BAB II
LANDASAN TEORI

2.1 Tinjauan Pustaka

Dalam penulisan penelitian tugas akhir ini peneliti menggali informasi dari penelitian-penelitian sebelumnya sabagai bahan perbandingan, baik mengenai kekurangan atau kelebihan yang sudah ada. Selain itu, peneliti juga menggali informasi dari buku-buku maupun penelitian sebelumnya dalam rangka mendapatkan suatu informasi yang ada sebelumnya tentang teori yang berkaitan dengan judul yang digunakan untuk memperoleh landasan teori ilmiah.

Berikut ini adalah ringkasan komprehensif dari penelitian sebelumnya tentang suatu topik yang berkaitan dengan penelitian Tugas Akhir ini, Jurnalnya antara lain :

a. *Artificial Neural Network* Untuk Memprediksi Beban Listrik Dengan Menggunakan Metode *Backpropagation.* Studi Kasus di PT. PLN Regional Sumatera Barat. Prediksi beban pemakaian listrik di pengaruhi oleh data input jumlah daya dan pelanggan dari berbagai macam sektor sehingga di ketahui jumlah beban listrik sebagai target. Data yang digunakan adalah data laporan penjualan aliran listrik dari tahun 2012 - 2016 PT. PLN Regional Sumatera Barat. Implemetasi dari *artificial neural network* metode *backpropagation* menggunakan *matlab* 8.5 sebagai *software* pendukung. Penelitian ini dalam memprediksi beban pemakaian listrik menggunakan *matlab* versi 8.5 dan dilakukan pengujian terhadap data *testing* tahun 2016 didapatkan nilai MSE 0.00095761, sehingga aplikasi *matlab* 8.5 menggunakan metode *backpropagation* dapat

memprediksi beban pemakaian listrik jangka menengah dengan baik (Wijaya, 2019).

b. *Artificial Neural Network Base Short-term Electricity Load Forecasting: A Case Study of a 132/33 kv Transmission Sub-station.* Objek yang di teliti adalah Port-Harcourt gardu transmisi di Negara Nigeria. Dalam penelitiannya menjelaskan bahwa ada empat parameter input untuk bagian input yang merupakan data muatan historis per jam (dalam MW), waktu dalam sehari (dalam jam), hari dalam seminggu dan akhir pekan sementara parameter *output* setelah pemrosesan (pelatihan, validasi, dan tes) adalah beban per jam minggu depan yang diprediksi untuk seluruh sistem. Teknik yang digunakan adalah JST dengan bantuan perangkat lunak *MATLAB*. Itu terbukti menjadi metode perkiraan yang baik karena menghasilkan nilai-R 0,988 yang memberikan deviasi absolut rata-rata 0,104 dan MSE 0,27. Dapat disimpulkan bahwa model JST adalah sangat baik sebagai metode untuk memprediksi perkiraan beban jangka pendek dengan mempertimbangkan tinggi Nilai R yang diperoleh (Samuel, et.al, 2019).

c. *Power Consumption Estimation Using Artificial Neural Networks: The Case Of Turkey.* Terdapat empat faktor utama yang dianggap mempengaruhi konsumsi listrik di Turki sebagai variabel independen. Variabel independen ini adalah; Populasi, Impor, Ekspor, Produk Domestik Bruto (PDB). Dalam studi yang dilakukan untuk solusi masalahnya, mereka menguji algoritma *backpropagation* dari jaringan saraf tiruan sebagai alat estimasi. Karena margin kesalahan 6,17E + 07 ternyata cukup dapat diterima, estimasi konsumsi listrik antara tahun 2017 dan 2030, yang merupakan tujuan utama penelitian ini, dilakukan dengan parameter yang sama, 6,17E + 07 margin kesalahan, untuk mendapatkan estimasi target (Metin, H.H dan Sibkat Kaçtıoğlu, 2018).

d. Prediksi Pembayaran Tagihan Listrik Menggunakan Model *Artificial Neural Network*. Prediksi biaya tagihan listrik bertujuan untuk menghasilkan prediksi yang lebih akurat yang selanjutnya digunakan sebagai pedoman untuk mengetahui dan merencanakan biaya listrik

kedepannya, guna meminimalisir kekeliruan dalam merencanakan anggaran. *Dataset* yang digunakan adalah *dataset primer time series* mulai Bulan Januari 2011 sampai Bulan Desember 2015. Metode yang digunakan pada penelitian ini adalah model *Artificial Neural Network* (ANN). Setelah beberapa kali dilakukan pengujian menghasilkan *performance* RMSE 0.090 dengan waktu eksekusi 1 detik. Hasil ini membuktikan bahwa metode dengan model ANN dapat digunakan sebagai metode untuk melakukan prediksi terhadap pembayaran tagihan listrik secara lebih akurat (Rahman, 2018).

e. *Effective Sampling and Windowing for an Artificial Neural Network Model Used in Currency Exchange Rate Forecasting,* Sebagian besar penelitian berfokus pada hal yang menjanjikan model prediksi dalam kecerdasan komputasi dikembangkan dalam beberapa dekade terakhir, yaitu Jaringan Saraf Tiruan (JST). Namun, tinjauan literatur yang ada menunjukkan bahwa *time step, prediction horizon and window size* belum dari esensi pusat. Oleh karena itu, makalah ini berupaya untuk memberikan sebuah analisis yang lebih formal tentang dampak aktual dari ketiga parameter yang disebutkan pada hasil prediksi dari JST. Melalui studi literatur, pemodelan dan eksperimen ditemukan bahwa tidak ada yang spesifik kombinasi *time step, prediction horizon and window size* menghasilkan perkiraan yang lebih tepat, tapi itu kombinasi tertentu dari ketiga parameter ini umumnya menghasilkan kinerja yang unggul (Johansson, Sam dan Shayan Nafar, 2017).

2.2 Definisi-definisi Peramalan Beban

Berikut di bawah ini adalah beberapa definisi yang berhubungan mengenai peramalan beban.

2.2.1 Pengertian Peramalan Beban

Peramalan pada dasarnya merupakan suatu dugaan atau prakiraan mengenai terjadinya suatu kejadian atau peristiwa dimasa yang akan datang. Ramalan di bidang tenaga elektrik pada dasarnya merupakan ramalan kebutuhan

energi elektrik dan ramalan beban tenaga elektrik (watt). Keduanya sering disebut dengan istilah *Demand and Load Forecasting*. Hasil peramalan ini dipergunakan untuk membuat rencana pemenuhan kebutuhan, maupun pengembangan penyediaan tenaga elektrik setiap saat, secara cukup dan baik serta terus menerus (Daman, 2014).

Secara garis besar pembuatan ramalan kebutuhan tenaga elektrik dapat dibagi dalam tiga tahap :

a. Pengumpulan dan penyiapan data.

b. Pengolahan dan analisa data.

c. Penentuan metoda dan pembuatan model.

Menurut jangka waktunya, peramalan dibagi menjadi 3 periode. Dalam peramalan beban listrik, ketiga periode peramalan beban listrik tersebut yakni : (Daman, 2014).

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

   Merupakan peramalan yang memperkirakan keadaan dalam waktu beberapa tahun kedepan (diatas 10 tahun). Tujuannya adalah untuk mempersiapkan ketersediaan unit pembangkit, system transmisi, serta distribusi.

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

   Merupakan peramalan dalam jangka waktu (3 sampai 10 tahun). Tujuannya untuk mempersiapkan jadwal persiapan dan operasional sisi pembangkit.

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

   Merupakan peramalan dalam jangka waktu jam, hari, atau bulan dalam 2 tahun. Biasa digunakan untuk studi perbandingan beban listrik perkiraan dengan data aktual realtime.

2.2.2 Tipe-tipe Beban

Ada beberapa tipe beban pada beban listrik, yaitu sebagai berikut : (Sari, 2014).

