

## **BAB II**

### **KAJIAN TEORI**

#### **A. Indek Harga Saham Gabungan**

IHSG menggambarkan suatu rangkaian informasi historis mengenai pergerakan harga saham gabungan seluruh saham, sampai pada tanggal tertentu (Sunariyah ,2006). Dalam hal ini mencerminkan suatu nilai yang berfungsi sebagai pengukuran kinerja suatu saham gabungan di bursa efek. Indeks harga saham gabungan seluruh saham adalah suatu nilai yang digunakan untuk mengukur kinerja gabungan seluruh saham yang tercatat di suatu bursa efek. Maksud gabungan dari seluruh saham ini adalah kinerja saham yang dimasukkan dalam perhitungan seluruh saham yang tercatat di bursa tersebut.

IIHSG memiliki lima fungsi (BEI, 2008) yaitu:

- 1) sebagai indikator tren pasar
- 2) sebagai indikator tingkat keuntungan
- 3) sebagai tolok ukur (benchmark) kinerja suatu portofolio
- 4) memfasilitasi pembentukan portofolio dengan strategi pasif
- 5) memfasilitasi berkembangnya produk derivatif

Sekarang ini Bursa Efek Indonesia memiliki 8 macam harga saham yang secara terus menerus disebarluaskan melalui media cetak maupun elektronik, sebagai salah satu pedoman bagi investor untuk berinvestasi di pasar modal (BEI, 2008).

Kedelapan macam indeks tersebut adalah sebagai berikut:

- a. IHSG, menggunakan semua emiten yang tercatat sebagai komponen perhitungan indeks.
- b. Indeks Sektoral, menggunakan semua emiten yang termasuk dalam masing-masing sektor.
- c. Indeks LQ45, menggunakan 45 emiten yang dipilih berdasarkan kriteria likuiditas dan kapitalisasi pasar, dengan kriteria-kriteria yang telah ditentukan.
- d. Jakarta Islamic Index (JII), menggunakan 30 emiten yang masuk dalam kriteria syariah dan termasuk saham yang memiliki likuiditas dan kapitalisasi tinggi.
- e. Indeks Kompas100, menggunakan 100 saham yang dipilih berdasarkan kriteria likuiditas dan kapitalisasi pasar, dengan kriteria-kriteria yang telah ditentukan.
- f. Indeks Papan Utama, menggunakan emiten yang masuk dalam kriteria papan utama.
- g. Indeks Papan Pengembangan, menggunakan emiten yang masuk dalam kriteria papan pengembangan.
- h. Indeks Individual, yaitu harga saham masing-masing emiten.

IHSG merupakan nilai yang digunakan untuk mengukur kinerja gabungan seluruh saham (perusahaan/emiten) tercatat di BEI.

IHSG merupakan salah satu indeks pasar saham yang digunakan oleh Bursa Efek Indonesia. Diperkenalkan pertama kali pada tanggal 1 April 1983, sebagai indikator pergerakan harga saham di BEI. Indeks ini mencakup pergerakan harga

seluruh saham biasa dan saham preferen ( yang tercatat di BEI. Hari Dasar untuk perhitungan IHSG adalah tanggal 10 Agustus 1982. Pada tanggal tersebut, Indeks ditetapkan dengan Nilai Dasar 100 dan saham tercatat pada saat itu berjumlah 13 saham.

Menurut Jogiyanto( 2008), IHSG sebenarnya merupakan angka indeks harga saham yang sudah disusun dan dihitung sehingga menghasilkan *trend*, di mana angka indeks adalah angka yang diolah sedemikian rupa sehingga dapat digunakan membandingkan kejadian berupa perubahan harga saham dari waktu ke waktu. Dalam perhitungan angka indeks ini digunakan waktu dasar (*base period*) dan waktu yang sedang berjalan (*given/parent period*).

Dasar perhitungan IHSG adalah jumlah nilai pasar dari total saham yang tercatat pada tanggal 10 Agustus 1982. Jumlah nilai pasar adalah total perkalian setiap saham tercatat (kecuali untuk perusahaan yang berada dalam program restrukturisasi) dengan harga di BEI pada hari tersebut. Formula perhitungannya adalah sebagai berikut:

$$IHSG = \frac{\sum p}{d} \times 100$$

dimana  $p$  adalah Harga Penutupan di Pasar Reguler, dan  $d$  adalah Nilai Dasar.

$$Rata - Rata IHSG = \frac{Jumlah\ IHSG\ periode\ harian\ selama\ 1\ bulan}{Jumlah\ periode\ waktu\ selama\ 1\ bulan}$$

IHSG memiliki nilai return indeks yang merupakan nilai keuntungan yang didapat para pelaku investor dalam melaksanakan perdagangan melalui IHSG. Nilai return indeks dapat dihitung menggunakan metode penghitungan sebagai berikut :

$$R_i = \frac{(P_{it}-P_{it-1})}{P_{it-1}} \quad (2.1)$$

dengan

$R_i$  = Return indeks pasar (IHSG)

$P_{it}$  = Indeks pasar (IHSG) pada periode t.

