BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Sejarah dan Perkembangan Yayasan dan Apotek Mutiara Hati

Apotek dan Praktek Dokter Bersama Mutiara Hati lahir atas prakarsa dua praktisi kesehatan, yaitu dokter dan apoteker. Tujuan berdirinya Apotek Mutiara Hati adalah untuk menyediakan layanan kesehatan terpadu yang berkualitas untuk penduduk Gedebage, khususnya Perumahan Bumi Adipura dan sekitarnya. Alasan dipilihnya daerah ini adalah karena populasinya terus berkembang. Sementara itu, tampaknya fasilitas pelayanan kesehatan di daerah ini masih kurang memadai terutama dalam hal pelayanan oleh dokter spesialis. Data menunjukan bahwa daerah ini baru dilayani oleh satu Puskesmas di daerah Derwati, beberapa dokter umum, beberapa bidan praktek swasta dan dua apotek yang berada dalam radius tiga kilometer satu sama lain. Minimnya fasilitas kesehatan ini tidak hanya dalam jumlah dan jarak, tetapi juga dalam fasilitas pelayanan yang dimiliki.

Apotek dan Praktek Dokter Bersama Mutiara Hati didirikan dengan tujuan untuk memberi kemudahan dan kenyamanan kepada pelanggannya dalam mendapatkan pelayanan kesehatan. Para Pelanggan akan dilayani oleh tenaga-tenaga yang professional di bidangnya. Serta sedapat mungkin dilayani seperti seolah-olah berada di rumah sendiri.

Apotek dan Praktek Dokter Bersama Mutiara Hati bernaung di bawah Yayasan Mutiara Hati, sebuah yayasan yang bergerak di bidang pendidikan, pengembangan sumber daya manusia, dan pelayanan kesehatan. Dalam konteks

pelayanan kesehatan ini, misi yayasan adalah untuk membantu masyarakat mendapatkan pelayanan kesehatan yang memadai, paripurna, terjangkau, dan professional, yang pada akhirnya dapat meningkatkan derajat kesehatan dan kualitas hidup masyarakat. Dalam hal ini, konteks masyarakat adalah masyarakat Gedebage umumnya, khususnya Bumi Adipura dan sekitarnya.

Untuk mendukung tujuan tersebut, maka selain layanan obat resep dan obat bebas, maka terdapat pula layanan praktek dokter spesialis dan laboratorium. Tersedia pula layanan pijat bayi dan senam hamil. Kesemuanya didukung oleh tenaga-tenaga professional untuk memastikan kualitas layanan terbaik bagi pelanggan Apotek dan Praktek Dokter Bersama Mutiara Hati yang tercinta.

Apotek Mutiara Hati berdiri pada tanggal 1 September 2006 berdasarkan Surat Keputusan Kepala Dinas Kesehatan Pemerintah Kota Bandung dengan Nomor Surat Izin Apotek (SIA) 442/407-SIA/5709-DINKES/2006. Berkedudukan di jalan Adi Flora Raya No. 15 Kawasan Pertokoan Perumahan Bumi Adipura Bandung. Dikelola oleh Dr. I Ketut Adnyana M.Si.,Apt sebagai apoteker dengan Surat Izin Kerja (SIK/No/Tgl) No 795/SIK/JB/1995, 27 Maret 1995, dan apoteker pendamping Swastuty Widiasih S.Si.,Apt dengna Surat Izin Kerja (SIK/No/Tgl) No Kp.01.01.V.5.2.20972.

Apotek dan Praktek Dokter Bersama Mutiara Hati Bandung dibawah Yayasan Mutiara Hati yang bergerak dibidang pendidikan, pengembangan sumber daya manusia dan pelayanan kesehatan berdasarkan keputusan Notaris Etic Srimartini, SH.Sp1 Nomor 04 tanggal 21 Juli 2006. Yayasan ini didirikan oleh

professional-professional kesehatan yang kompeten dan memiliki dedikasi dan integritas yang tinggi dibidangnya.

2.1.2 Visi, Misi dan Moto Apotek Mutiara Hati

1. Visi

Menjadi institusi pelayanan kesehatan yang professional, paripurna dan berdaya saing.

2. Misi

a. Secara terus menerus meningkatkan kinerja dan professionalitas pelayanan demi kepuasan pelanggan.

b. Meningkatkan dan mengembangkan segenap potensi agar dapat memperbesar cakupan layanan.

3. Moto

Moto kami adalah MITRA ANDA MENUJU SEHAT yang menunjukan komitmen Apotek Mutiara Hati adalah untuk memberikan pelayanan terbaik kepada pelanggannya.

2.1.1 Lokasi dan Jam Operasi

Apotek Mutiara Hati berlokasi di Kawasan Pertokoan Perumahan Bumi Adipura, tepatnya di jalan Adi Flora Raya Nomor 15, dengan nomor telepon 022-70828486. Apotek ini beroperasi setiap hari Senin – Minggu buka 24 jam sehari.

2.1.2 Pelayanan

1. Layanan Apotek

Apotek ini memberikan pelayanan :

1. Pelayanan obat atas resep dokter

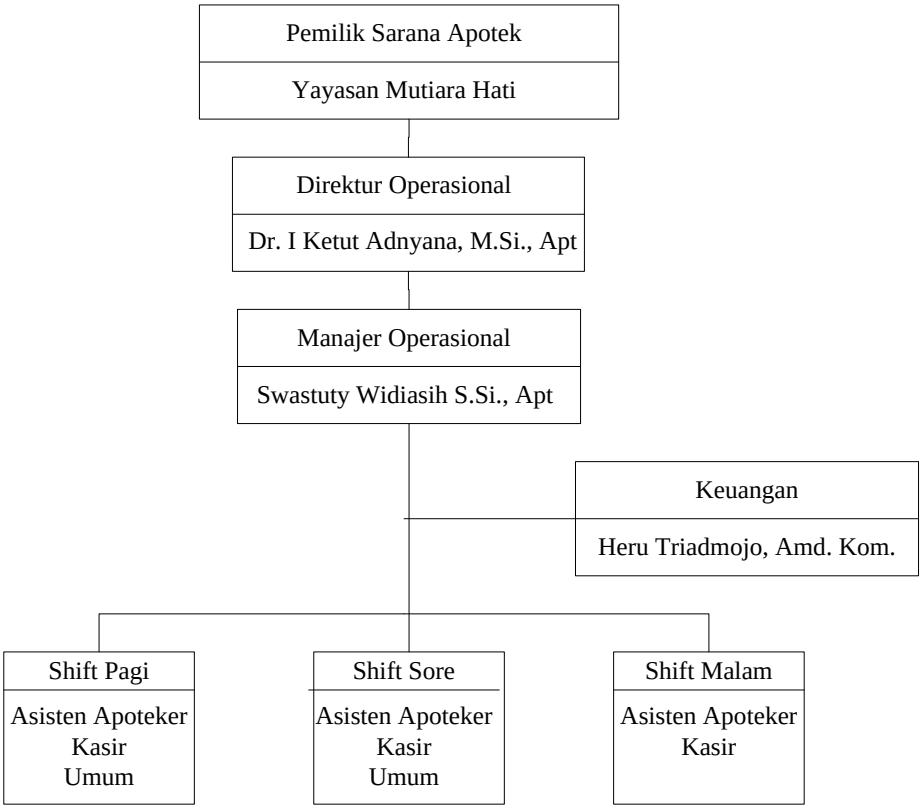
2. Pelayanan obat tanpa resep dokter
3. Memberikan pelayanan konsultasi obat
4. Menyediakan makanan dan minuman kesehatan (suplemen kesehatan, vitamin, dll)
5. Menyediakan produk kesehatan lain (produk perawatan luka, perawatan kulit, perawatan rambut, perawatan anak dan bayi)
6. Menyediakan susu dan produk susu

2. Layanan Praktek Dokter Bersama

Untuk melengkapi kebutuhan masyarakat akan pelayanan kesehatan terpadu, maka Apotek Mutiara Hati melengkapi diri dengan layanan Praktek Dokter Bersama, yang meliputi pelayanan :

1. Dokter Umum
2. Dokter Spesialis Anak
3. Dokter Gigi
4. Dokter Spesialis Kandungan dan Kebidanan

2.1.3 Struktur Organisasi



Gambar 2.1 Struktur Organisasi Apotek Mutiara Hati

Apotek Mutiara Hati dipimpin oleh seorang Direktur Operasional yang membawahi Manajer Operasional. Direktur Operasional bertanggung jawab kepada Pemilik Sarana Apotek yaitu Yayasan Mutiara Hati. Manajer Opersional membawahi bagian keuangan, asisten apoteker, kasir, dan pekerja umum.

2.2 Landasan Teori

2.2.1 Pengertian Sistem

Istilah sistem merupakan suatu istilah yang tidak asing lagi dan banyak digunakan secara luas pada lembaga-lembaga atau bidang-bidang ilmu pengetahuan dan teknologi. Istilah sistem berasal dari bahasa Yunani yaitu Sistema. Ditinjau dari asal katanya, sistem berarti sekumpulan objek yang bekerja

bersama-sama untuk menghasilkan suatu kesatuan metode yang digabungkan dan diatur sedemikian rupa yang berfungsi mencapai tujuan. Suatu sistem dapat terdiri dari beberapa subsistem yang saling berhubungan membentuk suatu kesatuan sehingga sasaran atau tujuan sistem tersebut dapat tercapai. Pada umumnya ciri-ciri sistem bertujuan, punya batas, terbuka, tersusun dari subsistem, saling berkaitan dan saling ketergantungan, merupakan suatu kebulatan yang utuh, melakukan kegiatan transformasi, ada mekanisme kontrol, dan memiliki kemampuan mengatur dan menyesuaikan diri sendiri.

2.2.2 Peramalan

2.2.2.1 Pengertian Peramalan

Peramalan merupakan kegiatan memperkirakan atau memprediksi apa yang terjadi pada waktu yang akan datang, sedangkan rencana merupakan penentuan apa yang akan dilakukan pada waktu yang akan datang. Peramalan menjadi sangat penting karena penyusunan suatu rencana diantaranya didasarkan pada suatu proyeksi atau peramalan.

Peramalan adalah suatu untuk memperkirakan keadaan dimasa yang akan datang melalui pengujian keadaan dimasa lalu. Dalam kehidupan sosial segala sesuatu itu serba tidak pasti, sukar diperkirakan secara tepat. Dalam hal ini perlu diadakan peramalan. Peramalan yang dibuat selalu diupayakan agar dapat meminimumkan pengaruh ketidakpastian ini terhadap sebuah permasalahan. Dengan kata lain peramalan bertujuan mendapatkan peramalan yang bisa meminimumkan kesalahan meramal (*forecat error*) yang biasanya diukur dengan *mean square error, mean absolute error,* dan sebagainya.

Kegunaan peramalan terlihat pada saat pengambilan keputusan. Keputusan yang baik adalah keputusan yang didasarkan atas pertimbangan-pertimbangan yang akan terjadi pada waktu keputusan itu dilaksanakan.

Keberhasilan dari suatu peramalan sangat ditentukan oleh:

a. Pengetahuan teknik tentang pengumpulan informasi (data) masa lalu, data ataupun informasi tersebut bersifat kuantitatif.
b. Teknik dan metode yang tetap dan sesuai dengan pola data yang telah dikumpulkan.

Gambaran perkembangan pada masa lalu dan yang akan datang diperoleh dari hasil analisis data yang didapat dari penelitian yang telah dilakukan. Perkembangan pada masa depan merupakan perkiraan apa yang akan terjadi, sehingga dapat dikatakan bahwa peramalan selalu diperlukan didalam penelitian. Ketepatan penelitian merupakan hal yang penting, walaupun demikian perlu diketahui bahwa sesuatu ramalan selalu ada unsur kesalahannya, sehingga yang perlu diperhatikan adalah usaha untuk memperkecil kesalahan dari ramalan tersebut.