- **Beban Perumahan (Domestic)**

Pada beban perumahan terdiri dari beban penerangan, kipas angin, alat-alat rumah tangga misalnya pemanas, lemari es, alat pendingin udara *(air conditioner)*, alat pengaduk, alat pemanggang, kompor listrik dan motor-motor kecil untuk pompa, dan lainnya. Untuk faktor kebutuhan pada beban rumah tangga biasanya 70-100% dan faktor bebannya 10-15 %.

- **Beban Komersial**

Beban komersial terdiri atas penerangan untuk toko-toko, reklame, *air conditioner*, dan beban listrik lainnya yang dipakai pada bangunan perdagangan seperti toko-toko, mall, restoran, pasar-pasar, dan lainnya. Faktor kebutuhan untuk beban komersial biasanya 90-100% dan faktor bebannya sebesar.

- **Beban Industri**

Beban industri ini memiliki tingkat daya seperti berikut :

- Industri rumah tangga : 5 KW
- Industri skala kecil : 5-25 KW
- Industri skala menengah : 25-100 KW
- Industri besar : 100-500 KW
- Industri berat : diatas

- **Beban Kota**

Beban kota ini adalah untuk penerangan jalan, penerangan taman kota, dan lain sebagiannya yang akan hidup sepanjang.

- **Beban-Beban Lainnya**

Diluar beban-beban yang telah disebutkan diatas, masih terdapat beban-beban yang lainnya seperti rumah ibadah, beban perkantoran dan industri lainnya yang memiliki pembangkit sendiri.

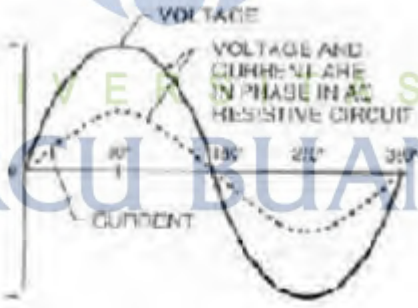
2.3 Pengertian Beban Listrik

Pemakaian beban listrik sektor rumah tangga pada umumnya memuncak pada malam hari. Peralatan – peralatan tersebut umumnya mempunyai nilai pemakaian energi yang besar berbeda tergantung pada komponen peralatan. Untuk komponen peralatan rumah tangga tidak lepas dari ketiga beban listrik. Beban listrik resistif murni tardapat pada penerangan yaitu lampu pijar, sedangkan untuk lampu hemat energi terdapat beban induktif di dalamnya. Begitu juga untuk paralatan rumah tangga lainnya yang termasuk dalam kategori peralatan elektronik ketiga beban di atas termasuk didalamnya.

Dalam sistem listrik arus bolak-balik, jenis beban dapat diklasifikasikan menjadi 3 macam, yaitu :

2.3.1 Beban Resistif (R)

Beban resistif (R) yaitu beban yang terdiri dari komponen tahanan ohm saja *(resistance)*, seperti elemen pemanas *(heating element)* dan lampu pijar. Beban jenis ini hanya mengkonsumsi beban aktif saja dan mempunyai faktor daya sama dengan satu. Tegangan dan arus satu fasa.



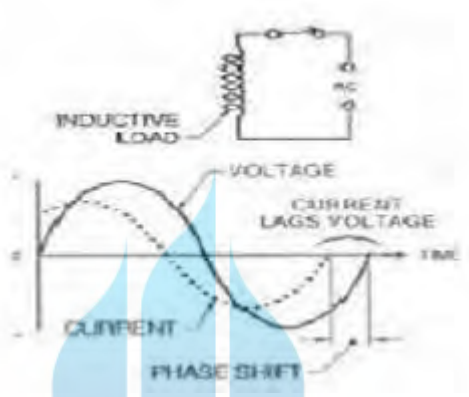
Gambar 2.1 Gelombang Resisitif AC

Gambar 2.2 Grafik Arus dan Tegangan Pada Beban Resistif

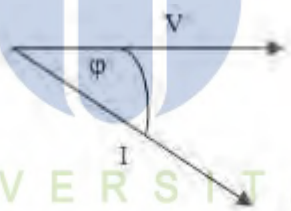
Sumber: Ramdhani, Mohamad. 2008. Rangkaian Listrik. Erlangga : Jakarta

2.3.2 Beban Induktif (L)

Beban induktif (L) yaitu beban yang terdiri dari kumparan kawat yang dililitkan pada suatu inti, seperti coil, *transformator*, dan solenoida. Beban ini dapat mengakibatkan pergeseran fasa *(phase shift)* pada arus sehingga bersifat *lagging*. Hal ini disebabkan oleh energi yang tersimpan berupa medan magnetis akan mengakibatkan fasa arus bergeser menjadi tertinggal terhadap tegangan.



Gambar 2.3 Gelombang Induktif AC



Gambar 2.4 Grafik Arus dan Tegangan Pada Beban Induktif

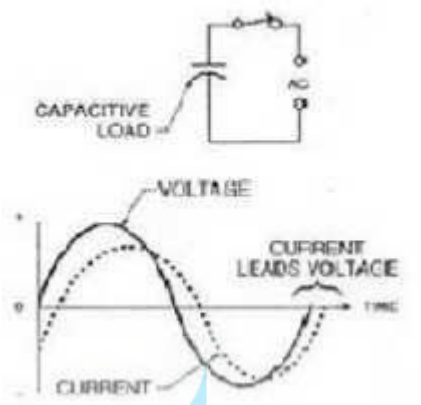
Sumber: Ramdhani, Mohamad. 2008. Rangkaian Listrik. Erlangga : Jakarta

Umumnya beban induktif banyak terdapat pada rangkaian elektronik. Peralatan rumah tangga umumnya menggunakan peralatan elektronik seperti TV, Radio, Kipas angin, Kulkas. Beban induktif dapat menimbulkan fluks magnet. Beban induktif juga dapat mempengaruhi daya reaktif dari suatu rangkaian listrik.

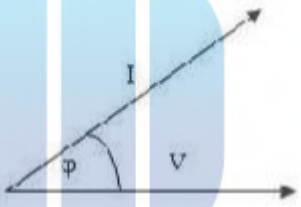
2.3.3 Beban Kapasitif (C)

Beban kapasitif (C) yaitu beban yang memiliki kemampuan kapasitansi atau kemampuan untuk menyimpan energi yang berasal dari pengisian elektrik *(electrical discharge)* pada suatu sirkuit. Komponen ini dapat menyebabkan arus

leading terhadap tegangan. Beban jenis ini menyerap daya aktif dan mengeluarkan daya reaktif.



Gambar 2.5 Gelombang Kapasitif AC



Gambar 2.6 Grafik Arus dan Tegangan Pada Beban Kapasitif

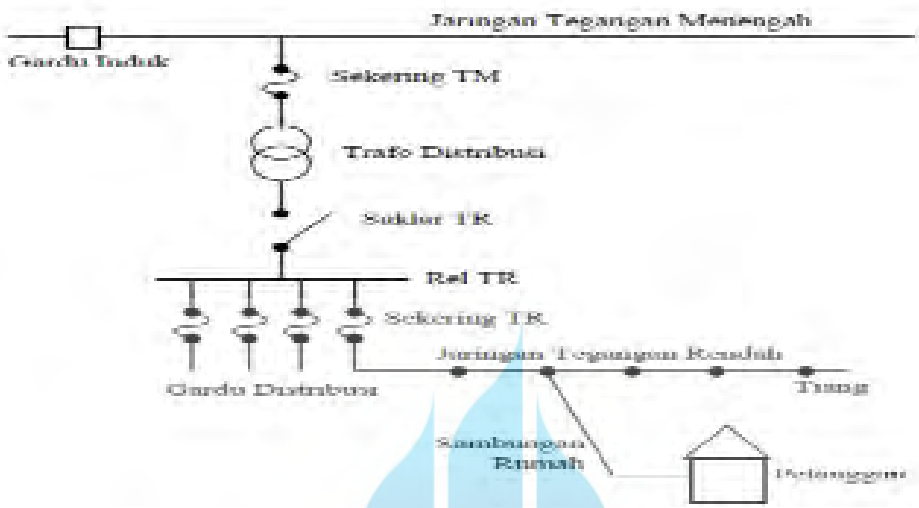
Sumber: Ramdhani, Mohamad. 2008. Rangkaian Listrik. Erlangga : Jakarta

2.4 Pengertian Jaringan Tegangan Rendah

Jaringan Tegangan Rendah (JTR) terdiri dari Saluran Udara Tegangan Rendah (SUTR) dan Saluran Kabel Tegangan Rendah (SKTR), yang berfungsi untuk menyalurkan tenaga listrik dari gardu distribusi ke konsumen. Jaringan Tegangan rendah merupakan bagian dari hilir yang biasa disebut Distribusi Sekunder dengan tegangan operasi nominal 400/230 volt.