$P_{it-1}$  = Indeks pasar (IHSG) pada periode t -1 (tahun sebelumnya).

Perhitungan Indeks merepresentasikan pergerakan harga saham di pasar/bursa yang terjadi melalui sistem perdagangan lelang. Nilai dasar akan disesuaikan secara cepat bila terjadi perubahan modal emiten atau terdapat faktor lain yang tidak terkait dengan harga saham. Penyesuaian akan dilakukan bila ada tambahan emiten baru, HMETD (*right issue*), *partial/company listing*, waran dan obligasi konversi demikian juga *delisting*. Dalam hal terjadi *stock split*, dividen saham atau saham bonus, Nilai Dasar tidak disesuaikan karena Nilai Pasar tidak terpengaruh. Nilai Pasar adalah kumulatif jumlah saham hari ini dikali harga pasar hari ini (kapitalisasi pasar), sedangkan nilai dasar adalah kumulatif jumlah saham pada hari dasar dikali harga dasar pada hari dasar (Thobarry, 2009).

## **B. Wavelet**

### **1. Pengertian *Wavelet***

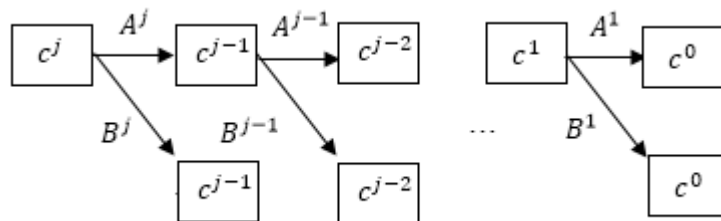
Gelombang (*wave*) adalah sebuah fungsi yang bergerak naik turun secara periodic. *Wavelet* merupakan gelombang yang dibatasi atau dapat dikatakan sebagai gelombang pendek (Sripathi, 2003). *Wavelet* pertama kali digunakan dalam analisis dan pemrosesan digital dari sinyal gempa bumi, yang tercantum dalam literatur oleh

A. Grossman dan J. Morlet. Penggunaan wavelet pada saat ini sudah semakin berkembang dengan penerapannya pada ilmu sains yang berhubungan dengan analisis wavelet dan teori transformasi wavelet. Dengan munculnya area sains ini wavelet mulai digunakan secara luas dalam filterasi dan pemrosesan data, pengenalan citra, sintesis dan pemrosesan berbagai variasi sinyal, kompresi dan pemrosesan citra, dan lain sebagainya.

Analisis dekomposisi wavelet merupakan fungsi basis yang memberikan alat baru sebagai pendekatan yang dapat digunakan dalam merepresentasikan data atau fungsi-fungsi yang lain. Algoritma *wavelet* mampu memproses data pada skala atau resolusi yang berbeda. *Wavelet* merupakan fungsi basis yang dapat digunakan dalam merepresentasikan data atau suatu fungsi tertentu kedalam 2 posisi yang diskalakan dengan variabel tertentu. Fungsi *wavelet* mempunyai nilai yang berbeda dari nol dalam interval waktu yang relatif pendek. Bentuk basis dari *wavelet* dikatakan sebagai transformasi *wavelet* yaitu membagi data dari sebuah fungsi kedalam beberapa komponen yang berbeda frekuensinya, serta menganalisa setiap komponen dengan skala yang berbeda.

## 2. Transformasi Wavelet

Transformasi merupakan suatu proses perubahan data kedalam bentuk lain agar mudah dianalisis, sebagai misal transformasi fourier merupakan suatu proses perubahan data (sinyal) kedalam beberapa gelombang kosinus yang berfrekuensi berbeda, sedangkan transformasi wavelet merupakan proses perubahan sinyal kedalam berbagai *wavelet* basis (*mother wavelet*) dengan berbagai fungsi pergeseran dan penyekalaan. Proses transformasi *wavelet* dilakukan dengan mengkonvolusi sinyal dengan data tapis atau dengan proses perata-rataan dan pengurangan secara berulang, yang sering disebut dengan metode *filter bank* (Bagus K, 2006). Gambar berikut ini menunjukkan proses transformasi *Wavelet* dengan cara *filter bank*



**Gambar 2.1.** Transformasi Wavelet. (Chui, 1992).

Terdapat dua jenis transformasi *wavelet* yaitu *Continue Wavelet Transform* (CWT) dan *Discrete Wavelet Transform* (DWT). CWT digunakan untuk sebuah fungsi yang berdomain bilangan real atas sumbu  $x$ , dan DWT digunakan untuk sebuah fungsi atas domain bilangan bulat (biasanya  $t = 0, 1, \dots, N-1$ , dimana  $N$  dinotasikan sebagai

banyaknya nilai dalam runtun waktu). Pada penelitian ini digunakan *Discrete Wavelet Transform* (DWT).