2.2.2.2 Teknik Peramalan

Situasi peramalan sangat beragam dalam horizon waktu peramalan, faktor yang menentukan hasil sebenarnya, tipe pola data dan berbagai aspek lainnya. Untuk menghadapi penggunaan yang luas seperti itu, beberapa teknik telah dikembangkan. Teknik tersebut dibagi dalam dua kategori utama, yaitu:

1. **Peramalan Kuantitatif**

   Peramalan kuantitatif adalah peramalan yang didasarkan atas data kuantitatif masa lalu. Hasil peramalan yang dibuat sangat bergantung pada metode yang dipergunakan dalam peramalan tersebut.

2. **Peramalan Kualitatif atau teknologi**

   Peramalan kualitatif adalah peramalan yang didasarkan atas data kualitatif pada masa lalu. Hasil peramalan yang dibuat sangat bergantung pada orang yang menyusunnya. Hal ini penting karena hasil peramalan tersebut ditentukan berdasarkan pemikiran yang intuisi, pendapat dan pengetahuan serta pengalaman penyusunnya.

Baik tidaknya metode peramalan yang digunakan tergantung dengan perbedaan atau penyimpangan antara hasil ramalan dengan kenyataan yang terjadi. Semakin kecil penyimpangan antara hasil ramalan dengan kenyataan yang akan terjadi maka semakin baik pula metode yang digunakan.

Menurut Makridakis, Whellwright dan McGee (1995:8) peramalan kuantitatif dapat diterapkan bila terdapat kondisi berikut:

a. Tersedia informasi (data) tentang masa lalu.
b. Informasi (data) tersebut dapat dikuantitatifkan dalam bentuk data numerik.
c. Dapat diasumsikan bahwa beberapa aspek pola masa lalu akan terus berlanjut pada masa yang akan datang.

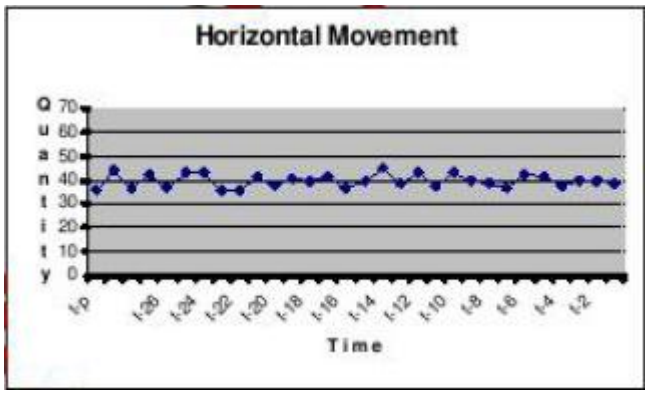
Peramalan kuantitatif dapat dibagi ke dalam deret berkala (*time* series) dan mertode kausal. Pada jenis deret berkala, pendugaan masa depan dilakukan berdasarkan nilai masa lalu dari suatu variabel dan/atau kesalahan masa lalu.

Tujuan metode peramalan itu adalah menemukan pola dalam deret data *historis* dan mengekstrapolasikan pola dalam menemukan pola didalam pola deret data *historis* dan mengekstrapolasikan data tersebut ke masa depan. Sedangkan model kausal mengasumsikan bahwa faktor yang diramalkan menunjukan suatu hubungan sebab-akibat dengan satu atau lebih variabel bebas.

Langkah penting dalam memilih suatu metode deret berkala yang tepat adalah dengan mempertimbangkan jenis pola data, sehingga metode yang paling tepat dengan pola tersebut dapat diuji. Menurut Makridakis, Whellwright dan McGee (195:10) pola dapat dibedakan menjadi empat jenis siklis (*crylical*) dan *trend* yaitu:

a. Pola Horizontal (H)

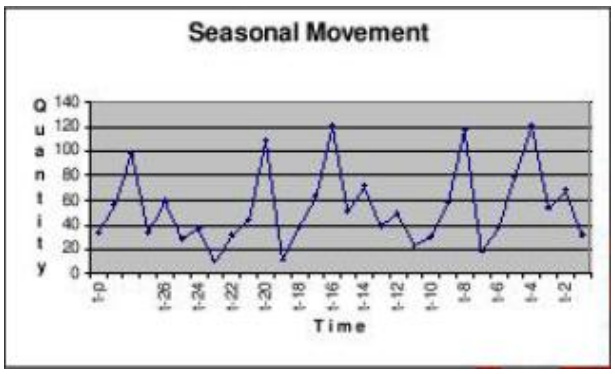
Pola horizontal terjadi bilamana nilai data berfluktuasi di sekitar nilai rata-rata konstan. Suatu produk yang penjualannya tidak meningkat atau menurun selama waktu tertentu termasuk jenis ini. Gambar 2.2 menunjukan pola khas horizontal.



Gambar 2.2 Pola Horizontal

b. Pola Musiman (S)

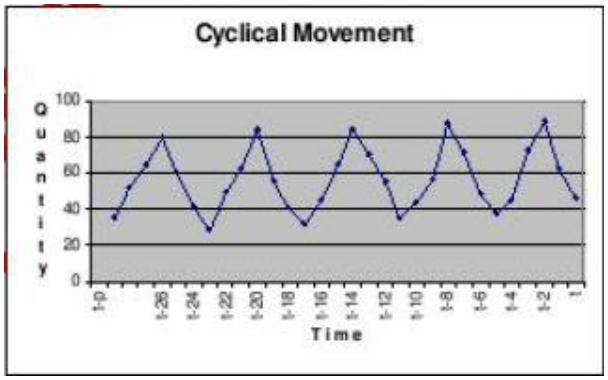
Pola musiman terjadi bilamana suatu deret dipengaruhi oleh faktor musiman (misalnya kuartal tahun tertentu, bulan atau harian). Penjualan dari produk seperti minuman ringan, es krim dan bahan bakar pemanas ruang, semuanya menunjukan jenis pola ini. Gambar 2.3 menunjukan pola khas dari data musiman.



Gambar 2.3 Pola Musiman

c. Pola Siklis (C)

Pola siklis terjadi bilamana datanya dipengaruhi oleh fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis seperti mobil, baja dan peralatan utama lainnya. Penjualan produk seperti mobil, baja dan peralatan utama lainnya menunjukan jenis pola ini, dapat dilihat pada Gambar 2.4.



Gambar 2.4 Pola Siklis

d. Pola Trend (T)

Pola trend terjadi bilamana terdapat kenaikan atau penurunan sekuler jangka panjang dalam data. Penjualan banyak perusahaan, produk bruto nasional, dan berbagai indikator bisnis atau ekonomi lainnya mengikuti pola trend selama perubahannya sepanjang waktu. Gambar 2.5 menunjukan pola trend.



Gambar 2.5 Pola Trend

2.2.2.3 Ukuran Ketepatan Metode Peramalan

Dalam pemodelan deret berkala, sebagian data yang diketahui dapat digunakan untuk meramalkan data berikutnya. Selisih besaran (ukuran kesalahan peramalan) data peramalan terhadap data aktual yang terjadi merupakan suatu data penting untuk menilai ketepatan suatu metode peramalan. Dengan membandingkan ukuran kesalahan beberapa metode peramalan, akan diperoleh metode mana yang mempunyai ukuran kesalahan terkecil, sehingga nilai peramalan dapat dipakai sebagai acuan dalam menentukan kebutuhan-kebutuhan dimasa yang akan datang.

Menurut Makridakis, Whellwright dan McGee (1995:32,43) terdapat beberapa ukuran kesalahan dalam peramalan antara lain RMSE.

a. Nilai tengah kesalahan akar kuadrat atau *Root Mean Squared Error* (RMSE) dengan persamaan :

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(X_t - F_t)^2}{n}} \quad (2.1)$$

RMSE merupakan metode alternatif untuk mengevaluasi teknik peramalan. RMSE adalah rata-rata kuadrat dari perbedaan nilai estimasi dengan nilaiobservasi suatu variabel. Jika nilai RMSE semakin kecil maka estimasi model atau variabel tersebut semakin valid.

Notasi dasar dari persamaan di atas secara ringkas adalah sebagai berikut :

X_t = nilai aktual pada periode *t*

F_t = nilai peramalan pada periode *t*

X_t - F_t = nilai kesalahan peramalan

n = jumlah data

2.2.3 Kecerdasan Buatan

Kecerdasan Buatan atau *Artificial Intelligence* didefinisikan sebagai kecerdasan yang ditunjukkan oleh suatu entitas buatan. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Kecerdasan buatan ini merupakan cabang dari ilmu komputer yang *concerned* dengan pengotomatisasi tingkah laku cerdas. Karena itu kecerdasan buatan harus didasarkan pada prinsip-prinsip teoretikal dan terapan yang menyangkut struktur data yang digunakan dalam representasi pengetahuan (*knowledge representation*), algoritma yang diperlukan dalam penerapan pengetahuan itu, serta bahasa dan teknik pemrograman yang dipakai dalam

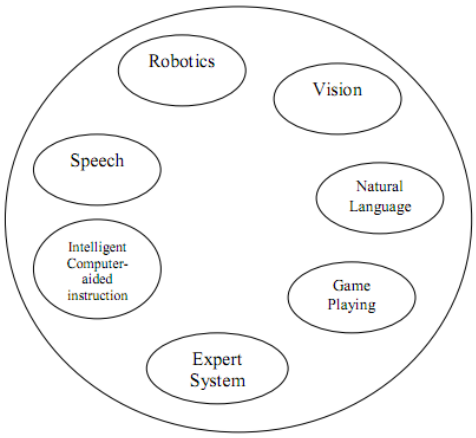
implementasinya. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan zaman, maka peran komputer semakin mendominasi kehidupan umat manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia.

2.2.4 Lingkup Kecerdasan Buatan

Makin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran kecerdasan buatan. Karakteristik ‘cerdas’ sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi. Kecerdasan buatan tidak hanya dominan di bidang ilmu komputer (informatika), namun juga sudah merambah di berbagai disiplin ilmu yang lain. Irisan antara psikologi dan kecerdasan buatan melahirkan sebuah area yang dikenal dengan nama *Cognition & Psycolinguistics*. Irisan antara teknik elektro dengan kecerdasan buatan melahirkan berbagai ilmu seperti: pengolahan citra, teori kendali, pengenalan pola dan robotika.

Adanya irisan penggunaan kecerdasan buatan di berbagai disiplin ilmu tersebut menyebabkan cukup rumitya untuk mengklasifikasikan kecerdasan buatan menurut disiplin ilmu yang menggunakannya. Untuk memudahkan hal tersebut, maka pengklasifikasian lingkup kecerdasan buatan didasarkan pada *output* yang diberikan yaitu aplikasi komersial (meskipun sebenarnya kecerdasan buatan itu sendiri bukan merupakan medan komersial). Lingkup utama dalam kecerdasan buatan adalah:

1. Sistem Pakar (*Expert System*). Disini komputer digunakan sebagai sarana untuk menyimpan pengetahuan para pakar. Dengan demikian komputer akan memiliki keahlian untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar.
2. Pengolahan Bahasa Alami (*Natural Language Processing*). Dengan pengolahan bahasa alami ini diharapkan user dapat berkomunikasi dengan komputer dengan menggunakan bahasa sehari-hari.
3. Pengenalan Ucapan (*Speech Recognition*). Melalui pengenalan ucapan diharapkan manusia dapat berkomunikasi dengan komputer dengan menggunakan suara.
4. Robotika & Sistem Sensor (*Robotics & Sensory Syste*ms).
5. *Computer Vision*, mencoba untuk dapat menginterpretasikan gambar atau objek-objek tampak melalui komputer.
6. *Intelligent Computer-Aided Instruction*. Komputer dapat digunakan sebagai tutor yang dapat melatih dan mengajar.
7. *Game Playing*.