Sambungan Tenaga Listrik adalah penghantar di bawah ataupun di atas tanah termasuk peralatannya sebagai bagian instalasi milik PLN yang menghubungkan jaringan tenaga listrik milik PLN dengan instalasi listrik pelanggan untuk menyalurkan tenaga listrik.Dapat juga dikatakan sebagai sambungan pelanggan yang merupakan titik akhir dari pelayanan listrik kepada pelanggan, dengan

tingkat mutu pelayanan yang dapat di lihat dari mutu tegangan dan tingkat kehandalan dari sisi pelayanan tersebut (sesuai SPLN No. 1:1995).



Gambar 2.7 Koneksi Tegangan Menengah ke Tegangan Rendah dan Konsumen (Sumber : Hardiansyah, 2016)

Sambungan Tenaga Listrik Tegangan Rendah (SLTR)

- Pelanggan tegangan rendah Fasa 1 dan dilayani dengan tegangan 220 V.
- Pelanggan tegangan rendah Fasa 3 dan dilayani dengan tegangan 220/380 Volt.

Pada umumnya radius paling jauh pada Jaringan Tegangan Rendah yaitu berkisar 350 meter di Indonesia susut tegangan yang diizinkan yaitu berkisar lebih 5 persen dan kurang 10 persen dari tegangan operasi.

Kontruksi JTR pada Saluran Udara Tegangan Rendah (SUTR) yaitu jaringan kawat yang berisolasi maupun tidak berisolasi. Bagian utama dari SUTR kawat tak berisolasi adalah tiang listrik (besi/beton). Cross Arm, Isolator dan penghantar Alumunium/Tembaga (Cu). Sedangkan kontruksi JTR pada Saluran Kabel Tegangan Rendah (SKUTR) kabel yang digunakan adalah jenis XLPE yang lebih dikenal dengan nama *Low Voltage Twiated Cable*. Jenis kabel ini direntangkan pada tiang penyangga. Bagian utama adalah tiang, kabel dan *suspension clamp bracket* yang berfungsi untuk menahan kabel pada tiang. Kontruksi SUTR ini

sering dijumpai terpasang pada jaringan karena dianggap kontruksi ini lebih handal.

2.5 Pengertian Jaringan Syaraf Tiruan

Jaringan saraf tiruan (JST) (Bahasa Inggris: *artificial neural network* (ANN), atau juga disebut *simulated neural network* (SNN), atau umumnya hanya disebut *neural network* (NN)), adalah jaringan dari sekelompok unit pemroses kecil yang dimodelkan berdasarkan sistem saraf manusia. JST merupakan sistem adaptif yang dapat mengubah strukturnya untuk memecahkan masalah berdasarkan informasi eksternal maupun internal yang mengalir melalui jaringan tersebut. Oleh karena sifatnya yang adaptif, JST juga sering disebut dengan jaringan adaptif (Darmeli, 2015).

Kemajuan teknologi informasi memungkinkan kegiatan peramalan energi listrik dapat dilakukan dengan berbagai metode. Metode jaringan syaraf tiruan perambatan balik dan jaringan syaraf adaptif telah diterapkan pada kegiatan peramalan energi listrik.

Berikut ini adalah tiga tahapan dasar yang terlibat dalam pemuatan jangka pendek peramalan (Isacc, 2019).

1. Pelatihan model

   Jaringan Syaraf Tiruan meniru cara kerja otak manusia. Oleh karena itu, menyiratkan bahwa pelatihan itu penting persyaratan untuk perkiraan yang akurat. Pelatihan dilakukan dengan cara memberi asupan jaringan dengan input yang sesuai dengan yang ditargetkan *output*. Jaringan kemudian disimulasikan dan disesuaikan hingga kesalahan paling sedikit dicapai. Biasanya, sebuah percobaan dan pendekatan kesalahan diadopsi dalam menyesuaikan terhadap metode *learning*, fungsi aktivasi, dan arsitektur jaringan.

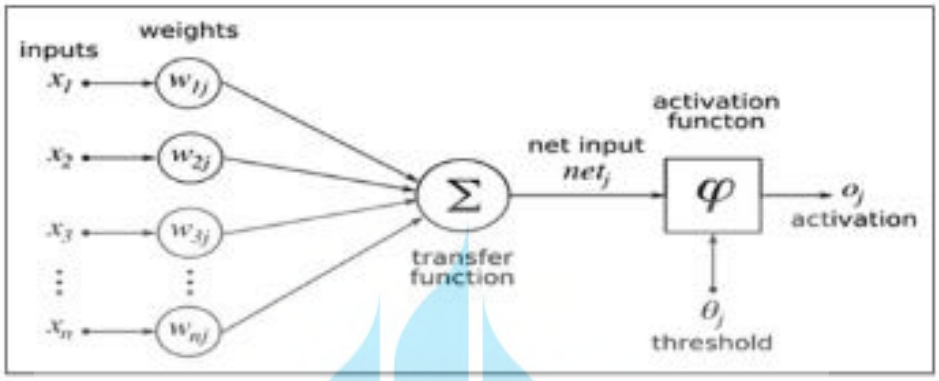
2. Validasi model

   Di sini, *output* dan input yang ditargetkan adalah diperkenalkan ke dalam algoritma yang dikembangkan dan disimulasikan. Perbandingan dibuat

antara *output* yang dihasilkan oleh JST dan *output* yang diinginkan untuk menunjukkan keakuratan model JST.

3. Peramalan dengan model yang terlatih

   Jaringan melakukan prediksi didasarkan pada hubungan yang diamati dari tahap pelatihan. Gambar 2.8 menunjukkan diagram JST.



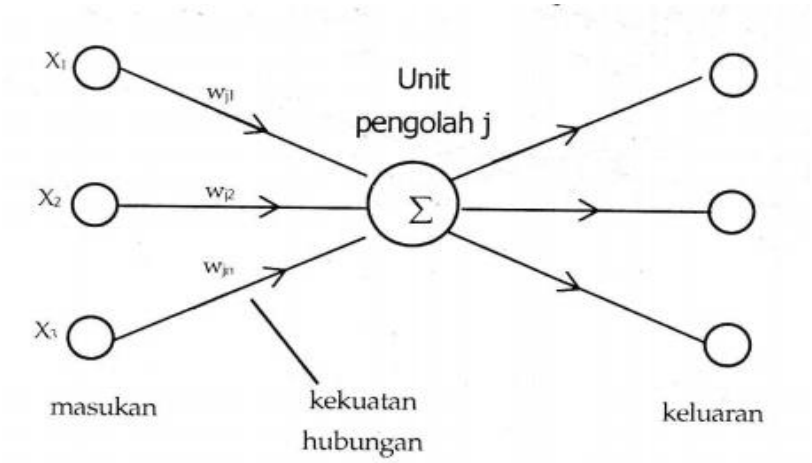
Gambar 2.8 Diagram Jaringan Syaraf Tiruan

Tiga faktor yang mempengaruhi kinerja jaringan syaraf tiruan yaitu sebagai berikut :

1. Pola hubung antara neuron yang disebut arsitektur jaringan.
2. Metode penentuan bobot penghubung atau koneksi yang disebut metode pembelajaran atau algoritma.
3. Fungsi aktivasi yang digunakan

2.5.1 *Neuron*

Neuron adalah unit yang memperoses informasi yang menjadi dasar dalam pengoperasian jaringan syaraf tiruan. Seperti halnya otak manusia, jaringan syaraf tiruan terdiri dari beberapa *neuron* dan ada hubungan antara *neuron-neuron* tersebut. Beberapa *neuron* akan mentransformasikan informasi yang diterima melalui *neuron-neuron* yang lain. Dengan kata lain, *neuron* saraf adalah sebuah unit pemroses informasi yang merupakan dasar operasi jaringan saraf tiruan. Gambar dibawah ini menunjukkan contoh suatu *neuron*.