### 3. Discrete Wavelet Transform (DWT)

Dasar dari DWT dimulai pada tahun 1976 dimana teknik untuk mendekomposisi sinyal waktu diskrit ditemukan. DWT dianalisis dengan menggunakan penggambaran sebuah skala waktu sinyal digital didapatkan dengan menggunakan teknik filterisasi digital. Secara garis besar proses dalam teknik ini adalah dengan melewatkan sinyal yang akan dianalisis pada filter dengan frekuensi dan skala yang berbeda.

DWT dioperasikan pada fungsi dengan domain diskrit atau runtun waktu ( $t$ ), biasanya menggunakan domain waktu  $t = 0, 1, \dots, N - 1$ . Wavelet menganalisa data runtun waktu untuk dilatasi dan translasi data diskrit dengan menggunakan *mother wavelet*  $\psi(t)$ . Dasarnya proses dilatasi menggunakan (menggunakan interval  $2^{j-1}, j = 1, 2, 3, \dots$ ). Translasi nilai menggunakan interval  $2^j$ .

Satu dari fungsi *mother wavelet* adalah *Wavelet Haar*, A. Haar memperkenalkannya pada tahun 1910.

*Wavelet multilevel* adalah model transformasi *wavelet* diskrit yang mentransformasikan suatu data secara berulang-ulang. Langkah transformasi *wavelet multilevel* menurut Ida Bagus, (2006) adalah

- a. Data di transformasikan menggunakan DWT sehingga diperoleh koefisien *approximation* dan koefisien *detail*.

$$f \xrightarrow{H_1} (a_1|d_1) \quad (2.2)$$

Untuk  $f = (x_1, x_2, \dots, x_N)$  dengan  $N = 2^n$  merupakan banyaknya anggota  $f$  dan  $n$  merupakan konstanta positif.

Hasil dekomposisi level 1 adalah sebagai berikut:

$$a_1 = \left( \frac{x_1+x_2}{2}, \frac{x_3+x_4}{2}, \dots, \frac{x_{N-1}+x_N}{2} \right) \quad (2.3)$$

$$d_1 = \left( \frac{x_1-x_2}{2}, \frac{x_3-x_4}{2}, \dots, \frac{x_{N-1}-x_N}{2} \right) \quad (2.4)$$

$a$  merupakan approximation data dan  $d$  hasil dekomposisi (DWs).

**Contoh 2.1** Misalkan terdapat nilai IHSG dengan =

(5.3452, 5.3233, 5.3555, 5.3258) . Dekomposisi *wavelet haar* level 1 adalah sebagai berikut:

$$a^1 = \left( \frac{5.3452 + 5.3233}{2}, \frac{5.3555 + 5.3258}{2} \right)$$

$$d^1 = \left( \frac{5.3452 - 5.3233}{2}, \frac{5.3555 - 5.3258}{2} \right)$$

Sehingga didapatkan

$$f \xrightarrow{H_1} (a_1|d_1)$$

$$(5.3452, 5.3233, 5.3555, 5.3258) \xrightarrow{H_1} ((5.3343, 5.3407)|(0.0110, 0.0148))$$

Dari hasil di atas data nilai tukar di dekomposisikan kedalam dua data yaitu  $a^1 = (5.3343, 5.3407)$  dan  $d^1 = (0.0110, 0.0148)$ .

- b. Transformasi dari koefisien yang pertama akan menghasilkan koefisien *approximation* dan koefisien *detail* yang kedua.

$$f \xrightarrow{H_2} (a_2 | d_2) \quad (2.5)$$

Hasil dekomposisi level 2 adalah sebagai berikut:

$$a_2 = \left( \frac{\dot{x}_1 + \dot{x}_2}{2}, \frac{\dot{x}_3 + \dot{x}_4}{2}, \dots, \frac{\dot{x}_{N-1} + \dot{x}_N}{2} \right) \quad (2.6)$$

dengan

$$\dot{x}_1 = \frac{x_1 + x_2}{2}, \dot{x}_2 = \frac{x_3 + x_4}{2}, \dots, \dot{x}_{N-1} = \frac{x_{N-3} + x_{N-2}}{2}, \dot{x}_N = \frac{x_{N-1} + x_N}{2}$$

$$d_2 = \left( \frac{\dot{x}_1 - \dot{x}_2}{2}, \frac{\dot{x}_3 - \dot{x}_4}{2}, \dots, \frac{\dot{x}_{N-1} - \dot{x}_N}{2} \right) \quad (2.7)$$

dengan

$$\dot{x}_1 = \frac{x_1 - x_2}{2}, \dot{x}_2 = \frac{x_3 - x_4}{2}, \dots, \dot{x}_{N-1} = \frac{x_{N-3} - x_{N-2}}{2}, \dot{x}_N = \frac{x_{N-1} - x_N}{2}$$

- c. Jika banyak levelnya adalah tiga, maka proses transformasi dilakukan sebanyak tiga kali.