Gambar 2.6 Lingkup Kecerdasan Buatan

Metodologi-metodologi yang digunakan dalam *Soft Computing* adalah :

1. Sistem Fuzzy (mengakomodasi ketidaktepatan); Logika Fuzzy (*fuzzy logic*)
2. Jaringan Syaraf (menggunakan pembelajaran); Jaringan Syaraf Tiruan (*neural network*)
3. *Probabilistic Reasoning* (mengakomodasi ketidakpastian)
4. *Evolutionary Computing* (optimasi); Algoritma Genetika

2.2.5 Logika Fuzzy

2.2.5.1 Pengertian Logika Fuzzy

Fuzzy secara bahasa diartikan sebagai kabur atau samar-samar. Suatu nilai dapat bernilai besar atau salah secara bersamaan. Dalam Fuzzy dikenal derajat keanggotaan yang memiliki rentang nilai 0 (nol) hingga 1(satu). Berbeda dengan himpunan tegas yang memiliki nilai 1 atau 0 (ya atau tidak).

Logika Fuzzy merupakan seuatu logika yang memiliki nilai kekaburan atau kesamaran (Fuzzy*ness*) antara benar atau salah. Dalam teori logika fuzzy suatu nilai bias bernilai benar atau salah secara bersama. Namun berapa besar keberadaan dan kesalahan suatu tergantung pada bobot keanggotaan yang dimilikinya. Logika Fuzzy memiliki derajat keanggotaan dalam rentang 0 hingga 1. Berbeda dengan logika digital yang hanya memiliki dua nilai 1 atau 0. Logika Fuzzy digunakan untuk menterjemahkan suatu besaran yang diekspresikan menggunakan bahasa (*linguistic*), misalkan besaran kecepatan laju kendaraan yang diekspresikan dengan pelan, agak cepat, cepat, dan sangat cepat. Dan Logika Fuzzy menunjukan sejauh mana suatu nilai itu benar dan sejauh mana suatu nilai itu salah. Tidak seperti Logika klasik *crisp/*tegas, suatu nilai hanya mempunyai 2

kemungkinan yaitu merupakan suatu anggota himpunan atau tidak. Derajat keanggotaan 0 (nol) artinya nilai bukan merupakan anggota himpunan dan 1 (satu) berarti nilai tersebut adalah anggota himpunan.

Logika Fuzzy adalah suatu cara yang tepat untuk memetakan suatu ruang *input* kedalam suatu ruang *output*, mempunyai nilai kontinyu. Fuzzy dinyatakan dalam derajat dari suatu keanggotaan dan derajat dari kebenaran. Oleh sebab itu sesuatu dapat dikatakan sebagian benar dan sebagian salah pada waktu yang sama (Kusumadewi. 2010).

Logika Fuzzy memungkinkan nilai keanggotaan antara 0 dan 1, tingkat keabuan dan juga hitam dan putih, dan dalam bentuk *linguistik*, konsep tidak pasti seperti "sedikit", "lumayan" dan "sangat" (Zadeh 1965). Kelebihan dari teori logika fuzzy adalah kemampuan dalam proses penalaran secara bahasa (*linguistic reasoning*). Sehingga dalam perancangannya tidak memerlukan persamaan matematik dari objek yang akan dikendalikan.

2.2.5.2 Himpunan Fuzzy

Dalam teori logika Fuzzy dikenal himpunan Fuzzy *(*Fuzzy *set)* yang merupakan pengelompokan sesuatu berdasarkan variabel bahasa (*linguistic variable*), yang dinyatakan dalam fungsi keanggotaan. Didalam semesta pembicaraan (*universe of discourse*) U, fungsi keanggotaan dari suatu himpunan Fuzzy tersebut bernilai 0 sampai dengan 1.

2.2.5.3 Fungsi Keanggotaan

Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukan pemetaan titik-titik *input* data ke dalam nilai keanggotaan (disebut juga derajat keanggotaan) yang memiliki interval antara 0 sampai 1.

Fungsi keanggotaan yang digunakan Jang, J.–S.R, Sun, C.-T dan Mitsuzani, E., 1997:24-26), adalah sebagai berikut:

1. Fungsi keanggotaan Segitiga

   Fungsi keanggotaan yang mempunyai tiga parameter {a,b,c} dengan formulasi :

   $$segitiga(x; a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (2.3)$$

2. Fungsi Keanggotaan Trapesium

   Fungsi keanggotaan yang mempunyai empat parameter {a,b,c,d} dengan formulasi :

   $$trapesium(x; a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \quad (2.4)$$

3. Fungsi Keanggotaan Gaussian

   Fungsi keanggotaan yang mempunyai dua parameter {c,σ} dengan formulasi :

   $$gaussian(x; \sigma, c) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (2.5)$$

4. Fungsi Keanggotaan Bell yang diperluas.

   Fungsi kenaggotaan yang mempunyai tiga parameter {a,b,c} dengan formulasi bell sebagai berikut:

   $$bell(x; a, b, c) = \frac{1}{\left(1+\left|\frac{x-c}{a}\right|^{2b}\right)} \quad (2.6)$$

c dan a merupakan pusat dan lebar dari MF, b untuk mengendalikan lereng pada titik-titik crossover. Dimana parameter b bernilai positif. Jika b bernilai negatif fungsi keanggotaan menjadi fungsi keanggotaan bell terbalik.

2.2.5.4 Turunan dari Parameter Fungsi Keanggotaan

Untuk menghasilkan suatu sistem fuzzy yang adaptif, diperlukan adanya turunan dari fungsi keanggotaan yang digunakan berdasarkan *input* dan parameter fungsi keanggotaan (Jang, J.–S.R, Sun, C.-T dan Mitsuzani, E., 1997:34). Turunan dari fungsi keanggotaan bell dapat dijabarkan sebagai berikut:

$$y = bell(x; a, b, c) = \frac{1}{\left(1+\left|\frac{x-c}{a}\right|^{2b}\right)} \tag{2.7}$$

$$\frac{dy}{dx} = \begin{cases} -\frac{2b}{x-c}y(1-y), & \text{if x} \neq \text{c} \\ 0, & \text{if x} = \text{c} \end{cases} \tag{2.8}$$

$$\frac{dy}{da} = \frac{2b}{a}y(1-y) \tag{2.9}$$

$$\frac{dy}{db} = \begin{cases} -2ln\left|\frac{x-c}{a}\right| y(1-y), & \text{if x} \neq \text{c} \\ 0, & \text{if x} = \text{c} \end{cases} \tag{2.10}$$

$$\frac{dy}{dc} = \begin{cases} \frac{2b}{x-c}y(1-y), & \text{if x} \neq \text{c} \\ 0, & \text{if x} = \text{c} \end{cases} \tag{2.11}$$

2.2.5.5 Operator-operator Fuzzy

Pada dasarnya ada 2 model operator fuzzy, yaitu operator-operator dasar yang dikemukaan oleh Zadeh dan operator-operator alternatif yang dikembangkan dengan menggunakan konsep transformasi tertentu.

a. Operator-operator Dasar Zadeh

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasi dan memodifikasi himpunan

fuzzy. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau α-predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh, yaitu AND, OR dan NOT (Kusumadewi,2010).

1. Operator AND

Operator ini berhubungan dengan operasi interaksi pada himpunan. α-predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan (Cox, 1994).

$$\mu_{A\cap B} = \min(\mu_{A(X)}, \mu_{B(Y)}) \tag{2.12}$$

2. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan. α-predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan (Cox, 1994).

$$\mu_{A\cup B} = \max(\mu_{A(X)}, \mu_{B(Y)}) \tag{2.13}$$

3. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan. α-predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai elemen pada himpunan yang bersangkutan dari 1 (Cox, 1994).

$$\mu_{A}{}' = 1 - \mu_{A(X)} \tag{2.14}$$

b. Operator-operator Alternatif

Pada dasarnya, ada 2 tipe operator alternatif, yaitu operator alternatif yang didasarkan pada transformasi aritmatika, seperti : *mean, product, dan bounded suml*; dan operator alternatif yang didasarkan pada transformasi fungsi yang lebih kompleks, seperti : Kelas Yager dan Sugeno.

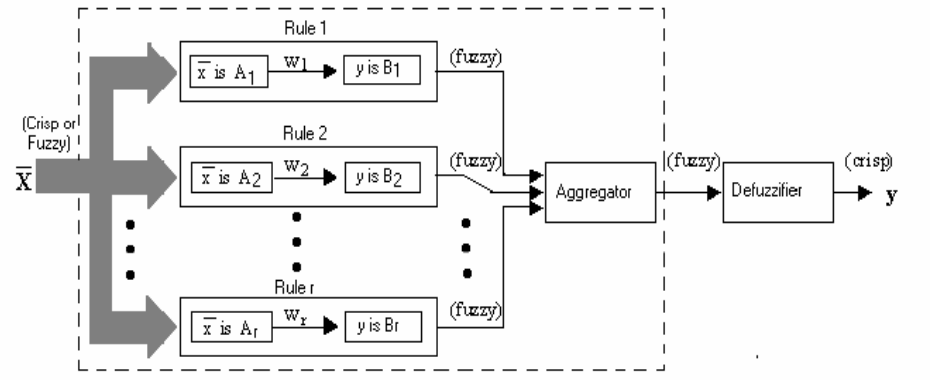
2.2.5.6 Sistem Inferensi Fuzzy

Sistem inferensi fuzzy adalah sebuah kerangka kerja perhitungan yang berdasar pada konsep teori himpunan fuzzy, aturan fuzzy If-Then, dan pemikiran fuzzy. Sistem inferensi fuzzy ini telah berhasil di aplikasikan pada berbagai bidang, seperti kontrol otomatis, klasifikasi data, analisis keputusan, sistem pakar, prediksi *time series*, robotika dan pengenalan pola. Sistem inferensi fuzzy juga dikenal dengan berbagai nama seperti *fuzzy rule based system* (sistem berbasis aturan fuzzy), *fuzzy expert system* (sistem pakar fuzzy), fuzzy model, *fuzzy associative memory, fuzzy logic controler* (pengendali logika fuzzy) dan sistem fuzzy sederhana.

Struktur dasar dari sistem inferensi fuzzy berisi tiga komponen konseptual:

1. Dasar aturan yang mana berisi sebuah pemilihan aturan fuzzy.
2. *Database* yang mendefinisikan fungsi keanggotaan yang digunakan dalam aturan fuzzy.
3. Mekanisme pemikiran yang mengerjakan prosedur inferensi terhadap aturan dan kenyataan yang diketahui untuk menurunkan *output* atau kesimpulan yang masuk akal.

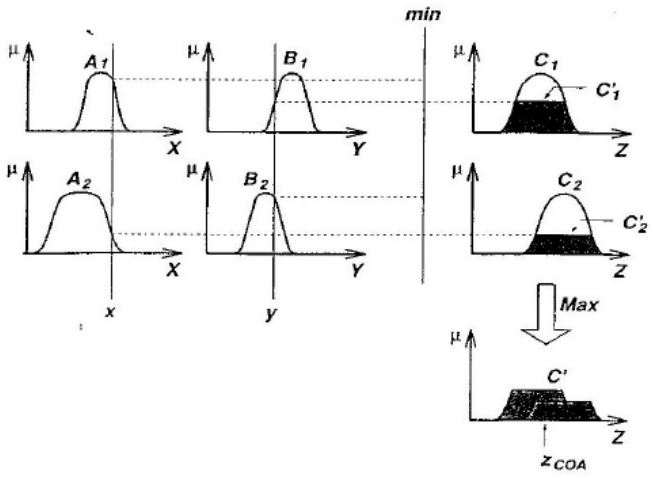
Sistem inferensi fuzzy dapat mengambil *input* fuzzy ataupun *crisp*, tetapi *output*nya hampir selalu menghasilkan himpunan fuzzy. Oleh karena itu, diperlukan suatu metode defuzzifikasi untuk mendapatkan nilai *crisp*.