Gambar 2.9 Struktur Unit Jaringan Syaraf Tiruan

Menurut Laurance (1994) *neuron* terdiri dari 3 elemen pembentuk, yaitu :

1. Himpunan unit-unit yang dihubungkan dengan jalur koneksi. Jalur-jalur tersebut memiliki bobot atau kekuatan yang berbeda-beda. Bobot yang bernilai positif akan memperkuat sinyal dan bobot yang bernilai negatif akan memperlemah sinyal yang dibawa.
2. Suatu unit penjumlah yang akan menjumlahkan input-input sinyal yang sudah dikalikan dengan bobotnya.
3. Fungsi aktivasi yang menentukan keluaran dari sebuah

2.5.2 Metode Pelatihan Jaringan Saraf Tiruan

Cara berlangsungnya pembelajaran atau pelatihan jaringan syaraf tiruan dikelompokkan menjadi 3 yaitu : (Puspitaningrum, 2006).

a. *Supervised Learning* (pembelajaran terawasi).

Pada metode ini, setiap pola yang diberikan kedalam jaringan syaraf tiruan telah diketahui keluarannya. Selisih antara pola *output* aktual (*output* yang dihasilkan) dengan pola *output* yang dikehendaki (*output* target) yang disebut *error* digunakan untuk mengoreksi bobot jaringan syaraf tiruan sehingga jaringan syaraf tiruan mampu menghasilkan *output* sedekat mungkin dengan pola target yang telah diketahui oleh jaringan syaraf tiruan. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah : *Hebbian, Perceptron, ADALINE, Boltzman, Hopfield, Backpropagation*.

b. *Unsupervised Learning* (pembelajaran tak terawasi).

Pada metode ini, tidak memerlukan target *output*. Pada metode ini tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu *range* tertentu tergantung pada nilai input yang diberikan. Tujuan pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok untuk klasifikasi pola. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah : *Competitive, Hebbian, Kohonen, LVQ (Learning Vector Quantization), Neocognitron.*

c. *Hybrid Learning* (pembelajaran hibrida).

Merupakan kombinasi dari metode pembelajaran *Supervised Learning* dan *Unsupervised Learning*, sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah: algoritma RBF. Metode algoritma yang baik dan sesuai dalam melakukan pengenalan pola-pola gambar adalah algoritma *Backpropagation* dan *Perceptron* Untuk mengenali teks berdasarkan tipe *font* akan digunakan algoritma *Backpropagation.*

2.5.3 Fungsi Aktivasi Jaringan Saraf Tiruan

Karakteristik yang harus dimiliki oleh fungsi aktivasi jaringan perambatanbalik antara lain harus kontinyu, terdiferensialkan, dan tidak menurun secara monotonis *(monotonically non-decreasing).* Lebih lanjut, untuk efisiensi komputasi, turunan fungsi tersebut mudah didapatkan dan nilai turunannya dapat dinyatakan dengan fungsi aktivasi itu sendiri (Iwan, 2007).

Beberapa dibawah ini adalah fungsi aktivasi dalam yang digunakan dalam jaringan syaraf tiruan adalah sebagai berikut :

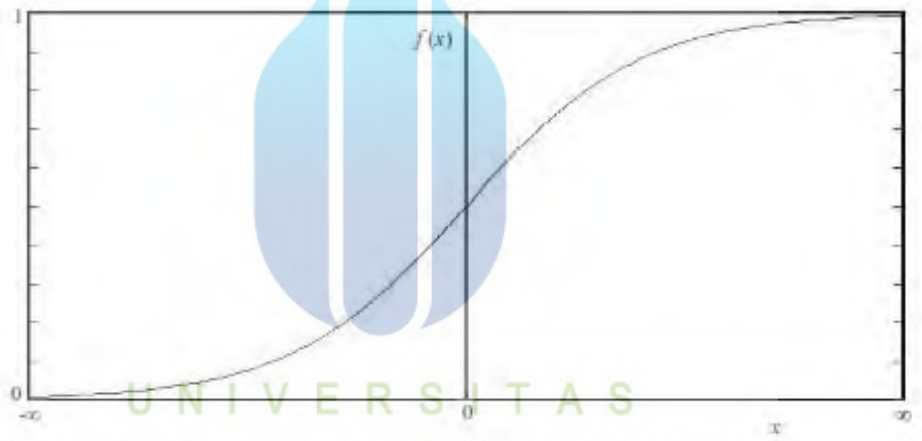
a. Fungsi *sigmoid biner*

Fungsi aktivasi pertama adalah *sigmoid biner* atau *sigmoid logistic*, yang memiliki jangkauan nilai [0,1] ditunjukkan pada Gambar 2.10, didefinisikan sebagai:

$$f(x) = \frac{1}{1+\exp(-\sigma x)} \qquad \text{.....(Persamaan 2.1)}$$

dengan

$$f'(x) = \sigma f(x)[1 - f(x)] \qquad \text{.....(Persamaan 2.2)}$$



Gambar 2.10 Fungsi aktivasi *sigmoid biner*

(Sumber : Fausett, L., 1994)

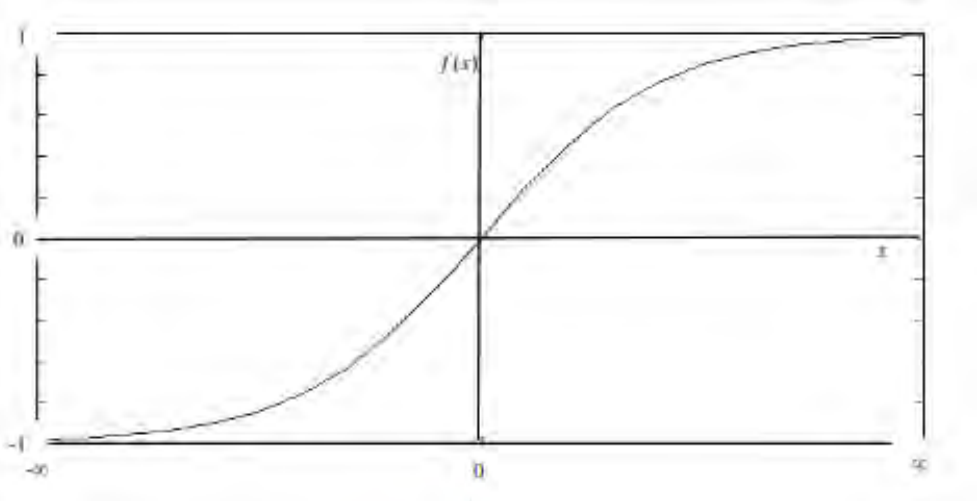
b. Fungsi *sigmoid bipolar*

Fungsi aktivasi kedua adalah *sigmoid bipolar* (Gambar 2.11), yang memiliki jangkauan nilai [-1,1] dan didefinisikan sebagai

$$f(x) = \frac{2}{1+\exp(-\sigma x)} - 1 \qquad \text{.....(Persamaan 2.3)}$$

dengan

$$f'(x) = \frac{\sigma}{2}[1 + f(x)][1 - f(x)] \qquad \text{.....(Persamaan 2.4)}$$