Level maksimum dari transformasi signal *wavelet multilevel* adalah

$$level_{\max} = \frac{\ln\left(\frac{\text{panjang data(signal)}}{\text{panjang filter}}\right)}{\ln(2)} \quad (2.8)$$

Pada penelitian ini yang digunakan adalah transformasi *wavelet multilevel haar* pada transformasi signal *One Dimensional* karena penggunaannya yang lebih sederhana dengan panjang filter *haar* adalah 2.

## C. Logika *Fuzzy*

### 1. Pengertian Himpunan *Fuzzy*

Himpunan klasik (*crisp set*) adalah himpunan yang membedakan anggota dan bukan anggota dengan batasan yang jelas (Ross, 2010). Himpunan *fuzzy* merupakan perluasan dari himpunan klasik dimana keberadaan suatu elemen tidak lagi bernilai benar atau salah, tetapi akan selalu bernilai benar jika mempunyai derajat keanggotaan yang berada dalam rentang  $[0,1]$ . (Klir, 1997).

#### Definisi 2.1 (Klir dan Bo Yuan, 1995:7)

Himpunan *fuzzy*  $A$  pada himpunan *universal*  $U$  didefinisikan sebagai himpunan yang direpresentasikan dengan fungsi yang mengawankan setiap  $x \in U$  dengan bilangan *real* pada interval  $[0,1]$ , ditulis  $u_A(x) \rightarrow [0,1]$  dengan nilai  $u_A(x)$  menyatakan derajat keanggotaan  $x$  di  $A$ .

Apabila suatu elemen  $x$  dalam suatu himpunan  $A$  memiliki derajat keanggotaan *fuzzy*  $u_A(x) = 0$  berarti  $x$  tidak menjadi anggota himpunan  $A$ , dan jika derajat keanggotaan *fuzzy*  $u_A(x) = 1$  berarti  $x$  menjadi anggota penuh dari himpunan  $A$ .

Himpunan *fuzzy* memiliki 2 atribut (Sri Kusumadewi, 2010). yaitu :

- a. Linguistik, yaitu penamaan suatu himpunan yang memiliki suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami.
- b. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy* yaitu (Sri Kusumadewi, 2010).

a. Variabel *Fuzzy*

Variabel *fuzzy* merupakan variabel yang akan dibahas dalam suatu sistem *fuzzy*.

b. Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan suatu grup yang mewakili suatu kondisi atau keadaan tertentu dalam suatu variabel *fuzzy*.

c. Semesta Pembicaraan

Semesta pembicaraan atau *universal* adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*.

d. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diizinkan dan boleh dioperasikan dalam suatu himpunan *fuzzy*.

## 2. Fungsi Keanggotaan

Fungsi keanggotaan adalah suatu fungsi yang menunjukkan pemetaan titik-titik *input* kedalam derajat keanggotaan. Pendekatan fungsi merupakan salah satu cara yang dapat digunakan untuk mendapatkan derajat keanggotaan.

Ada beberapa fungsi yang dapat digunakan (Sri Kusumadewi, 2010) :

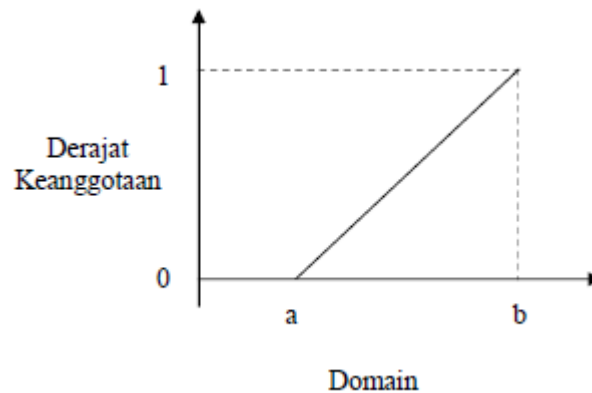
a. Representasi Linear

Pada representasi linear, pemetaan *input* derajat keanggotaanya digambarkan sebagai suatu garis lurus. Bentuk ini paling sederhana dan menjadi pilihan yang baik untuk mendekati suatu konsep yang kurang jelas. Keadaan linear himpunan *fuzzy* terdiri dari dua keadaan linear naik dan linear turun.