Gambar 2.7 Sistem Inferensi Fuzzy

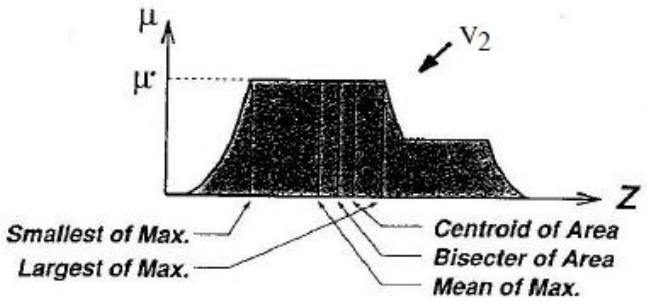
a. Model Fuzzy Mamdani

Sistem inferensi fuzzy mamdani diusulkan sebagai usaha awal untuk mengendalikan mesin uap dan kombinasi boiler dengan sebuah himpunan aturan kendali linguistik yang diperoleh dari pengalaman operator manusia. Gambar 2.8 mengilustrasikan bagaimana dua aturan sistem inferensi mamdani menurunkan semua *output* z ketika ditunjuk oleh dua *input crisp* x dan y.



Gambar 2.8 Sistem Inferensi Fuzzy Mamdani

Defuzzifikasi mengacu pada cara nilai *crisp* diekstrak dari sebuah himpunan fuzzy sebagai nilai representatif. Pada umumnya, ada 5 metode untuk defuzzifikasi sebuah himpunan fuzzy A dari semesta Z. Berikut ini penjelasan masing-masing strategi defuzzifikasi.



Gambar 2.9 Defuzzifikasi dari sistem inferensi fuzzy mamdani

- *Centroid of area* z_{COA} :

$$z_{COA} = \frac{\int_Z \mu_A(z) z dz}{\int_Z \mu_A(z) dz} \tag{2.15}$$

dimana μA (z) adalah *output* MF teragregasi.

- *Bisector of area* z_{BOA} :

$$\int_{\alpha}^{z_{BOA}} \mu_A(z) dz = \int_{z_{BOA}}^{\beta} \mu_A(z) dz \tag{2.16}$$

dimana $\alpha = \min\{z \mid z \in Z\}$ dan $\beta = \max\{z \mid z \in Z\}$. $z = z_{BOA}$ membagi daerah antara $z = \alpha$, $z = \beta$, $y = 0$ dan $y = \mu_A(z)$ ke dalam dua daerah yang sama.

- *Mean of maximum* z_{MOM} :

z_{MOM} adalah rata-rata dari maksimalisasi z pada MF yang mencapai maksimum μ^*.

$$z_{MOM} = \frac{\int_{Z'} z dz}{\int_{Z'} dz} \tag{2.17}$$

- *Smallest of maximum* z_{SOM} :

  z_{SOM} adalah minimum dari maksimisasi z.

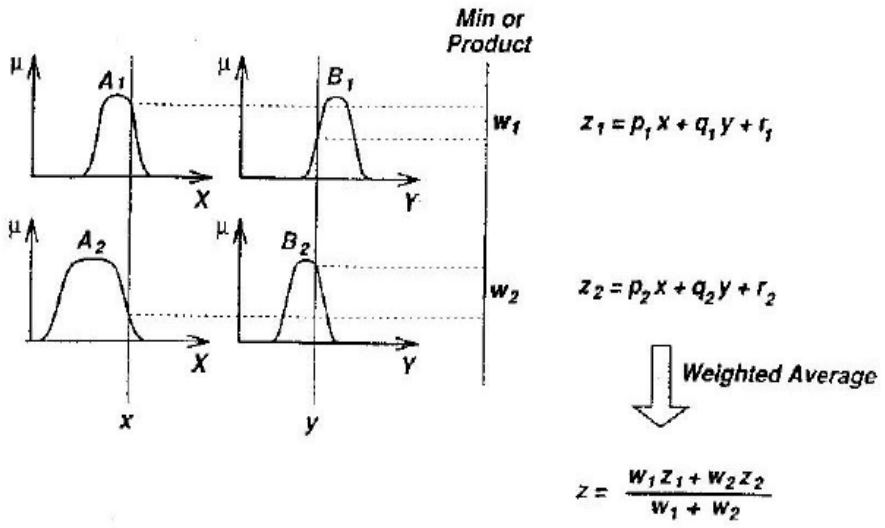
- *Largest of maximum* z_{LOM} :

  z_{LOM} adalah maksimum dari maksimisasi z.

b. Model Fuzzy Sugeno

Model fuzzy Sugeno diusulkan oleh Takagi, Sugeno dan Kang dalam usaha membangun pendekatan sistematis untuk meng-*generate* aturan fuzzy dari data set *input-output* yang diberikan. Aturan fuzzy tipikal dalam sebuah model fuzzy Sugeno berbentuk :

Jika x adalah A dan y adalah B maka z = f(x,y)



Gambar 2.10 Sistem Inferensi Fuzzy Sugeno

Dimana A dan B adalah himpunan fuzzy dalam *antecedent*, sedangkan z = f(x,y) adalah fungsi *crisp* dalam *consequent*. Biasanya f(x,y) adalah sebuah

polinomial dalam variabel *input* x dan y, tetapi ini dapat menjadi suatu fungsi selama dapat menjelaskan *output* model dalam daerah fuzzy yang telah ditentukan oleh aturan *antecedent* secara sesuai. Ketika f(x,y) adalah polinomial orde satu, menghasilkan sistem inferensi fuzzy disebut model fuzzy Sugeno orde satu. Ketika f adalah konstan, disebut model fuzzy Sugeno orde nol.

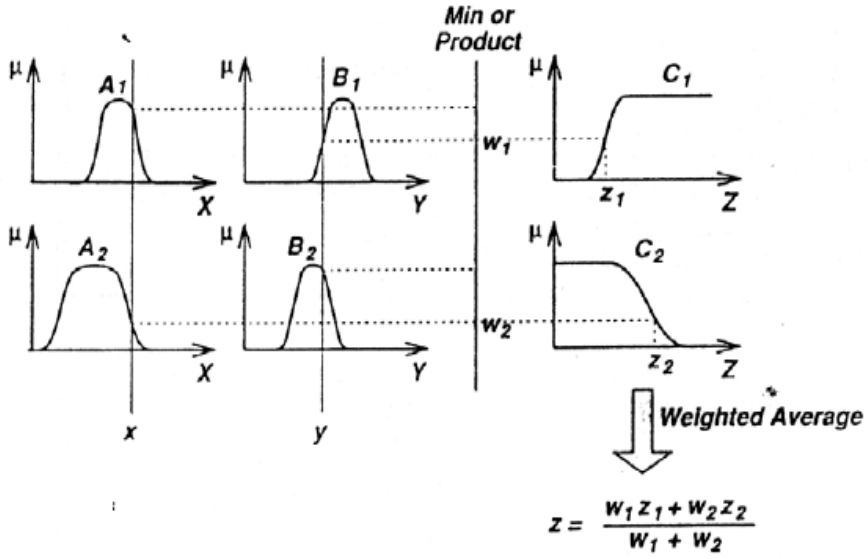
Untuk 2 aturan pada basis aturan model Sugeno akan berbentuk :

If x_1 is A_1 and x_2 is B_1 Then $y_1 = c_{11}x_1 + c_{12}x_2 + c_{10}$

If x_1 is A_2 and x_2 is B_2 Then $y_2 = c_{21}x_1 + c_{22}x_2 + c_{20}$

c. **Model Fuzzy Tsukamoto**

Dalam model fuzzy Tsukamoto, *consequent* dari masing-masing aturan fuzzy If-Then direpresentasikan oleh satu set fuzzy dengan MF monoton. Sebagai hasilnya *output* yang terinferensi dari masing-masing aturan didefinisikan sebagai nilai *crisp* diinduksikan oleh aturan *firing strength*. *Output* keseluruhan diambilkan sebagai rata-rata terbobot dari tiap aturan *output*.

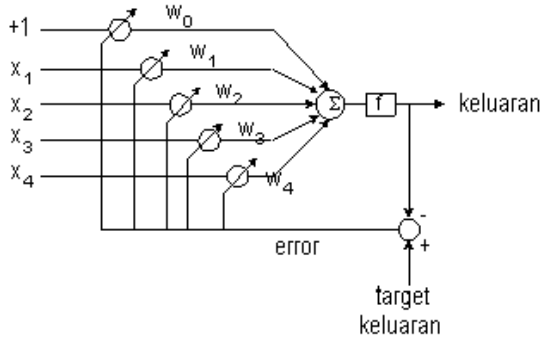


Gambar 2.11 Sistem Inferensi Fuzzy Tsukamoto

2.2.6 Jaringan Syaraf Tiruan

2.2.6.1 Gambaran Umum Jaringan Syaraf Tiruan

Jaringan syaraf tiruan merupakan algoritma komputasi yang meniru cara kerja sel syaraf. Semua sinyal yang masuk dikalikan dengan bobot yang ada pada tiap masukan, oleh sel neuron, semua sinyal yang sudah dikalikan dengan bobot dijumlahkan kemudian ditambah lagi dengan bias. Hasil penjumlahan ini di*input*kan ke suatu fungsi (fungsi aktivasi) menghasilkan keluaran dari neuron (di sini digunakan fungsi aktivasi linier). Selama proses pembelajaran, bobot-bobot dan bias selalu diperbaharui menggunakan algoritma belajar, jika ada *error* pada keluaran. Untuk proses identifikasi, bobot-bobot yang secara langsung memboboti masukan inilah yang dinamakan sebagai parameter yang dicari, seperti terlihat pada Gambar 2.12, parameter yang dicari adalah harga w1, w2, w3 dan w4. Dalam identifikasi secara *on-line*, neuron ataupun jaringan neuron akan selalu 'belajar' setiap ada data masukan dan keluaran.



Gambar 2.12 Sel neuron ketika sedang melakukan proses belajar

Algoritma untuk memperbaharui bobot pada neuron satu lapis adalah seperti pada bagian algoritma pemrograman JST satu lapis langkah ke-7.

Sedangkan untuk JST dua lapis adalah seperti pada bagian algoritma pemrograman JST dua lapis langkah ke-8 dan 9.