Gambar 2.11 Fungsi aktivasi *sigmoid bipolar*

(Sumber : Fausett, L., 1994)

c. Fungsi Identitas

Digunakan jika keluaran yang dihasilkan oleh jaringan syaraf tiruan merupakan sembarang bilangan riil (bukan hanya pada range [0,1] atau [1,-1]). Y = X

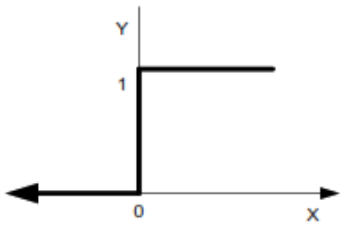
Gambar 2.12 Fungsi aktivasi Identitas

(Sumber : Kusumadewi, 2004)

d. Fungsi *Threshold* (batas ambang).

Fungsi *Threshold* merupakan fungsi *threshold biner*. Untuk kasus bilangan *bipolar*, maka angka 0 diganti dengan angka -1. Adakalanya dalam jaringan syaraf tiruan ditambahkan suatu unit masukkan yang nilainya selalu 1. Unit

tersebut dikenal dengan *bias*. *Bias* dapat dipandang sebagai sebuah input yang nilainya selalu 1. Bias berfungsi untuk mengubah *threshold* menjadi = 0.



Gambar 2.13 Fungsi *Threshold*

(Sumber : Kusumadewi, 2004)

$$F(x)=\begin{cases}1\\0\end{cases}$$ .....(Persamaan 2.5)

Jika x ≥a

Jika x<a

2.5.4 Arsitektur Jaringan Saraf Tiruan

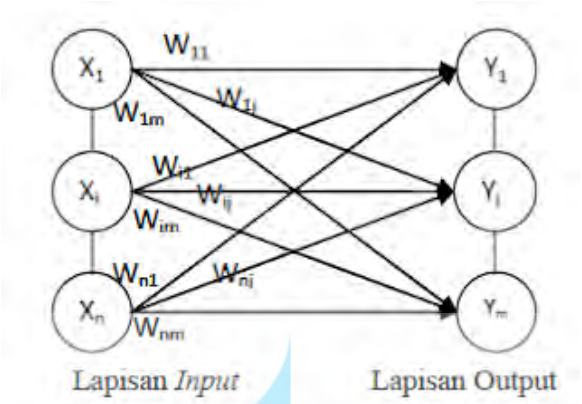
Pada jaringan saraf tiruan, *neuron-neuron* akan dikumpulkan dalam sebuah lapisan yang disebut dengan lapisan *neuron (neuron layers). Neuron-neuron* pada satu lapisan akan dihubungkan dengan lapisan-lapisan lainnya. Informasi yang didapatkan pada sebuah *neuron* akan disampaikan ke semua lapisan-lapisan yang ada, mulai dari lapisan masukan sampai dengan lapisan keluaran melalui lapisan tersembunyi *(hidden layer).* Pada jaringan saraf tiruan ini tiga lapisan bukanlah sebuah struktur umum karena beberapa jaringan saraf ada yang tida memiliki lapisan tersembunyi.

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

a. Jaringan dengan lapisan tunggal *(single layer net)*

Di dalam Jaringan Saraf Tiruan dengan satu *layer*, *neuron-neuron* diorganisasi dalam bentuk *layer-layer*. Dalam bentuk paling sederhana dari Jaringan Saraf Tiruan dengan satu *layer*, kita mempunyai sebuah *input layer* dari *node* sumber

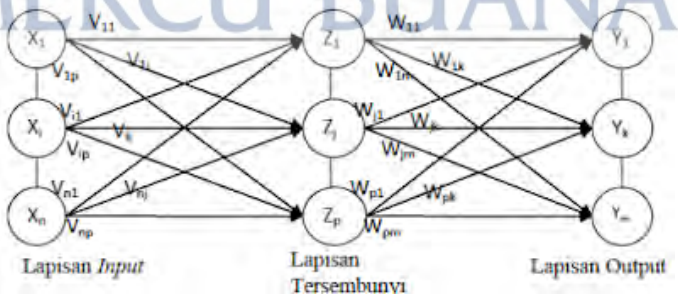
di mana informasi diproyeksikan ke *output layer* dari *neuron* tapi tidak bisa sebaliknya. Dengan kata lain, jaringan ini adalah tipe *feed forward*. *Input layer* dari *node* sumber tidak dihitung karena tidak ada perhitungan yang dilakukan.



Gambar 2.14 Jaringan Dengan Lapisan Tunggal

b. Jaringan dengan banyak lapisan *(multilayer net)*

Merupakan jaringan dengan satu atau lebih lapisan tersembunyi *(hidden layer)*. Jaringan multi lapis ini memiliki kemampuan lebih dalam memecahkan masalah bila dibandingkan dengan jaringan lapis tunggal, namun pelatihannya mungkin lebih rumit. Pada beberapa kasus, pelatihan pada jaringan ini lebih baik karena memungkinkan bagi jaringan untuk memecahkan masalah yang tidak dapat diselesaikan jaringan berlapis tunggal karena jaringan tidak bisa dilatih untuk menampilkan secara benar.

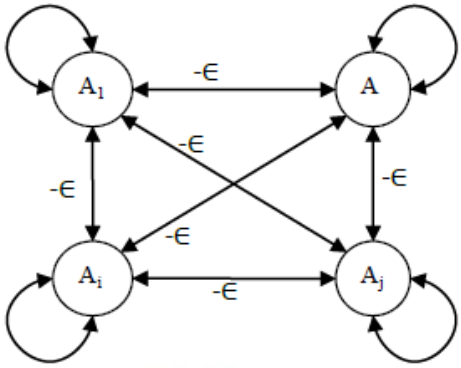


Gambar 2.15 Jaringan Dengan Banyak Lapisan

c. Jaringan dengan lapisan kompetitif *(competitive layer net)*

Bentuk lapisan kompetitif merupakan jaringan saraf tiruan yang sangat besar. Interkoneksi antar *neuron* pada lapisan ini tidak ditunjukkan pada arsitektur seperti jaringan yang lain. Pada jaringan ini sekumpulan *neuron* bersaing untuk

mendapatkan hak menjadi aktif atau sering pula disebut dengan prinsip *winner takes all* atau yang menanglah yang mengambil semua bagiannya.