### 1) Representasi Linear Naik

Pada representasi linear naik, kenaikan himpunan dimulai pada nilai domain yang memiliki derajat keanggotaan [0] bergerak ke kanan menuju nilai domain yang memiliki derajat keanggotaan lebih tinggi dengan fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0 & , x < a \\ \frac{x-a}{b-a} & , a \leq x \leq b \end{cases} \quad (2.9)$$



**Gambar 2.2.** Representasi Linear Naik

Keterangan :

a = nilai domain saat derajat keanggotaan sama dengan nol

b = nilai domain saat derajat keanggotaan sama dengan satu

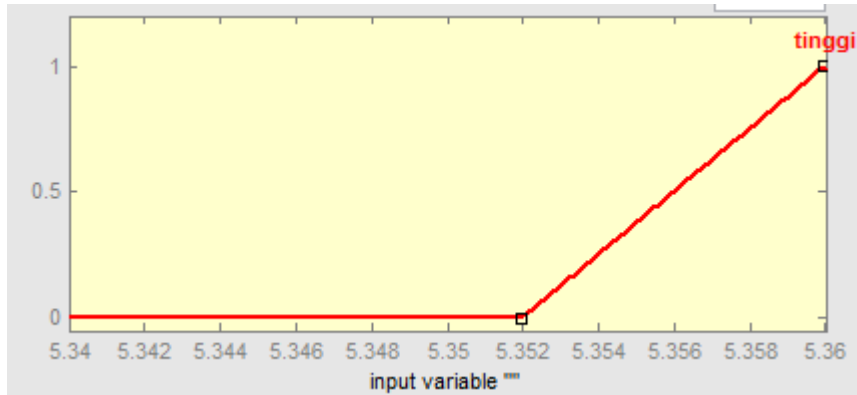
**Contoh 2.2** *Tinggi* merupakan salah satu himpunan *fuzzy* nilai IHSG dengan himpunan universal = [5.34, 5.36] yang mempunyai fungsi keanggotaan:

$$\mu_{tinggi}(x) = \begin{cases} 0 & ; x \leq 5.34 \\ \frac{x - 5.34}{0.02} & ; 5.34 \leq x \leq 5.36 \\ 1 & ; x \geq 5.36 \end{cases}$$

Berdasarkan dari fungsi keanggotaan diatas, dimisalkan untuk menentukan derajat

keanggotaan IHSG dengan nilai sebesar 5.354 maka dilakukan perhitungan:

$$\mu_{tinggi}(5.354) = \frac{5.354-5.34}{0.02} = \frac{0.014}{0.02} = 0.7$$

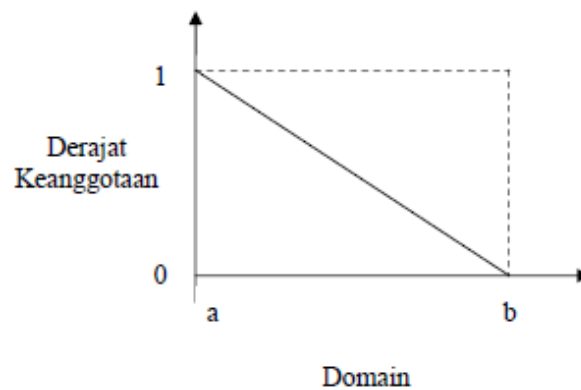


**Gambar 2.3.** Contoh fungsi keanggotaan linear naik

## 2) Representasi Linear Turun

Pada linear turun, garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki keanggotaan lebih rendah dengan fungsi keanggotaan:

$$\mu(x) = \begin{cases} \frac{b-x}{b-a}, & a \leq x \leq b \\ 0, & x > b \end{cases} \quad (2.10)$$



**Gambar 2.4** Representasi Linear Turun

Keterangan :

a = nilai domain saat derajat keanggotaan sama dengan satu

b = nilai domain saat derajat keanggotaan sama dengan nol

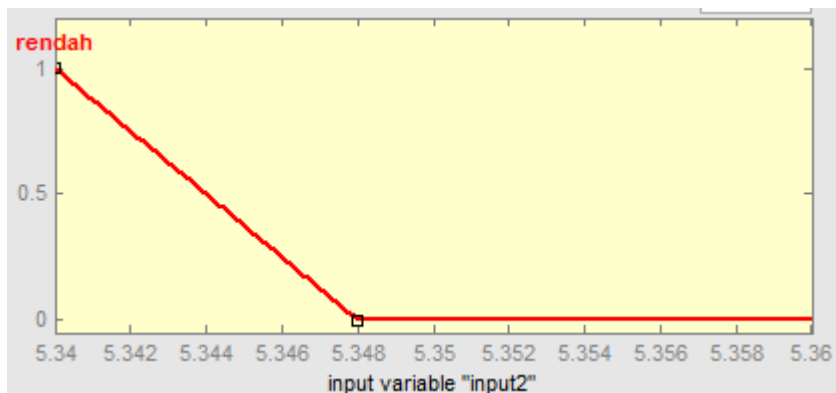
**Contoh 2.3** *Rendah* merupakan salah satu himpunan *fuzzy* nilai IHSG dengan himpunan

universal = [5.34, 5.36] yang mempunyai fungsi keanggotaan:

$$\mu_{rendah}(x) = \begin{cases} 1 & ; x \leq 5.34 \\ \frac{5.36 - x}{0.02} & ; 5.34 \leq x \leq 5.36 \\ 0 & ; x \geq 5.36 \end{cases}$$