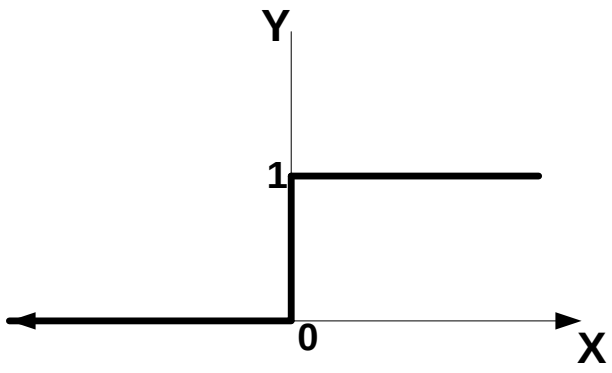
2.2.6.2 Fungsi Aktivasi

Menurut Kusumadewi (2010:77) ada beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan, antara lain :

1. Fungsi Undak Biner (*Hard Limit*)

Jaringan dengan lapisan tunggal sering menggunakan fungsi undak (*step function*) untuk mengkonversikan *input* dari suatu variabel yang bernilai kontinu ke suatu *output* biner (0 atau 1) Gambar 2.13. Fungsi undak biner (*hard limit*) dirumuskan sebagai (Demut,1998):

$$y = \begin{cases} 0, & jika\ x \leq 0 \\ 1, & jika\ x \geq 0 \end{cases} \tag{2.18}$$

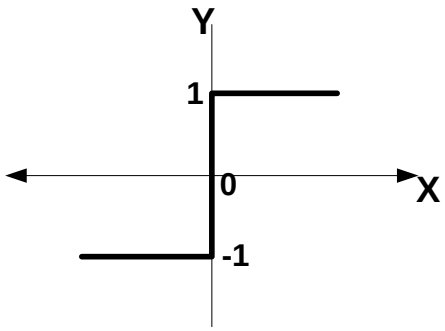


Gambar 2.13 Fungsi Aktivasi: Undak Biner (*Hard Limit*)

2. Fungsi Bipolar (*Symetric Hard Limit*)

Fungsi bipolar sebenarnya hampir sama dengan fungsi undak biner, hanya saja *output* yang dihasilkan berupa 1, 0 atau -1 (Gambar 2.14). Fungsi *Symetric Hard Limit* dirumuskan sebagai (Demuth,1998):

$$y = \begin{cases} 1, & jika\ x \geq 0 \\ -1, & jika\ x < 0 \end{cases} \tag{2.19}$$

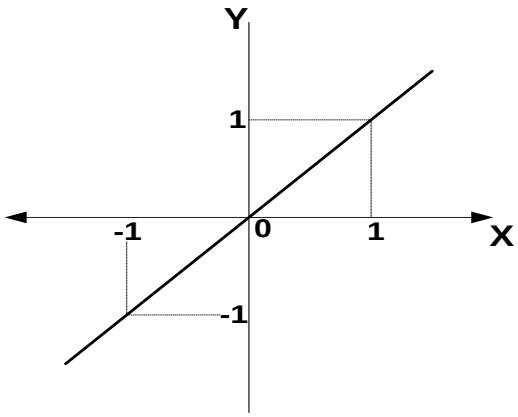


Gambar 2.14 Fungsi Aktivasi: Bipolar (*Symetric Hard Limit*)

3. Fungsi Linear (identitas)

Fungsi linear memiliki nilai *output* yang sama dengan nilai *input*nya (Gambar 2.15). Fungsi linear dirumuskan sebagai (Demuth, 1998):

$$y = x \tag{2.20}$$



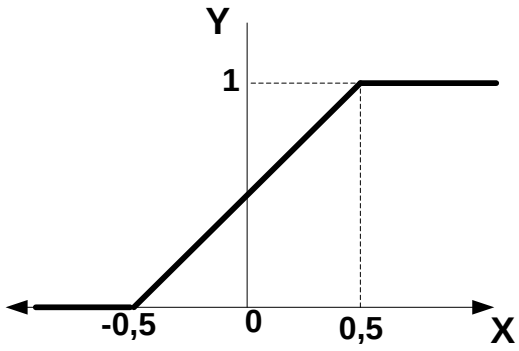
Gambar 2.15 Fungsi Aktivasi: Linear (Identitas)

4. Fungsi *Saturating Linear*

Fungsi ini akan bernilai 0 jika *input*nya kurang dari -½, dan akan bernilai 1 jika *input*nya lebih dari ½. Sedangkan jika nilai *input* terletak antara -½ dan ½, maka *output*nya akan bernilai sama dengan nilai *input* ditambah ½ (Gambar 2.16).

Fungsi saturating linear dirumuskan sebagai (Demuth, 1998):

$$y = \begin{cases} 1; & \text{jika } x \geq 0{,}5 \\ x + 0{,}5; & \text{jika } -0{,}5 \leq x \leq 0{,}5 \\ 0; & \text{jika } x \leq -0{,}5 \end{cases} \quad (2.21)$$



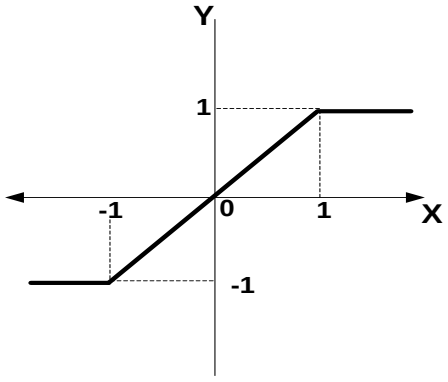
Gambar 2.16 Fungsi Aktivasi: *saturating Linear*

5. Fungsi *Symetric Saturating Linear*

Fungsi ini akan bernilai -1 jika *input*nya kurang dari -1, dan akan bernilai 1 jika *input*nya lebih dari 1. Sedangkan jika nilai *input* terletak antara -1 dan 1, maka *output*nya akan bernilai sama dengan nilai *input*nya (Gambar 2.17).

Fungsi *Symetric Saturating Linear* dirumuskan sebagai (Demuth, 1998):

$$y = \begin{cases} 1; & \text{jika } x \geq 1 \\ x; & \text{jika } -1 \leq x \leq 1 \\ -1; & \text{jika } x \leq -1 \end{cases} \quad (2.22)$$



Gambar 2.17 Fungsi Aktivasi: *Symetric Saturating Linear*

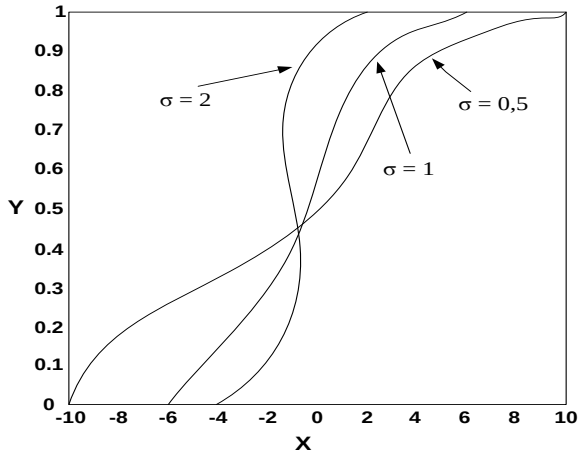
6. Fungsi Sigmoid Biner

Fungsi ini digunakan untuk jaringan syaraf yang dilatih dengan menggunakan metode *backpropagation.* Fungsi sigmoid biner memiliki nilai pada range 0 sampai 1. Oleh karena itu, fungsi ini sering digunakan untuk jaringan syaraf yang membutuhkan nilai *output* yang terletak pada interval 0 sampai 1. Namun, fungsi ini bisa juga digunakan oleh jaringan syaraf yang nilai *output*nya 0 atau 1 (Gambar 2.18).

Fungsi sigmoid biner dirumuskan sebagai (Demuth,1998):

$$y = f(x) = \frac{1}{1+e^{-\sigma x}} \tag{2.23}$$

dengan : $f'(x) = \sigma f(x)[1 - f(x)]$



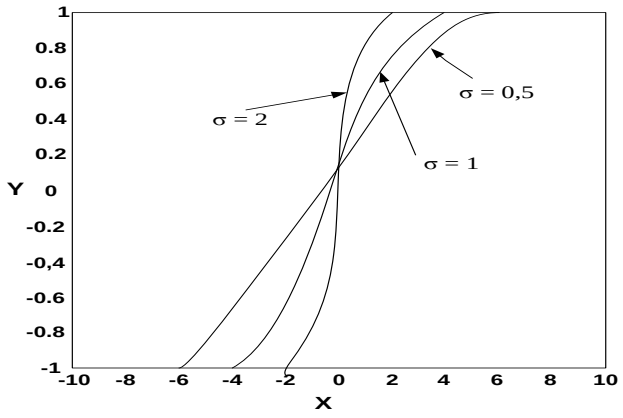
Gambar 2.18 Fungsi Aktivasi: Sigmoid Biner

7. Fungsi Sigmoid Bipolar

Fungsi sigmoid bipolar hampir sama dengan fungsi sigmoid biner, hanya saja *output* dari fungsi ini memiliki range anara 1 sampai -1 (Gambar 2.19).

Fungsi sigmoid bipolar dirumuskan sebagai (Demuth,1998):

$$y = f(x) = \frac{1-e^{-x}}{1+e^{-x}} \tag{2.24}$$



Gambar 2.19 Fungsi Aktivasi: Sigmoid Bipolar

2.2.6.3 Metode Pelatihan/ Pembelajaran

Cara berlangsungnya pembelajaran atau pelatihan JST dikelompokkan menjadi 3 yaitu:

a. *Survised Learning* (pembelajaran terawasi)

   Pada metode ini, setiap pola yang diberikan kedalam JST telah diketahui keluarannya. Selisih antara pola keluaran aktual (keluaran yang dihasilkan) dengan pola keluaran yang dikehendaki (target keluaran) yang disebut *error* digunakan untuk mengkoreksi bobot JST sehingga JST mampu menghasilkan keluaran sedekat mungkin dengan pola kelauran target yang telah diketahui oleh JST. Contoh algoritma JST yang menggunakan metode ini adalah: *Perceptron, ADALINE, Boltzman, Hopfield,* LVQ *(Learning Vector Quantization)* dan *Backpropagation*.

b. *Unsupervised Learning* (pembelajaran tak terawasi)

   Pada metode ini, tidak memerlukan target keluaran. Pada metode ini tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam suatu

range tertentu. Pembelajaran ini biasanya sangat cocok untuk klasifikasi pola. Contoh algoritma JST menggunakan metode ini adalah: *Competitive, Hebbian, Kohonen,* dan *Neocognitron*.

c. *Hybrid Learning* (pembelajaran hibrida)

Merupakan kombinasi dari metode pembelajaran *Supervised Learning* dan *Unsupervised Learning.* Sebagian bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi. Contoh algoritma JST yang menggunakan metode ini yaitu : 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 digunakan algoritma *Backpropagation*.

2.2.7 Sistem Neuro Fuzzy

Menurut Jang, J.–S.R, Sun, C.-T dan Mitsuzani, E., (1997:226) jaringan neural adalah struktur jaringan yang keseluruhan tingkah laku masukan-keluaran ditentukan oleh sekumpulan parameter-parameter yang dimodifikasi. Salah satu struktur jaringan neural adalah *multilayer perceptrond* (MLP). Jenis jaringan ini khusus bertipe umpan maju. MLP telah diterapkan dengan sukses untuk menyelesaikan masalah-masalah yang sulit dan beragam dengan melatihnya menggunakan algoritma propagasi balik dari kesalahan atau *Error Back-Propagation* (EBP).

Secara garis besar proses EBP mengandung dua tahap melalui jaringan. Yang pertama, adalah tahap umpan maju, dengan suatu pola aktivitas (*vector*

input) diberikan kepada jaringan dan efeknya merambat melalui jaringan. Akhirnya suatu set keluaran dihasilkan sebagai respon jaringan. Tahap kedua adalah tahap mundur, dan bobot sinaptik ($W_{i,j}$) dari jaringan diubah-ubah sesuai dengan aturan koreksi kesalahan. Secara rinci, respon aktual dari jaringan disubstraksi dengan suatu respon yang diinginkan untuk menghasilkan sinyal kesalahan. Sinyal kesalahan dirambatkan ke belakang melalui jaringan melawan arus bobot sinaptik, sehingga dinamakan propagasi balik dari kesalahan. Bobot sinaptik diubah sehingga respon aktual jaringan semakin mendekati respon yang diinginkan. Kegunaan dari sistem ini adalah kemampuannya untuk belajar sendiri dari data-data numerik (pasangan data masukan-keluaran).