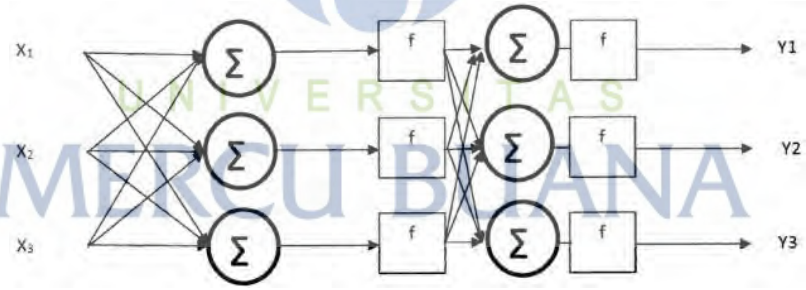


Gambar 2.16 Jaringan Dengan Lapisan Kompetitif

Berdasarkan aliran sinyal masukan, arsitektur jaringan syaraf tiruan dapat diklasifikasikan menjadi 2 kelas, yaitu : (Sri, 2004)

1. Jaringan Umpan Maju *(Feedforward Network)*

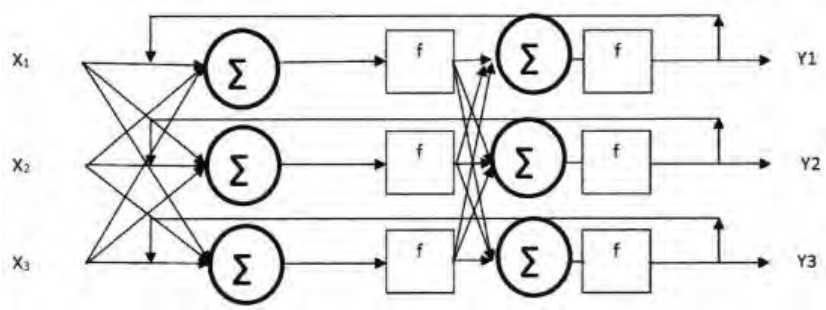
Pada *feedforward network*, sinyal mengalir dari unit *input* ke unit output dalam arah maju. Berikut gambar arsitektur jaringan syaraf tiruan dengan umpan maju



Gambar 2.17 Jaringan Umpan Maju *(Feedforward Network)*

2. Jaringan Umpan Balik *(Recurrent Network)*

Pada jaringan syaraf dengan umpan balik terdapat *neuron output* yang memberikan sinyal pada unit input yang sering disebut dengan *feedback loop*. Berikut gambar arsitektur jaringan syaraf tiruan dengan umpan balik.



Gambar 2.18 Jaringan Umpan Balik *(Recurrent Network)*

2.5.5 *Backpropagation*

Algoritma Perambatan Mundur atau *backpropagation* merupakan algoritma untuk melakukan proses pembelajaran terarah (*supervised learning)* pada jaringan saraf tiruan (JST) untuk mencari beban *(weight)* pada setiap *neuron* yang menghasilkan nilai kesalahan seminimal mungkin melalui data pembelajaran (*training* data) yang diberikan. Metode ini memanfaatkan teknik optimasi berdasarkan penurunan *gradien*. Metode ini dilakukan setelah proses perambatan maju yang merambatkan data dari data masukan ke keluaran melalui koleksi *neuron* dan lapisan JST untuk kemudian dirambatkan balik ke belakang dari lapis keluaran ke lapis masukan untuk menghitung nilai kesalahan pada masing-masing *neuron* dibandingkan dengan nilai keluaran yang seharusnya (nilai target).

Tujuan dari metode *Backpropagation* ini adalah untuk mendapatkan keseimbangan antara kemampuan jaringan untuk mengenali pola yang digunakan selama proses pelatihan berlangsung serta kemampuan jaringan memberikan respon yang benar terhadap pola masukkan yang berbeda dengan pola masukkan pelatihan (Anggi, 2019).

Jaringan syaraf lapis-jamak *(multilayer)* sudah terbukti handal dipakai untuk aplikasi umum. Yang termasuk jaringan lapis-jamak dengan pelatihan terbimbing *(supervised)* antara lain jaringan perambatan-balik *(backpropagation)*. Metode pelatihan perambatan-balik secara sederhana adalah metode *gradient descent* (penurunan *gradien*) untuk meminimalkan total galat kuadrat keluaran. Aplikasi jaringan ini melibatkan pemetaan sekumpulan masukan terhadap

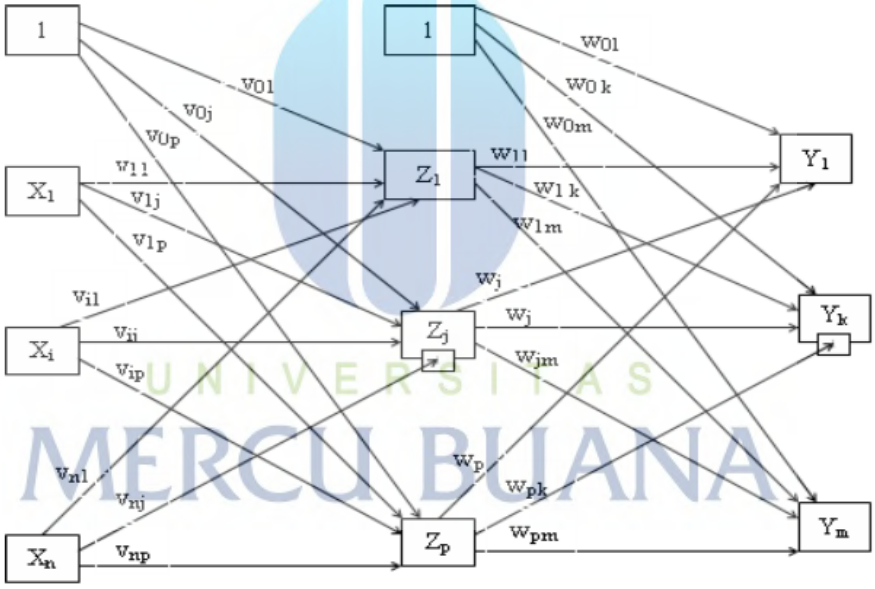
sekumpulan target keluaran, jadi termasuk kategori jaringan dengan pelatihan terbimbing (Iwan, 2007).

Backpropagation adalah metode penurunan gradien untuk meminimalkan kuadrat *error* keluaran. Ada 3 tahap yang dilakukan dalam pelatihan jaringan yaitu adalah sebagai berikut :

- tahap perambatan maju *(forward propagation)* ,
- tahap perambatan-balik,
- tahap perubahan bobot dan bias.

2.5.6 Arsitektur dan Pelatihan Jaringan *Backpropagation*

Arsitektur *backpropagation* memiliki beberapa *layer neuron*. Selain *input layer* dan *output layer*, terdapat *layer* lain yang disebut *hidden layer*.



Gambar 2.19 Arsitektur Jaringan *Backpropagation*

Adapun dibawah ini merupakan beberapa tahapan yang dilakukan selama pelatihan Algoritma *backpropagation* yaitu sebagai berikut :

Tahap Perambatan Maju *(forward propagation)*

a) Setiap unit *input* (Xi, i=1,2,3,...,n) menerima sinyal xi dan meneruskan sinyal tersebut ke semua unit pada lapisan tersembunyi.
b) Setiap unit tersembunyi (Zi, j=1,2,3,...,p) menjumlahkan bobot sinyal input dengan persamaan

$$z_{in_j} = v_{0_j} + \sum_{i=1}^{n} x_i v_{ij}$$

…….(Persamaan 2.6)

Dan menerapkan fungsi aktivasi untuk menghitung sinyal keluaran :

$$z_j = f(z\_in_j)$$

…….(Persamaan 2.7)

biasanya fungsi aktivasi yang digunakan adalah fungsi *sigmoid*. kemudian mengirimkan sinyal tersebut ke semua unit *output*.

c) Setiap unit *output* (Yk, k=1,2,3,...,m) menjumlahkan bobot sinyal *input*

$$y_{in_k} = w_{0_k} + \sum_{i=1}^{p} z_i w_{jk}$$

…….(Persamaan 2.8)

Dan menerapkan fungsi aktivasi untuk menghitung sinyal outputnya :

$$y_k = f(y\_in_k)$$

…….(Persamaan 2.9)

Tahap Perambatan-Balik (Backpropagation)

d) Setiap unit *output* (Yk, k=1,2,3,...,m) menerima pola target yang sesuai dengan pola input pelatihan, kemudian hitung *error* dengan persamaan berikut:

$$\delta k = (tk - yk)\, f'(y\_ink)$$

…….(Persamaan 2.10)

f ' adalah turunan dari fungsi aktivasi kemudian hitung koreksi bobot dengan persaamaan berikut :

$$\Delta wjk = \alpha\, \delta k\, zj$$

…….(Persamaan 2.11)