Berdasarkan dari fungsi keanggotaan diatas, dimisalkan untuk menentukan derajat keanggotan IHSG dengan nilai sebesar 5.354 maka dilakukan perhitungan:

$$\mu_{rendah}(5.354) = \frac{5.36 - 5.354}{0.02} = \frac{0.006}{0.02} = 0.3$$

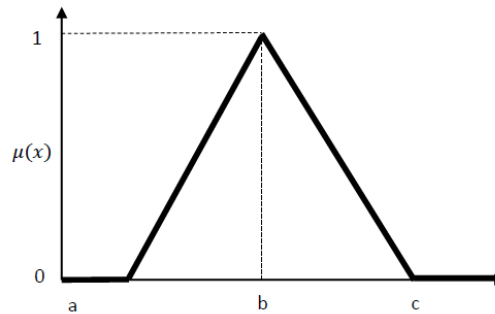


**Gambar 2.5.** Contoh fungsi keanggotaan linear turun

b. Representasi Kurva Segitiga

Representasi kurva segitiga pada dasarnya terbentuk dari gabungan 2 garis linear, yaitu linear naik dan linear turun. Kurva segitiga hanya memiliki satu nilai  $x$  dengan derajat keanggotaan tertinggi [1], hal tersebut terjadi ketika  $x=b$ . Nilai yang

tersebar dipersekitaran  $b$  memiliki perubahan derajat keanggotaan menurun dengan menjauhi 1. Berikut adalah gambar representasi kurva segitiga.



**Gambar 2.6.** Representasi Kurva Segitiga

Fungsi keanggotaannya adalah

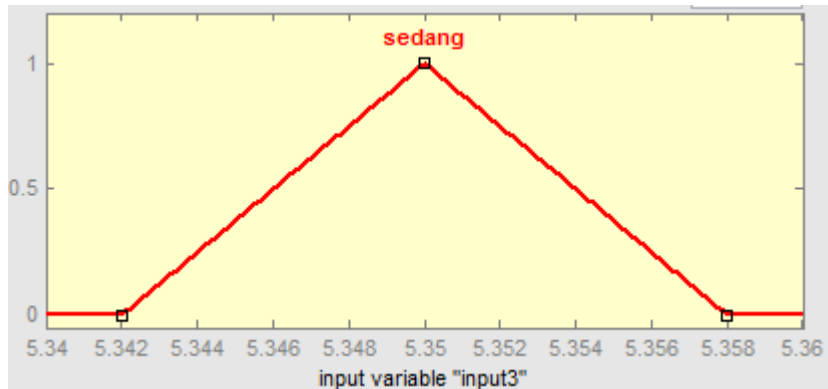
$$\mu(x) = \begin{cases} 0 & , x \leq a \text{ dan } x \geq c \\ \frac{x-a}{b-a} & , a \leq x \leq b \\ \frac{b-x}{c-b} & , b \leq x < c \end{cases} \quad (2.11)$$

**Contoh 2.4** *Sedang* merupakan salah satu himpunan *fuzzy* nilai IHSG dengan himpunan universal  $= [5.34, 5.36]$  yang mempunyai fungsi keanggotaan:

$$\mu_{sedang}(x) = \begin{cases} 0 & ; x \leq 5.34 \text{ dan } x \geq 5.36 \\ \frac{x - 5.34}{0.01} & ; 5.34 \leq x \leq 5.35 \\ \frac{5.35 - x}{0.01} & ; 5.35 \leq x \leq 5.36 \end{cases}$$

Berdasarkan dari fungsi keanggotaan diatas, dimisalkan untuk menentukan derajat keanggotaan IHSG dengan nilai sebesar 5.344 maka dilakukan perhitungan:

$$\mu_{sedang}(5.344) = \frac{5.35 - 5.344}{0.01} = \frac{0.006}{0.01} = 0.6$$

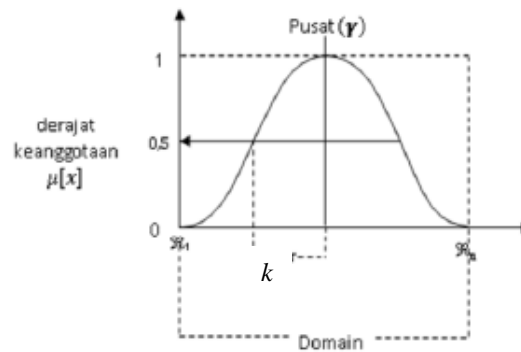


**Gambar 2.7.** Contoh fungsi keanggotaan kurva segitiga

c. Kurva Gauss

Kurva Gauss merupakan kurva berbentuk lonceng dengan derajat keanggotaan 1 terletak pada pusat dengan domain  $\gamma$ , dan lebar kurva  $k$  seperti pada gambar berikut:

$$G(x; k; \gamma) = e^{-\frac{(x-\gamma)^2}{2k^2}} \quad (2.12)$$



**Gambar 2.8.** Representasi Kurva Gauss

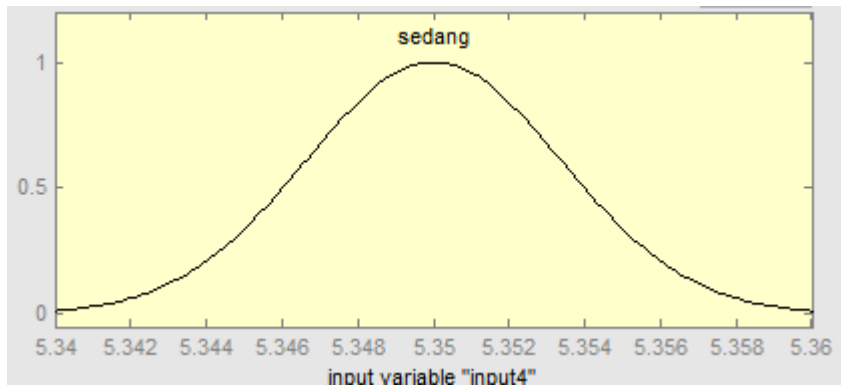
**Contoh 2.5** *Sedang* merupakan salah satu himpunan fuzzy nilai IHSG dengan himpunan universal  $= [5.34, 5.36]$  yang mempunyai fungsi keanggotaan:

$$G(x, 0.0034, 5.35) = e^{-\frac{(x-5.35)^2}{2(0.0034)^2}}$$

Berdasarkan dari fungsi keanggotaan diatas, dimisalkan untuk menentukan derajat keanggotaan IHSG dengan nilai sebesar 5.354 maka dilakukan perhitungan:

$$G(5.354, 0.0034, 5.35) = e^{-\frac{(5.354-5.35)^2}{2(0.0034)^2}}$$

$$G(5.354, 0.0034, 5.35) = 0.500553$$



**Gambar 2.9.** Contoh fungsi keanggotaan GAUSS

### 3. Operator *Fuzzy*

Terdapat 3 operator dasar yang diciptakan oleh Zadeh yaitu (Sri Kusumadewi, 2010:23)

#### a. Operator *AND* ( $\cap$ )

Operator AND berhubungan dengan operasi interseksi pada 2 himpunan yang berisi elemen-elemen yang terdapat pada kedua himpunan tersebut.  $\alpha$ -predikat merupakan hasil operasi dari operator AND yang didapat berdasarkan nilai keanggotaan terkecil antar elemen pada himpunan yang bersangkutan (Cox, 1994).

Operator AND didefinisikan sebagai berikut:

$$\mu_{A \cap B}(x, y) = \min(\mu_A(x), \mu_B(y)), \forall x, y \in U \quad (2.13)$$

b. Operator *OR* ( $\cup$ )

Operator *OR* merupakan operator yang berhubungan dengan operasi *union* pada himpunan *a – predikat* sebagai hasil dengan operator *OR* diperoleh dengan mengambil derajat keanggotaan terbesar antar elemen pada himpunan-himpunan yang bersangkutan. Misalkan A dan B adalah himpunan *fuzzy* pada  $U$ , maka himpunan *fuzzy*  $A \cup B$  didefinisikan dengan fungsi keanggotaan berikut.

$$\mu_{A \cup B}(x, y) = \max(\mu_A(x), \mu_B(y)), \forall x, y \in U \quad (2.14)$$

c. Operator NOT

Operator NOT merupakan operator yang berhubungan dengan operasi komplemen pada himpunan *a – predikat* sebagai hasil dengan operator NOT diperoleh dengan mengurangi derajat keanggotaan elemen pada himpunan yang bersangkutan dari 1. Misalkan A adalah himpunan *fuzzy* pada  $U$ . Sedangkan  $A'$  merupakan komplemen dari suatu himpunan *fuzzy* A, maka himpunan *fuzzy*  $A'$  didefinisikan dengan fungsi keanggotaan berikut:

$$\mu_{A'}(x) = 1 - \mu_A(x) \quad (2.15)$$

#### 4. Logika Fuzzy

Logika *fuzzy* merupakan perluasan dari logika klasik. Proposisi pada logika klasik hanya mengenal benar atau salah dengan proposisi nilai 0 atau 1. Sedangkan logika *fuzzy* menyamaratakan 2 nilai logika klasik dengan membiarkan proposisi nilai

kebenaran pada interval  $[0,1]$  (Wang, 1997).