Selanjutnya, sistem fuzzy dapat melukiskan suatu sistem dengan pengetahuan linguistik yang mudah dimengerti. Sistem inferensi fuzzy dapat ditelaah dengan algoritma propagasi balik berdasarkan pasangan data masukan-keluaran menggunakan arsitektur jaringan neural. Dengan cara ini memungkinkan sistem fuzzy belajar. Menurut Jang, J.–S.R, Sun, C.-T dan Mitsuzani, E., (1997:1,458) gabungan sistem fuzzy dengan jaringan neural inilah yang disebut dengan neuro fuzzy.

Menurut Rahmat (2006:6) ada dua macam struktur neuro fuzzy yaitu, *Adaptive Neuro-Fuzzy Inference System* (ANFIS) dan *Modified Adaptive Neuro-Fuzzy Inference System* (Mod_ANFIS). Sistem neuro-fuzzy berstruktur ANFIS termasuk dalam kelas jaringan neural namun berdasarkan fungsinya sama dengan sistem inferensi fuzzy. Pada neuro-fuzzy, proses belajar pada jaringan neural

dengan jumlah pasangan data berguna untuk memperbaharui parameter-parameter sistem inferensi fuzzy.

ANFIS (*Adaptive Neuro Fuzzy Inference System atau Adaptive Network-based Fuzzy Inference System*) adalah arsitektur yang secara fungsional sama dengan *fuzzy rule base* model Sugeno. Arsitektur ANFIS juga sama dengan jaringan syaraf dengan fungsi radial dengan sedikit batasan tertentu. Bisa dikatakan bahwa ANFIS adalah suatu metode yang mana dalam melakukan penyetelan aturan digunakan algoritma pembelajaran terhadap sekumpulan data. ANFIS juga memungkinkan aturan-aturan untuk beradaptasi.

2.2.7.1 Proses Belajar ANFIS

Menurut Jang, J.–S.R, Sun, C.-T dan Mitsuzani, E., (1997;340) ANFIS dalam kerjanya menggunakan algortima belajar *hybrid*, yaitu menggabungkan metode *Least-Squares Estimotor* (LSE) dan *Error Back-Propagation* (EBP). Dalam struktur ANFIS metode EBP dilakukan di lapisan ke-1, sedangkan metode LSE dilakukan di lapisan ke-4.

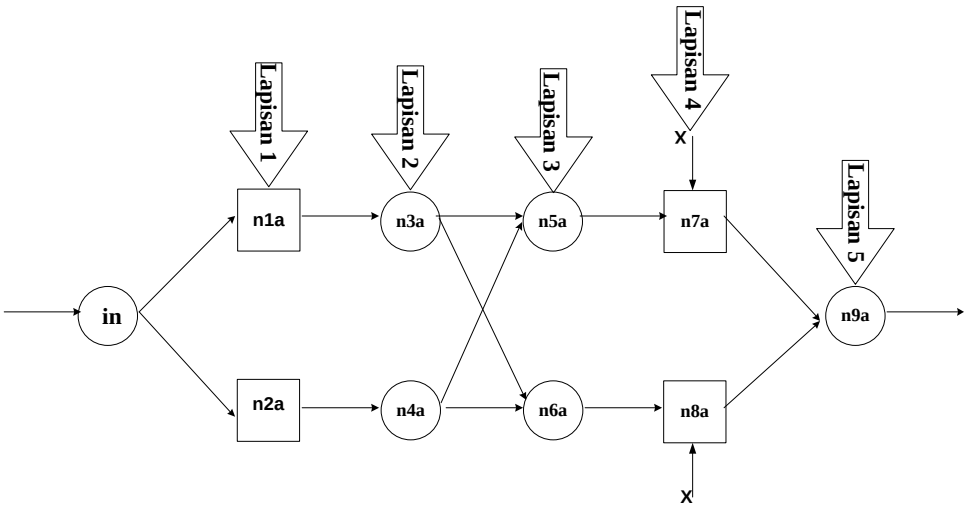
Pada lapisan ke-1 parameternya merupakan parameter dari fungsi keanggotaan himpunan fuzzy sifatnya nonlinier terhadap keluaran sistem. Proses belajar pada parameter ini menggunakan EBP untuk memperbaharui nilai parameternya. Sedangkan pada lapisan ke-4, parameter merupakan parameter linier terhadap keluaran sistem, yang menyusun basis kaidah fuzzy. Proses belajar untuk memperbaharui parameter. Lapisan ini menggunakan metode LSE. Proses belajar ANFIS dapat dilihat pada tabel berikut :

Tabel 2.1 Proses Belajar ANFIS (Jang, 1997;340)

	Arah Maju	**Arah Mundur**
Parameter premis	Tetap	EBP
Parameter konsekuen	LSE	Tetap
Sinyal	Keluaran simpul	Sinyal kesalahan

1. Tahap Maju

Untuk sistem dengan satu masukan dan satu keluaran arsitektur ANFIS digambarkan sebagai berikut :



Gambar 2.20 Struktur ANFIS

Penjelasan pada masing-masing labisan sebagai berikut:

Lapisan ke-1:

Mendefinisikan parameter fungsi keanggotaan (a1, a2, b1, b2, c1, c2), kemudian mengimplementasikan fungsi keanggotaan pada lapisan ini (menggunakan fungsi bell), dengan demikian keluaran dari simpul di lapisan ini merupakan fungsi bell. Untuk semua keluaran simpul pada tahap maju diberi simbol ‘a’, sehingga pada lapisan 1 diperoleh keluaran simpul n1a dan n2a. Tanda

‘a’ diberikan untuk membedakan dengan nilai keluaran simpul baru yang diberi simbol ‘b’ (setelah dikoreksi).

Setiap simpul pada lapisan ini adalah simpul adaptif dengan fungsi simpul:

n1a = Bell (x;a1,b1,c1)

n2a = Bell (x;a2,b2,c2) (2.25)

Dengan x adalah masukan bagi simpul n1a dan n2a, sedangkan a1,b1,c1,a2,b2,c2 adalah parameter fungsi keanggotaan Bell. Dan fungsi Bell yang digunakan dinyatakan dengan persamaan sebagai berikut:

$$\mu A(x) = \frac{1}{1+\left|\frac{x-c_i}{a_i}\right|^{2b}} \quad (2.26)$$

Dengan {ai, bi, ci} adalah himpunan parameter. Parameter pada lapisan ini disebut parameter-parameter premis.

Lapisan ke-2:

Setiap simpul pada lapisan ini diberi label n3a dan n4a, bersifat non-adaptif (parameter tetap) yang meneruskan hasil dari lapisan ke-1. Karena sistem yang digunakan hanya satu masukan, maka tidak ada logika fuzzy (mekanisme inferensi AND). Dengan demikian keluaran dari lapisan ke-2 adalah:

n3a = n1a

n4a = n2a 2.27)

Lapisan ke-3:

Pada lapisan ini dilakukan normalisasi dari sinyal yang masuk yang digunakan untuk menghasilkan data yang telah tersinkronisasi dari simpul pada lapisan ke-2. Setiap simpul pada lapisan ke-3 ini diberi label n5a dan n6a, juga

bersifat non-adaptif. Masing-masing simpul menampilkan derajat pengaktifan ternormalisasi dengan bentuk sebagai berikut:

$$\text{n5a} = \frac{\text{n3a}}{\text{n3a+n4a}}$$

$$\text{n6a} = \frac{\text{n4a}}{\text{n3a+n4a}} \quad (2.28)$$

Lapisan ke-4:

Least-Squares Estimator (LSE), ditulis sebagai berikut (Jang, 1997):

$$A^T A\hat{\theta} = A^T y \quad (2.29)$$

Tiap simpul pada lapisan ini berupa simpul adaptif, oleh karena itu pada lapisan ini diperoleh matriks A, untuk ANFIS matriks A dituliskan sebagai berikut:

$$\text{A} = \begin{bmatrix} (\text{n5a})\text{x}_\text{i} & (\text{n5a}) & (\text{n6a})\text{x}_\text{i} & (\text{n6a}) \\ \vdots & \vdots & \vdots & \vdots \\ (\text{n5a})\text{x}_\text{n} & (\text{n5a}) & (\text{n6a})\text{x}_\text{n} & (\text{n6z}) \end{bmatrix} \quad (2.30)$$

Jumlah baris dari matriks A sebanyak jumlah data masukan X. Pada lapisan ini dicari nilai parameter konsekuen θ (p1,q1,p2,q2) dengan menggunakan metode *least squares estimator* (LSE).

Persamaan untuk metode LSE adalah sebagai berikut:

$$\theta = \text{inv}(\text{A}^\text{T}\text{A})\text{A}^\text{T}\,.\,\text{y} \quad (2.31)$$

dengan, y = keluaran atau target yang diinginkan, sehingga diperoleh parameter :

$$\theta = [\text{p1 q1 p2 q2}]^\text{T} \quad (2.32)$$

Selanjutnya untuk menghitung keluaran dari lapisan ke-4 (n7a dan n8a) digunakan persamaan sebagai berikut:

$$\text{n7a} = \text{n5a(p1x + q1)}$$

$$\text{n8a} = \text{n6a(p2x + q2)} \quad (2.33)$$

Lapisan ke-5:

simpul tunggal pada lapisan ini diberi label n9a, yang mana menhitung semua keluaran sebagai penjumlahan dari semua sinyal yang masuk, yaitu:

n9a = n7a + n8a (2.34)

yang selanjutnya sebagai keluaran jaringan.

2. Tahap Mundur

Untuk melakukan koreksi kesalahan keluaran jaringan digunakan metode penurunan gradient atau *gradient descent* menggunakan algoritma *error backpropagation* (EBP).

Dimisalkan pada struktur ANFIS terdapat L lapisan dan N(ℓ) simpul serta terdapat P pasangan data antara proses belajar jaringan adaptif. Pengukuran kesalahan (*error measure*) pada tiap pasangan data latih ke-p ($1 \leq p \leq P$) dapat didefinisikan sebagai jumlah kuadrat kesalahan atau :

$$Ep = \sum_{k=1}^{N(\ell)} (d_k^p - x_{L,k}^p)^2 \quad (2.35)$$

dimana:

L = jumlah lapisan jaringan adaptif

N(ℓ) = jumlah simpul

d_k^p = komponen ke-k dari vektor keluaran yang diinginkan

$x_{L,k}^p$ = vektor keluaran aktual yang dihasilkan sistem jaringan adaptif dengan masukan dari vektor masukan ke-p dari P pasangan data.