Dan menghitung koreksi bias dengan persamaan berikut :

$$\Delta w0k = \alpha\, \delta k$$

…….(Persamaan 2.12)

Sekaligus mengirimkan δk ke unit-unit yang ada di lapisan paling kanan.

e) Setiap unit tersembunyi (Zj, j=1,2,3,...,p) menjumlahkan delta inputnya (dari unit-unit yang berada pada lapisan di kanannya):

$$\delta_{in_j} = \sum_{k=1}^{m} \delta_k w_{jk}$$

…….(Persamaan 2.13)

untuk menghitung informasi *error*, kalikan nilai ini dengan turunan dari fungsi aktivasinya:

$$\delta j = \delta\_inj f'(z\_inj)$$ .....(Persamaan 2.14)

kemudian hitung koreksi bobot dengan persamaan berikut :

$$\Delta vjk = \alpha\ \delta j\ xi$$ .....(Persamaan 2.15)

Setelah itu hitung juga koreksi bias dengan persamaan berikut :

$$\Delta v0j = \alpha\ \delta j$$ .....(Persamaan 2.16)

Tahap Perubahan Bobot dan Bias

f) Setiap unit output (Yk, k=1,2,3,...,m) dilakukan perubahan bobot dan bias (j=0,1,2,...,p) dengan persamaan berikut :

$$wjk(baru) = wjk(lama) + \Delta wjk$$ .....(Persamaan 2.17)

Setiap unit tersembunyi (Zj, j=1,2,3,...,p) dilakukan perubahan bobot dan bias (i=0,1,2,...,n) dengan persamaan berikut :

$$vij(baru) = vij(lama) + \Delta vij$$ .....(Persamaan 2.18)

g) Tes kondisi berhenti.

2.6 *Performance Evaluation*

Pengukuran kinerja diperlukan untuk mengevaluasi kinerja prediksi. Ada beberapa cara berbeda untuk mengukur kinerja suatu prediksi, diberikan input model, target dan prediksi. Dua di antaranya adalah *Root Mean Squared Error* (RMSE) dan hit rate. RMSE adalah ukuran penyimpangan nilai prediksi dari nilai target dan didefinisikan sebagai akar kuadrat dari *Mean Squared Error* (MSE).

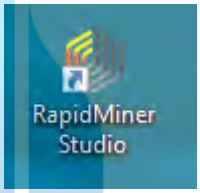
2.6.1 *Root Mean Square Error (RMSE)*

Root Mean Square Error adalah penjumlahan kuadrat error atau selisih antara nilai sebenarnya (aktual) dan nilai prediksi, kemudian membagi jumlah tersebut dengan banyaknya waktu data peramalan dan kemudian menarik akarnya. Jika nilai RMSE semakin kecil maka prediksi model atau variabel tersebut semakin valid. Nilai RMSE dapat dirumuskan sebagai berikut (Pang & Zhao, 2008).

$$RMSE = \sqrt{\frac{\sum(Aktual - Prediksi)^2}{n}}$$ .....(Persamaan 2.19)

2.7 Aplikasi *RapidMiner*

RapidMiner merupakan perangakat lunak yang bersifat terbuka (open source). *RapidMiner* adalah sebuah solusi untuk melakukan analisis terhadap *data mining, text mining* dan analisis prediksi. *RapidMiner* menggunakan berbagai teknik deskriptif dan prediksi dalam memberikan wawasan kepada pengguna sehingga dapat membuat keputusan yang paling baik. *RapidMiner* memiliki kurang lebih 500 operator *data mining*, termasuk operator untuk input, *output*, *data preprocessing* dan visualisasi.



Gambar 2.20 Aplikasi *RapidMiner Studio*

RapidMiner merupakan *software* yang berdiri sendiri untuk analisis data dan sebagai mesin *data mining* yang dapat diintegrasikan pada produknya sendiri. *RapidMiner* ditulis dengan munggunakan bahasa *java* sehingga dapat bekerja disemua sistem operasi.

RapidMiner memiliki beberapa sifat sebagai berikut :

- Ditulis dengan bahasa pemrogaman *Java* sehingga dapat dijalankan di berbagai sistem operasi.
- Proses penemuan pengetahuan dimodelkan sebagai operator *trees*.
- Representasi XML internal untuk memastikan format standar pertukaran data.
- Bahasa *scripting* memungkinkan untuk eksperimen skala besar dan otomatisasi eksperimen.
- Konsep *multilayer* untuk menjamin tampilan data yang efisien dan menjamin penanganan data.

- Memiliki GUI, *command line mode* dan *Java* API yang dapat dipanggil dari program lain.

Bebrapa Fitur dari *RapidMiner*, antara lain :

- Banyaknya algoritma *data mining*, seperti *decision tree* dan *self-organization map*.
- Bentuk grafis yang canggih, seperti tumpang tindih diagram histrogram, *tree chart* dan 3D *Scatter plots*.
- Banyaknya variasi *plugin*, seperti *text plugin* untuk menganalisis teks.
- Menyediakan prosedur *data mining* dan *machine learning* termasuk: ETL (*extraction, transformation, loading*), data *peprocessing*, visualisasi, *modelling* dan evaluasi.
- Proses *data mining* tersusun atas operator-operator yang *nestable*, dideskripsikan dengan XML dan dibuat dengan GUI.
- Mengintegrasikan proyek *data mining Weka* dan statistika R.

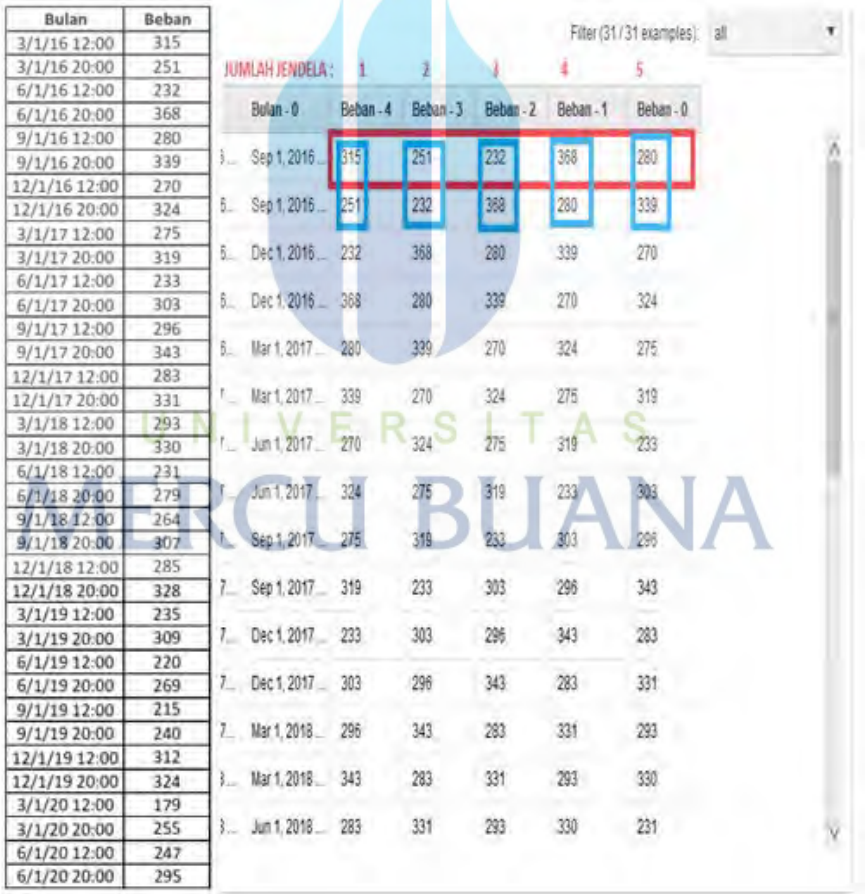
2.8 Operator *Windowing*

Operator *windowing* mengubah seperangkat contoh menjadi satu set contoh bernilai tunggal baru. Parameter terdiri dari *series representation, window size* dan *step size*. *Series representasi* diatur ke nilai default (disandikan seri oleh contoh). *Window size* memberikan lebar jendela atau jumlah contoh yang dipertimbangkan pada suatu waktu. *Step size* pada dasarnya adalah jarak antara pertama nilai-nilai yang pertama. *Step size* diatur ke satu (nilai default) (Thirunavukkarasu, 2016). *Step size* menentukan berapa banyak nilai untuk melangkah maju untuk menetapkan setiap jendela baru.

