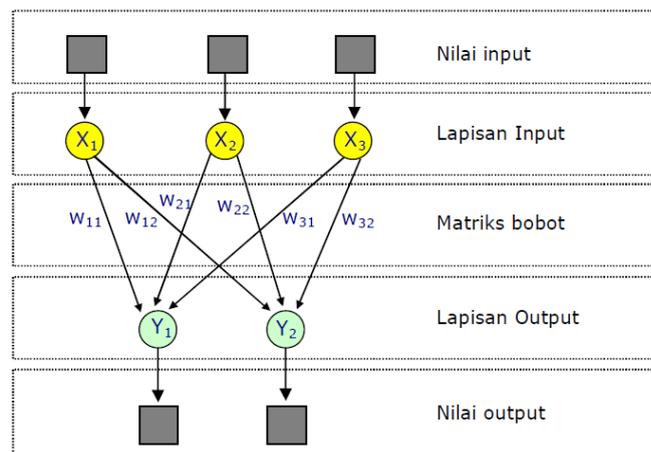
2.2.6 Jaringan Saraf Tiruan

Jaringan syaraf tiruan (JST) merupakan sistem kecerdasan tiruan dengan kemampuan belajar dan menghimpun pengetahuan hasil pembelajaran 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 mendatang yang menuntut aspek otomatisasi dan aspek interaktif antara alat dan manusia. (Muis, 2017).

2.2.6.1 Arsitektur Jaringan

Dalam jaringan saraf tiruan, neuron ini dikelompokkan menjadi lapisan-lapisan (layer). Neuron pada setiap lapisan akan dihubungkan dengan neuron pada lapisan yang lain. Informasi akan melewati satu lapisan ke lapisan selanjutnya, mulai dari input sampai ke output lapisan melalui hidden lapisan, seperti dapat dilihat pada Gambar 2.2 berikut ini:



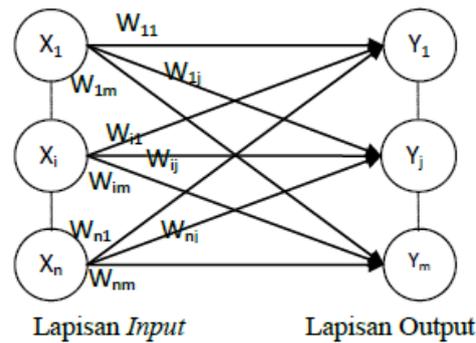
Gambar 2. 2 Arsitektur Jaringan Saraf Tiruan

(Sumber: <http://saraf-tiruan.blogspot.com/2016/10/arsitektur-jaringan-syaraf-tiruan.html>)

Menurut Haykin (2009), secara umum, ada tiga jenis arsitektur dari Jaringan Saraf Tiruan yaitu:

1. Jaringan Lapisan Tunggal (single layer network)

Pada JST dengan satu layer ini, merupakan lapisan yang terdiri dari beberapa neuron. Beberapa neuron input ini di sambungkan dengan neuron output lainnya. Tidak ada unit input yang dihubungkan dengan unit input lainnya. Demikian pula dengan unit output, seperti dapat dilihat pada Gambar 2.3 berikut ini:

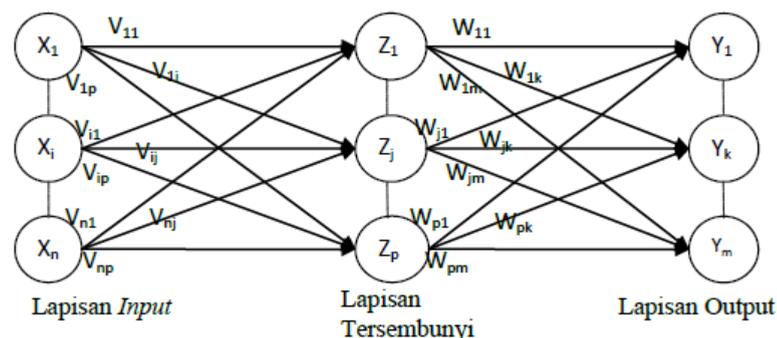


Gambar 2. 3 Jaringan layar tunggal

(Sumber: <https://www.kajianpustaka.com/2016/11/jaringan-saraf-tiruan-jst.html>)

2. Jaringan Layar Jamak (*multi layer network*)

Pada jaringan layar jamak ini agak sedikit tambahan dari layer tunggal, yaitu adanya layer tersembunyi. Jumlah layer tersembunyi dapat lebih dari satu. Sama seperti pada unit input dan output, unit-unit dalam satu layer tidak saling berhubungan, seperti dapat dilihat pada Gambar 2.4 berikut ini:

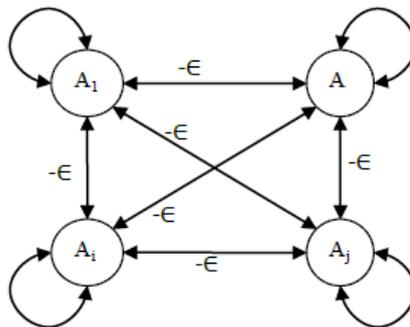


Gambar 2. 4 Jaringan layar jamak

(Sumber: <https://www.kajianpustaka.com/2016/11/jaringan-saraf-tiruan-jst.html>)

3. 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, seperti dapat dilihat pada Gambar 2.5 berikut ini:



Gambar 2. 5 Jaringan recurrent

(Sumber: <https://www.kajianpustaka.com/2016/11/jaringan-saraf-tiruan-jst.html>)

2.2.6.2 Fungsi Aktivasi

Pada fungsi aktivasi ialah hanya dipakai untuk menentukan keluaran suatu neuron. Berikut adalah fungsi aktivasi yang sering dipakai:

- a. Fungsi threshold (batas ambang)

$$f(x) = \begin{cases} 1, & \text{jika } x \geq a \\ 0, & \text{jika } x \leq a \end{cases} \dots\dots\dots(2.6)$$

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

$$f(x) = \begin{cases} 1, & \text{jika } x \geq a \\ 0, & \text{jika } x \leq a \end{cases} \dots\dots\dots(2.7)$$

- b. Fungsi Sigmoid

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

$$f(x) = \frac{1}{1+e^{-x}} \dots\dots\dots(2.8)$$

- c. Fungsi identitas

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

$$f(x) = x \dots\dots\dots(2.9)$$

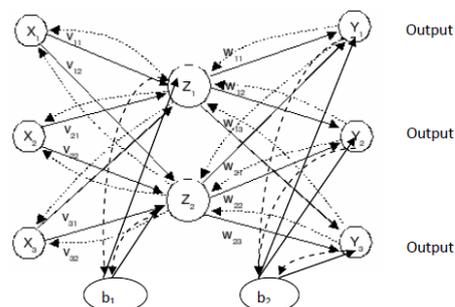
2.2.6.3 Algoritma *Backpropagation*

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 tersembunyinya. Algoritma *backpropagation* menggunakan error output untuk mengubah nilai bobot-bobotnya adalah arah mundur (*backward*). Untuk mendapatkan error ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu.

JST diberikan pola masukan sebagai pola pelatihan maka pola tersebut menuju ke unit-unit pada lapis tersembunyi untuk diteruskan ke unit-unit lapis keluaran oleh karena itu algoritma ini disebut sebagai propagasi balik,. 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 lapis masukan. Oleh karenanya maka mekanisme pelatihan tersebut dinamakan *backpropagation*. standar algoritma *backpropagation* adalah

a. Arsitektur *backpropagation*

Backpropagation memiliki beberapa unit yang ada dalam satu atau lebih layar tersembunyi (*hidden layer*). Gambar 2.4 adalah arsitektur *backpropagation* dengan n buah masukan (ditambah sebuah bias), sebuah layar tersembunyi yang terdiri dari p unit (ditambah sebuah bias), serta m unit keluaran, seperti dapat dilihat pada Gambar 2.6 berikut ini:



Gambar 2. 6 Arsitektur *Propagation*

Sumber: <https://dafikrifajri.wordpress.com/arsitektur-backpropagation/>

V_{ij} merupakan bobot garis dari unit masukan x_i ke unit layar tersembunyi z_j (v_{0j} merupakan bobot garis yang menghubungkan bias di unit masukan ke unit layar tersembunyi z_j). w_{jk} merupakan bobot dari unit layar tersembunyi z_j ke unit keluaran y_k (w_{0k} merupakan bobot dari bias di layar tersembunyi ke unit keluaran y_k).

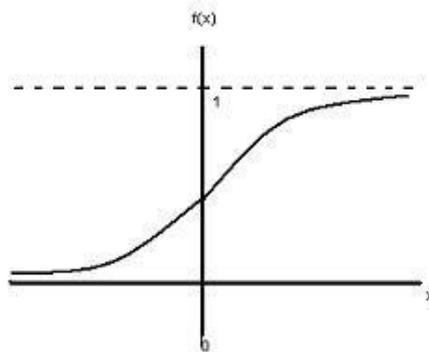
b. Fungsi Aktivasi

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

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

Turunannya ialah $f'(x) = f(x)(1 - f(x))$ (2.10)

Grafik dari fungsi sigmoid biner seperti dapat dilihat pada Gambar 2.7 berikut ini



Gambar 2. 7 Grafik Fungsi sigmoid

<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.

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 adalah sebagai berikut:

Fase I

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

$$z_{netj} = v_{jo} + \sum_{i=1}^n x_i v_{ji} \dots\dots\dots(2.11)$$

$$z_j = f(z_{netj}) = \frac{1}{1 + e^{-z_{netj}}} \dots\dots\dots(2.12)$$

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

$$y_{netj} = w_{ko} + \sum_{j=1}^p z_j w_{kj} \dots\dots\dots(2.13)$$

$$y_k = f(y_{netk}) = \frac{1}{1 + e^{-y_{netk}}} \dots\dots\dots(2.14)$$

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

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

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

$$\delta_{netj} = \sum_{k=1}^m \delta w_{kj} \dots \dots \dots (2.15)$$

Faktor δ unit tersembunyi :

$$\delta_j = \delta_{netj} f'(z_{netj}) = \delta_{netj} z_j (1 - z_j) \dots \dots \dots (2.16)$$

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

$$\Delta v_{ji} = \alpha \delta_j x_i ; j = 1, 2, \dots, p; i = 0, 1, \dots, n \dots \dots \dots (2.17)$$

Fase III : Perubahan Bobot

Langkah 8 : hitung semua perubahan bobot

Perubahan bobot garis yang menuju ke unit keluaran:

$$w_{kj}(\text{baru}) = w_{kj}(\text{lama}) + \Delta w_{kj} (k = 1, 2, \dots, m; j = 0, 1, \dots, p) \dots \dots \dots (2.18)$$

Perubahan bobot garis yang menuju ke unit tersembunyi:

$$v_{ji}(\text{baru}) = v_{ji}(\text{lama}) + \Delta v_{ji} (j = 1, 2, \dots, p; i = 0, 1, \dots, n) \dots \dots \dots (2.19)$$

Setelah selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (lagkah 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 Jaringan pada MATLAB

MATLAB menyediakan fitur neural network yang memudahkan penggunaannya dalam membangun suatu jaringan syaraf tiruan. 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*

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*

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*).