Dengan mendefinisikan sinyal kesalahan (*error signal*) $\varepsilon_{\ell,i}$ sebagai turunan pertama dari pengukuran kesalahan Ep terhadap keluaran simpul ke-i pada lapisan ke- ℓ, maka dapat dinotasikan dengan:

$$\varepsilon_{\ell,i} = \frac{\partial^{+} Ep}{\partial x_{\ell,i}} \tag{2.36}$$

Sinyal kesalahan untuk keluaran simpul ke-i pada lapisan ke-L dapat dihitung secara langsung dengan

$$\varepsilon_{L,i} = \frac{\partial^{+} Ep}{\partial x_{L,i}} = \frac{\partial Ep}{\partial x_{L,i}} \tag{2.37}$$

Jika pengukuran kesalahan seperti yang didefinisikan pada persamaan (2.35), maka persamaan menjadi:

$$\varepsilon_{L,i} = 2\left(d_i^p - x_{L,i}^p\right) \tag{2.38}$$

Untuk simpul dalam, sinyal kesalahan dapat diperoleh dengan menggunakan aturan rantai

$$\varepsilon_{\ell,i} = \frac{\partial^{+} Ep}{\partial x_{\ell,i}} = \sum_{m=1}^{N(\ell+1)} \frac{\partial^{+} Ep}{\partial x_{\ell+i,m}} \frac{\partial f_{\ell+1,m}}{\partial x_{\ell,i}} = \sum_{m=1}^{N(\ell+1)} \varepsilon_{\ell+1,m} \frac{\partial f_{\ell+1,m}}{\partial x_{\ell,i}} \tag{2.39}$$

Dengan $0 \leq \ell \leq L-1$. Sinyal kesalahan simpul dalam pada lapisan ke- ℓ dapat dinyatakan sebagai kombinasi linier dari sinyal kesalahan dari simpul pada lapisan ke- ℓ+1. Untuk memperoleh sinyal kesalahan simpul ke-I pada lapisan ke- ℓ $\left(0 \leq \ell \leq \mathrm{L} \text{ dan } 1 \leq \mathrm{i} \leq \mathrm{N}(\ell)\right)$ pertama digunakan persamaan (2.37) untuk mendapatkan sinyal kesalahan pada lapisan keluaran kemudian secara iteratif sampai mencapai lapisan yang diinginkan menggunakan persamaan (2.39). Prosedur di atas disebut propagasi balik (*backpropagation*) karena sinyal kesalahan dihitung secara mundur dari lapisan keluaran hingga lapisan masukan.

Vektor gradien didefinisikan sebagai turunan pertama dari pengukuran kesalahan terhadap tiap parameter. Jika α adalah parameter simpul ke-I lapisan ke- ℓ, maka diperoleh:

$$\frac{\partial^+ Ep}{\partial \alpha} = \frac{\partial Ep}{\partial x_{\ell,i}} \qquad \frac{\partial f_{\ell,i}}{\partial \alpha} = \varepsilon_{\ell,i} \frac{\partial f_{\ell,i}}{\partial x_{\ell,i}} \tag{2.40}$$

Jika α merupakan parameter yang ada pada beberapa simpul maka persamaan (2.36) menjadi

$$\frac{\partial^+ Ep}{\partial \alpha} = \sum_{x \in S} \frac{\partial^+}{\partial x^*} \frac{\partial f^*}{\partial \alpha} \tag{2.41}$$

Dengan S merupakan himpunan simpul yang mempunyai parameter α, x*, dan f* adalah keluaran dan fungsi dari simpul yang bersangkutan. Turunan masing-masing secara keseluruhan terhadap pengukuran kesalahan akan menghasilkan

$$\frac{\partial^+ E}{\partial \alpha} = \sum_{p=1}^{P} \frac{\partial^+ Ep}{\partial \alpha} \tag{2.42}$$

Untuk mempercepat konvergensi propagasi balik parameter α dengan metode *simple steepest descent* maka

$$\Delta\alpha = -\eta \frac{\partial^+ E}{\partial \alpha} \tag{2.43}$$

Dengan η adalah laju proses belajar, dan didefinisikan:

$$\eta = \frac{K}{\sqrt{\sum_\alpha \left(\frac{\partial^+ E}{\partial \alpha}\right)^2}} \tag{2.44}$$

K adalah ukuran langkah (*step size*), yang mana nilai K dapat diubah-ubah untuk mempercepat konvergensi. Parameter untuk simpul selanjutnya diperbaharui dengan:

$$\alpha_{i+1} = \alpha_i + \Delta\alpha \tag{2.45}$$

Untuk sistem pada Gambar 2.21 dengan menggunakan persamaan 2.37 dan persamaan 2.39 maka diperoleh persamaan-persamaan:

$$\varepsilon_9 = \frac{\partial \mathrm{E_P}}{\partial \mathrm{n9a}} \tag{2.46}$$

adalah sinyal kesalahan lapisan keluaran dari jaringan adpatif. Kemudian secara iteratif dengan propagasi balik diperoleh sinyal kesalahan lapisan ke-5 sampai dengan lapisan ke-1, yaitu:

$$\varepsilon_8 = W_{9,8}\varepsilon_9 \tag{2.47}$$

$$\varepsilon_7 = W_{9,7}\varepsilon_9 \tag{2.48}$$

$$\varepsilon_6 = W_{8,6}\varepsilon_8 \tag{2.49}$$

$$\varepsilon_5 = W_{7,5}\varepsilon_7 \tag{2.50}$$

$$\varepsilon_4 = W_{6,4}\varepsilon_6 + W_{5,4}\varepsilon_5 \tag{2.51}$$

$$\varepsilon_3 = W_{5,3}\varepsilon_5 + W_{6,3}\varepsilon_6 \tag{2.52}$$

$$\varepsilon_2 = W_{4,2}\varepsilon_4 \tag{2.53}$$

$$\varepsilon_1 = W_{3,1}\varepsilon_3 \tag{2.54}$$

Jika sinyal kesalahan di lapisan ke-1 sudah diperoleh, maka untuk mendapatkan nilai fungsi kenggotaan Bell yang seharusnya (nilai fungsi Bell yang baru) digunakan persamaan:

$$\begin{aligned} \mathrm{n1b} &= \mathrm{n1a} + \varepsilon_1 \\ \mathrm{n2b} &= \mathrm{n2a} + \varepsilon_2 \end{aligned} \tag{2.55}$$

dengan n1b dan n2b adalah nilai fungsi keanggotaan yang baru, sedangkan n1a dan n2a nilai fungsi kenaggotaan yang lama ditambahkan dengan sinyal kesalahan yang telah diperoleh (ε_1 dan ε_2).

Selanjutnya untuk mendapatkan parameter fungsi keanggotaan Bell yang baru digunakan fungsi turunan keanggotaan Bell. Fungsi bell dan turunannya seperti pada persamaan 2.9 s.d. 2.12. Sehingga parameter fungsi keanggotaan

fuzzy yang baru adalah fungsi keanggotaan fuzzy yang lama ditambah dengan turunannya, yaitu:

$$a_{baru} = a_{lama} + \frac{\partial z}{\partial a} \quad (2.56)$$

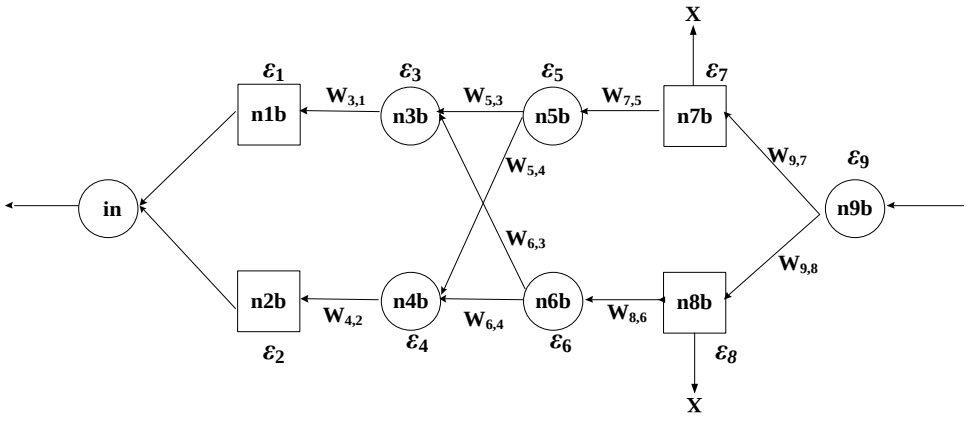
$$b_{baru} = b_{lama} + \frac{\partial z}{\partial b}, \text{dan} \quad (2.57)$$

$$c_{baru} = c_{lama} + \frac{\partial z}{\partial c} \quad (2.58)$$

Sedangkan untuk mencari turunan sebagai bobot sinyal kesalahan dari simpul I ke simpul j pada persamaan 2.47 s.d. 2.54 digunakan persamaan (2.59) berikut:

$$W_{i,j} = \frac{\partial f_i}{\partial x_j}, \text{untuk i} > j \quad 2.59)$$

Dengan demikian proses belajar propagasi balik (EBP) untuk contoh ini bisa digambarkan sebagai berikut:



Gambar 2.21 Proses Belajar Propagasi Balik pada ANFIS

Jika parameter fungsi keanggotaan yang baru sudah diperoleh, maka iterasi dilanjutkan dengan proses maju seperti yang telah dijelaskan. Jika telah diperoleh keluaran jaringan maka sinyal kesalahannya diperiksa lagi. Selanjutnya sinyal kesalahan ini dipropagasi balik sampai lapisan ke-1 untuk diperoleh lagi

parameter keanggotaan yang baru. Demikian seterusnya, proses ini berulang sampai sinyal kesalahan dapat diterima atau sampai dengan iterasi maksimum tercapai.

2.2.8 Pemodelan Pengembangan Sistem

2.2.8.1 Entity-relationship Diagram (ERD)

Diagram E-R adalah diagram grafikal keseluruhan struktur logika dari sebuah basis data. *Entity-Relationship* diagram tidak menggambarkan aliran data atau proses data. E-R Diagram menggambarkan data pada data *store*.

Diagram E-R ini berfungsi untuk menggambarkan relasi dari dua file atau dua tabel yang dapat digolongkan dalam tiga macam bentuk relasi yaitu satu ke satu, satu kebanyak dan banyak ke banyak.

Model E-R yang berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta yang ditinjau, dapat digambarkan dengan lebih sistematis dengan menggunakan *Diagram Entity-Relationship* (Diagram E-R).

2.2.8.2 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data di mana komponen-komponen tersebut terdapat asal, tujuan, dan penyimpanan dari data tersebut.

Data Flow Diagram digunakan untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem informasi yang ada, atau untuk menyusun dokumentasi untuk sistem informasi yang baru. *Data Flow Diagram* (DFD) merupakan alat

bantu dari pengembangan sebuah sistem yang dibangun scara terstruktur atau prosedural dan DFD terdiri dari beberapa level.

2.2.8.3 Diagram Konteks

Diagram Konteks adalah sebuah diagram sederhana yang menggambarkan hubungan antara *entity* luar, masukan dan keluaran dari sistem. Diagram konteks direpresentasikan dengan lingkaran tunggal yang mewakili keseluruhan sistem (Kristanto, 2008).

2.2.8.4 Flowmap

Flowmap adalah campuran peta dan *flowchart*, yang menunjukan pergerakan benda dari satu lokasi ke lokasi lain, seperti jumlah orang dalam migrasi, jumlah barang yang diperdagangkan, atau jumlah paket dalam jaringan.

Flowmap menolong seorang analis dan *programmer* untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.

2.2.8.5 *Data Dictionary* (DD/Kamus Data)

Kamus data (*Data Dictionary*) adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data dapat mendefinisikan data yang mengalir pada sistem dengan lengkap. Kamus data dapat digunakan pada tahap analisa dan perancangan sistem. Pada tahap perancangan sistem, kamus data digunakan untuk merancang masukan (*input*), merancang laporan–laporan dan *database*.

Dengan adanya kamus data, didapat definisi-definisi dari bentuk–bentuk yang tidak dimengerti dalam DFD yaitu aliran data, file, proses dan elemen-

elemen data. Arus data pada DFD bersifat global, hanya ditunjukan nama arus datanya saja.

2.2.9 Basis Data

Basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil *query* basis data disebut sistem manajemen basis data (*database management system*, DBMS).

Basis data digunakan karena memiliki keuntungan sebagai berikut:

1. Mengurangi redundansi
2. Data dapat di-*share* antar aplikasi
3. Dapat dilakukan standardisasi data
4. Batasan *security* dapat diterapkan
5. Mengelola integritas data (akurasinya terjamin)
6. Independensi data (objektif DBS), basis data dapat berkembang tanpa mempengaruhi aplikasi yang telah ada.