Seperti yang disebutkan sebelumnya, *Window size* menentukan jumlah titik data yang merupakan input berturut-turut ke model, dimana keluaran selanjutnya nilai akan diprediksi. Ukuran jendela kecil dapat menyebabkan jumlah data input tidak mencukupi menangkap pola dalam data sementara ukuran jendela besar mungkin menghasilkan pola yang terlalu canggih. Besar ukuran jendela juga mengarah pada proses pelatihan yang lebih lama (M. Alamaili, 2011).

2.8.1 Konsep *Windowing*

Algoritma NN akan diuji dengan menggunakan operator *windowing*. Tabel paling sebelah kiri merupakan data asli rentet waktu, dan tabel yang sebelah kanan merupakan data yang ter-*windowing* dengan *window size* adalah 5 (diberi tanda dengan kotak merah) dan *step size* 1 (diberi tanda dengan kotak biru). *Window size* membuat data menjadi lebih banyak, sehingga membuat input NN nya menjadi banyak. Dikarenakan data yang ada hanya 1 kolom, dengan *window size* dibuat beberapa kolom untuk *predict* nilai window ke (n+1), dengan *Window size* bertujuan dari data yang terbatas dapat di training dengan NN, agar NN dapat bekerja optimal karena semakin banyaknya input.



Bulan	Beban
3/1/16 12:00	315
3/1/16 20:00	251
6/1/16 12:00	232
6/1/16 20:00	368
9/1/16 12:00	280
9/1/16 20:00	339
12/1/16 12:00	270
12/1/16 20:00	324
3/1/17 12:00	275
3/1/17 20:00	319
6/1/17 12:00	233
6/1/17 20:00	303
9/1/17 12:00	296
9/1/17 20:00	343
12/1/17 12:00	283
12/1/17 20:00	331
3/1/18 12:00	293
3/1/18 20:00	330
6/1/18 12:00	231
6/1/18 20:00	279
9/1/18 12:00	264
9/1/18 20:00	307
12/1/18 12:00	285
12/1/18 20:00	328
3/1/19 12:00	235
3/1/19 20:00	309
6/1/19 12:00	220
6/1/19 20:00	269
9/1/19 12:00	215
9/1/19 20:00	240
12/1/19 12:00	312
12/1/19 20:00	324
3/1/20 12:00	179
3/1/20 20:00	255
6/1/20 12:00	247
6/1/20 20:00	295

Filter (31 / 31 examples): all

JUMLAH JENDELA : 1 2 3 4 5

	Bulan - 0	Beban - 4	Beban - 3	Beban - 2	Beban - 1	Beban - 0
5...	Sep 1, 2016 ...	315	251	232	368	280
6...	Sep 1, 2016 ...	251	232	368	280	339
6...	Dec 1, 2016 ...	232	368	280	339	270
6...	Dec 1, 2016 ...	368	280	339	270	324
6...	Mar 1, 2017 ...	280	339	270	324	275
7...	Mar 1, 2017 ...	339	270	324	275	319
7...	Jun 1, 2017 ...	270	324	275	319	233
7...	Jun 1, 2017 ...	324	275	319	233	303
7...	Sep 1, 2017 ...	275	319	233	303	296
7...	Sep 1, 2017 ...	319	233	303	296	343
7...	Dec 1, 2017 ...	233	303	296	343	283
7...	Dec 1, 2017 ...	303	296	343	283	331
7...	Mar 1, 2018 ...	296	343	283	331	293
8...	Mar 1, 2018 ...	343	283	331	293	330
8...	Jun 1, 2018 ...	283	331	293	330	231

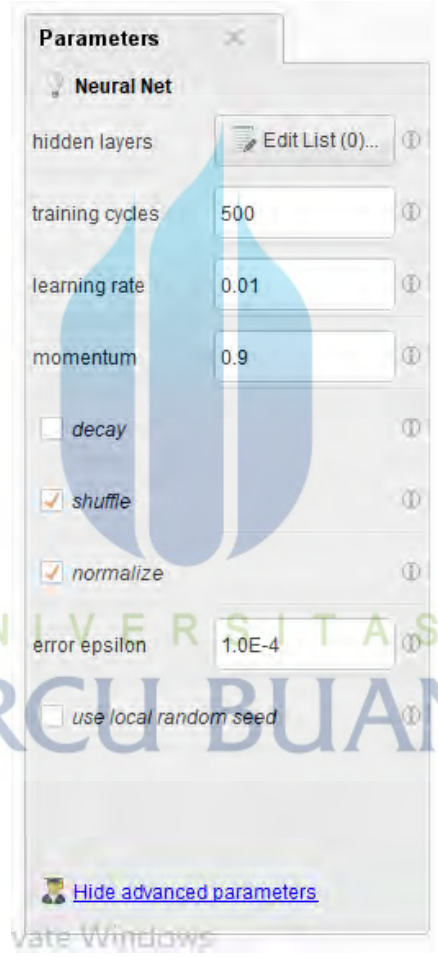
Gambar 2.21 Konsep *Windowing*

Tidak ada aturan khusus untuk menentukan Parameter *Windowing* akan tetapi dampaknya jika *window* kurang maka akan sulit mendapatkan pola,

sedangkan jika *window* terlalu besar polanya akan jadi rumit dan pemrosesannya menjadi lama.

2.9 Operator *Neural Net*

Operator Neural net memiliki beberapa *parameters* yang dapat diubah, pada pengujian ini parameter yang diatur yaitu *training cycles, learning rate* dan *momentum*.



Gambar 2.22 Operator *Windowing*

2.9.1 *Hidden layers*

Parameter ini menjelaskan nama dan ukuran semua lapisan tersembunyi. Pengguna dapat menentukan struktur jaringan saraf dengan parameter ini. Setiap *entri* daftar menggambarkan lapisan tersembunyi baru. Setiap *entri* membutuhkan nama dan ukuran lapisan tersembunyi. Nama *layer* dapat dipilih secara sewenang-

wenang. Ini hanya digunakan untuk menampilkan model. Perhatikan bahwa jumlah aktual *node* akan menjadi satu lebih dari nilai yang ditentukan sebagai ukuran lapisan tersembunyi karena *node* konstan tambahan akan ditambahkan ke setiap lapisan. *Node* ini tidak akan terhubung ke lapisan sebelumnya.

2.9.2 *Training cycles*

Parameter ini menentukan jumlah siklus pelatihan yang digunakan untuk pelatihan jaringan saraf. Dalam *backpropagation* nilai *output* dibandingkan dengan jawaban yang benar untuk menghitung nilai dari beberapa fungsi kesalahan yang telah ditentukan. Kesalahan kemudian diumpankan kembali melalui jaringan. Dengan menggunakan informasi ini, algoritma menyesuaikan bobot setiap koneksi untuk mengurangi nilai fungsi kesalahan dengan jumlah kecil. Proses ini diulangi beberapa kali.

2.9.2 *Learning rate*

Parameter ini menentukan seberapa banyak kita mengubah bobot pada setiap langkah.

2.9.3 *Momentum*

Momentum hanya menambahkan sebagian kecil dari pembaruan berat sebelumnya dengan yang sekarang. Ini mencegah maksimum lokal dan menghaluskan arah optimasi.