Secara definitif, basis data merupakan suatu objek terstruktur. Objek terstruktur tersebut terdiri atas data dan metadata. Data pada basis data merupakan informasi deskriptif yang benar-benar tersimpan, misalnya ‘Nama’ atau ‘Alamat’. Sedangkan metadata merupakan bagian yang menjelaskan tentang struktur data tersebut dalam basis data, misalnya *field* untuk ‘Nama’ dan ‘Alamat’, panjang *field*, atau tipe data untuk masing-masing *field*.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya, penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut.

Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layanan mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel. Istilah basis data mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai sistem manajemen basis data (database management system/DBMS).

2.2.10 *Database* Management System (DBMS)

Database Management System (DBMS) adalah suatu sistem perangkat lunak yang digunakan untuk memanipulasi/memproses basis data. Sedangkan istilah *relational database management system* digunakan untuk menyebut suatu perangkat lunak yang dapat menangani basis data relasional dan berkomunikasi dengan *engine* basis data tersebut .

Diperlukan suatu sistem untuk diintegrasikan data file kedalam suatu file sehingga bisa melayani berbagai *user* yang berbeda. Perangkat keras serta prosedur yang mengelola *database* merupakan suatu *database* manajemen sistem.

DBMS memungkinkan untuk memebentuk dan meremajakan file-file, memilih, mendatakan dan menyortir data, dan untuk menghasilkan laporan-laporan.

2.2.11 Borland Delphi 7

Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Salah satu kelebihan Delphi adalah aplikasinya bisa dikembangkan diatas berbagai macam sistem operasi, misalnya Windows, UNIX, LINUX dan sebagainya. Keunggulan bahasa pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrograman yang terstruktur. Keunggulan lain Delphi adalah dapat dipergunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis windows.

Delphi menggunakan bahasa Objek Pascal sebagai dasar. Untuk mempermudah pemograman dalam membuat program aplikasi, Delphi menyediakan fasilitas pemograman yang sangat lengkap. Khusus untuk pemograman *database,* Delphi menyediakan objek yang sangat kuat, canggih dan lengkap, sehingga memudahkan pemograman dalam merancang, membuat dan menyelesaikan aplikasi *database* yang diinginkan. Selain itu Delphi juga dapat menangani data dalam berbagai format *database*, misalnya MS.Accses, SyBase, Oracle, FoxPro, Informix, InterBase, SQL Server, dll. Format *database* yang dianggap asli dari Delphi adalah Paradox dan dBase.

a. Komponen Delphi

Component palette terdiri dari beberapa komponen yang dapat dipilih yang digunakan untuk menangani beberapa tugas pemrograman. Komponen-komponen yang terletak pada bagian *component palette* sudah ditata dalam beberapa tab yang masing-masing menunjukan maksud dan fungsi. Masing-masing tab ditampilkan dalam konfigurasi default dan semua tergantung pada versi delphi yang digunakan.

Tabel berikut menunjukan daftar tab default dan beberapa komponen yang terdapat di dalamnya.

Tabel 2.2 Komponen Delphi

Nama Tab	Isi
Standart Kontrol	Kontrol-kontrol standar program windows dan menu
Additional Kontrol	Kontrol-kontrol tambahan
Win32 Kontrol	Kontrol-kontrol umum windows 9x/NT 4.0
System	Komponen dan kontrol-kontrol dari sistem komputer termasuk *timer*, multimedia dan DDE
Data Access	Komponen-komponen non-visual yang digunakan untuk mengakses tabel-tabel *database*, *query*, dan *report*
Data Controls	Komponen-komponen visual, dan kontrol-kontrol data*aware*
dbExpress	Komponen-komponen non-visual yang digunakan aplikasi untuk berhubungan dengan *database* dengan menggunakan *dbExpress*
DataSnap	Komponen dan kontrol-kontrol non-visual yang digunakan untuk membuat aplikasi *database* bertingkat (*multi-tiered*)
BDE	Komponen dan kontrol-kontrol non-visual yang digunakan untuk menghubungkan Informasi *database* dengan menggunakan *Borland Database Engine* (BDE)
ADO	Komponen dan kontrol-kontrol non-visual yang digunakan untuk menghubungkan Informasi *database* dengan menggunakan *ActiveX Data Object* (ADO)

Nama Tab	Isi
InterBase	Komponen dan kontrol-kontrol non-visual yang digunakan untuk menghubungkan secara langsung *database interbase* tanpa menggunakan BDE ataupun ADO
InternetExpress	Komponen yang digunakan untuk membangun aplikasi *InternetExspress* yang simultan dengan *Web Server* dan klien dari suatu aplikasi *database* bertingkat

b. Fitur Pada Delphi 7

Fitur baru dan perbaikan yang ada pada Borland Delphi 7 ini adalah :

1. IDE (*Interface Development Environtment*)

   Lingkungan pengembangan aplikasi (IDE) Borland Delphi 7 telah mengalami perubahan dari versi sebelumnya. Diantaranya, terdapat *Compiler Message*, perubahan pada *Component Pallete, Code Insight* dan *Debugger*.

2. *Web*

   Borland Delphi 7 menyediakan Intraweb buatan AtoZed Software, yang dapat digunakan untuk membuat aplikasi web server dengan sarana standar. Borland Delphi 7 juga mendukung pada Apache 2. Borland juga menghilangkan Win-CGI sebagai target aplikasi *Web server* dan *web service*. Fasilitas untuk Web server juga mengalami perbaikan – perbaikan.

3. COM

   Sekarang ini dengan Delphi 7.0, dapat membuat CoClass wrapper bagi pengembangan – pengembangan .NET dengan cara menggunakan kotak dialog Import Type Library. Dengan adanya resulting wrapper, maka fitur interoperabilitas dari Microsoft’s NET Framework dapat digunakan.

4. *Database*

   Pada Delphi 7.0 driver dbExpress telah diupdate bagi Informix SE, Oracle 9i, DB2 7.2, InterBase 6.5, dan MySQL 3.23.49. Driver baru bagi MSSQL 2000 juga tersedia. Disamping itu, beberapa hal baru dan perubahan juga dilakukan pada komponen *database*. Borland juga telah membuang SQL Links. Borland merekomendasikan pemakaian dbExpress bagi *database* SQLServer yang diakses di Delphi.

5. *Component Library*

   Jika ditelusuri komponen librari Delphi 7.0, maka akan ditemukan komponen baru, unit baru, komponen yang berubah, komponen yang hilang dan komponen yang mendukung bagi tema Windows XP.

6. *Runtime Library*

   Beberapa perubahan di *Runtime Library* antara lain ialah perubahan pada unit *Classes, Math, StdConv,StrUtils, SysUtils, VarCmplx,* dan *Variants*.

7. *Compiler*

   Kompiler Delphi dcc32 sekarang ini *support* terhadap tiga *warning* kompiler tambahan, yaitu *Unsafe_Type, Unsafe_Code*, dan *Unsafe_Cast*. Warnings tersebut *default*nya adalah *disabled*, tetapi dapat di-*enabled*. Fitur ini sangat membantu ketika akan mem*port* kode ke lingkungan eksekusi terkendali di *platform* Microsoft's .NET.

8. *Model Maker*

   Sarana baru yang disebut *Model Maker* dapat membantu memudahkan proses desain, konstruksi, dan pengelolaan *class* dan *interface*. *Model Maker* juga

memiliki sarana untuk pembuatan diagram UML-s*tyle*, yang dapat dipakai untuk membuat dan memodifikasi *source code project*.

2.2.12 Pengertian MySQL

MySQL adalah perangkat lunak *database server* atau sebut saja *Database Smart*. *Database* ini semakin lama semakin populer. Dengan menggunakan *database* ini, data semakin aman dan berdaya guna. *Database* ini juga banyak dipakai pada web *database* sehingga data semakin terintegrasi antara *database* dekstop dengan *database* web. Untuk menggunakan *database* MySQL harus menginstalasinya dahulu ke komputer.

MySQL adalah server basis data yang kompak dan kecil yang ideal untuk banyak aplikasi basis data on-line. MySQL mendukung SQL standar (ANSI), meskipun tidak selengkap subset yang menjadi standar seperti PostgreSQL. MySQL dapat dijalankan di banyak platform dan memiliki kemampuan *multithreading* pada *server* UNIX. Pada lingkungan bukan UNIX, MySQL dapat dijalankan sebagai servis pada Windows NT dan sebagai proses normal pada mesin Windows 95/98. MySQL adalah *server* DBMS relasional SQL yang mendukung *multithreading* dan *multi-us*er. MySQL mengimplementasikan *client/server* yang terdiri dari sebuah *daemon server* (servis di *server*) dan banyak program dan pustaka klien yang berbeda-beda.

MySQL menjamin setiap unit kerja bersifat konsisten. Hal ini dilakukan dengan cara menulis data sebelum dan sesudah transaksi pada sebuah log transaksi. Log tersebut dapat dipakai untuk me-*restore database* ke keadaan konsisten jika sebuah aplikasi melakukan *rollback* (membatalkan operasi yang

sudah dikerjakan karena transaksi gagal) atau aplikasi akan me-*recover* data karena kegagalan sistem.

MySQL juga *Relational Database Management System*(RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial.

Sebagai *server database* dengan konsep *database* modern, MySQL memiliki keistimewaan. Beberapa keistimewaan dimiliki MySQL sebagai berikut:

1. *Portability*

   Database MySQL berfungsi dengan stabil tanpa kendala, berarti berlaku pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X *Server*, Solaris, Amiga, HP-Unix, dan lain-lain

2. *Open Source*

   MySQL merupakan *database* open source (gratis), di bawah lisensi GPL sehingga dapat memperoleh dan menggunakannya secara cuma-cuma tanpa membayar sepersen pun.

3. *Multiuser*

   MySQL merupakan *database* yang dapat digunakan untuk menangani beberapa user dalam waktu bersamaan tanpa mengalami masalah. Dan memungkinkan sebuah *database server* MySQL dapat diakses client secara bersamaan pula.

4. *Performance Tuning*

   MySQL mempunyai kecepatan yang cukup baik dalam menangani query-query sederhana, serta mampu memproses lebih banyak *SQL* per satuan waktu.

5. *Column Type*

   Database MySQL didukung dengan tipe data yang sangat kompleks, seperti signed/unsigned *integer*, float, double, char, varchar, text, blob, data, time, datetime, timestamp, year, set serta enum

6. *Command And Functions*

   MySQL *server* memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam query.

7. *Security*

   Sistem *Security* pada MySQL mempunyai beberapa lapisan sekuritas seperti tingkatan *subnetmask, hostname*, dan izin akses user dengan sistem perizinan yang mendetil serta *password* terenkripsi.

8. *Scalability dan Limits*

   MySQL mempunyai kemampuan menangani *database* dalam skala cukup besar, dengan jumlah record lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu dapat menampung indeks sampai 32 indeks pada tiap tabelnya.

9. *Connectivity*

   Adanya kemampuan MySQL melakukan koneksi dengan *client* menggunakan protokol TCP/IP, Unix socket (Unix), atau Named Pipes (NT).

10. *Localization*

    Adanya kemampuan dalam mendeteksi kesalahan (*error code*) pada *client* menggunakan lebih dari dua puluh bahasa.

11. *Interface*

    MySQL memiliki *interface* terhadap berbagai aplikasi dan bahasa pemograman menggunakan fungsi API (*Application Programming Interface*).

12. *Clients dan Tools*

    Database MySQL dilengkapi berbagai tools yang dapat digunakan untuk administrasi *database*.

13. Struktur Tabel MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE dibandingkan *database* lainnya.