Alasan digunakannya logika *fuzzy* antara lain (Sri Kusumadewi, 2010):

- a. Konsep logika *fuzzy* mudah dimengerti dengan konsep matematis sebagai dasar dari penalaran *fuzzy* yang sangat sederhana dan mudah dimengerti.
- b. Logika *fuzzy* sangat fleksibel, artinya mampu beradaptasi dengan perubahan-perubahan, dan ketidakpastian yang menyertai permasalahan.
- c. Logika *fuzzy* memiliki toleransi terhadap data-data yang tidak tepat. Jika diberikan sekelompok data yang cukup homogen, dan kemudian ada beberapa data yang “eksklusif”, maka logika *fuzzy* memiliki kemampuan untuk menangani data eksklusif.
- d. Logika *fuzzy* mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks.
- e. Logika *fuzzy* dapat mengaplikasikan pengalaman-pengalaman para ahli secara langsung tanpa harus melalui proses pelatihan. Dalam hal ini, sering dikenal dengan nama *Fuzzy Expert System* menjadi bagian terpenting.
- f. Logika *fuzzy* dapat bekerjasama dengan teknik-teknik kendali secara konvensional. Hal ini umumnya terjadi pada aplikasi di bidang teknik mesin maupun teknik elektro.
- g. Logika *fuzzy* didasarkan pada bahasa alami. Logika *fuzzy* menggunakan bahasa sehari-hari sehingga mudah dimengerti.

## **5. Sistem Fuzyy**

Sistem *fuzzy* dapat diartikan sebagai deskripsi linguistik (aturan *fuzzy* Jika-

Maka) yang lengkap tentang proses yang dapat dikombinasikan kedalam sistem (Wang, 1997). Ada beberapa sistem *fuzzy* yang biasa digunakan, salah satunya yaitu sistem *fuzzy* Mamdani.

a. Sistem *Fuzzy* Mamdani

Sistem Mamdani ini dikenalkan oleh Ebrahim Mamdani pada tahun 1975 atau sering disebut dengan Metode *Max-Min*. Pada sistem ini diperlukan empat tahap untuk mendapatkan *output*, antara lain (Sri Kusumadewi, 2010)

1) Pembentukan himpunan *fuzzy*

Pada metode Mamdani, variabel *input* dan variabel *output* dibagi menjadi satu atau lebih himpunan *fuzzy*.

2) Aplikasi Fungsi Implikasi

Aturan atau implikasi pada sistem ini secara umum dapat dipresentasikan sebagai berikut: Jika ( $x_1$  is  $A_1$ ) dan ( $x_2$  is  $A_2$ ) .... dan ( $x_n$  is  $A_n$ ) Maka ( $y$  is  $B$ ) dengan ( $x_1$  is  $A_1$ )..... ( $x_n$  is  $A_n$ ) menyatakan *input* sedangkan ( $y$  is  $B$ ) menyatakan *output* dengan  $x_1, \dots, x_n$  dan  $y$  menyatakan variabel dan  $A_1, \dots, A_n$  dan  $B$  menyatakan himpunan *fuzzy*. Pada metode ini, fungsi implikasi yang digunakan adalah Min.

3) Inferensi *fuzzy*

Inferensi diperoleh dari kumpulan dan korelasi antar aturan. Metode yang biasa digunakan dalam melakukan inferensi adalah metode Max (*Maximum*). Pada metode *max* solusi himpunan *fuzzy* diperoleh dengan cara mengambil nilai maksimum aturan yang kemudian digunakan untuk memodifikasi daerah *fuzzy* dan mengaplikasikannya

ke *output* dengan menggunakan operator *OR* (union/gabungan).

Secara umum dapat dituliskan :

$$\mu_{sf}[x_i] \leftarrow \max(\mu_{sf}[x_i], u_{kf}[x_i]) \quad (2.16)$$

dengan

$\mu_{sf}[x_i]$  menyatakan derajat keanggotaan solusi *fuzzy* sampai aturan ke-i.

$u_{kf}[x_i]$  menyatakan derajat keanggotaan konsekuen *fuzzy* sampai aturan ke-i.

#### 4) Penegasan ( )

Defuzzifikasi adalah komponen penting dalam pemodelan sistem *fuzzy*. Defuzzifikasi digunakan untuk menghasilkan nilai variabel solusi yang diinginkan dari suatu daerah konsekuen *fuzzy* (Setiadji, 2009).

Terdapat tiga jenis defuzzifikasi (Wang, 1997), yaitu:

##### a) *Center of Gravity* (Metode *Centroid*)

Pada metode ini, solusi tegas diperoleh dengan cara mengambil titik pusat ( $y^*$ ) daerah *fuzzy*, secara umum dirumuskan

$$y^* = \frac{\int_y y\mu(y)dy}{\int_y \mu(y)dy}; \text{ untuk variabel kontinu} \quad (2.17)$$

$$y^* = \frac{\sum_{i=1}^n y_i\mu(y_i)}{\sum_{i=1}^n \mu(y_i)}; \text{ untuk variable diskrit} \quad (2.18)$$

dengan

$y$  = nilai tegas

$\mu_y$  = derajat keanggotaan nilai tegas

##### b) *Center Average Defuzzifier* (CAD)

Defuzzifikasi ini dapat digunakan jika *output* fungsi keanggotaan dari beberapa proses *fuzzy* mempunyai bentuk yang sama. Metode ini mengambil nilai rata-rata dengan menggunakan pembobotan berupa derajat keanggotaan. Pada defuzzifikasi ini rumus yang digunakan yaitu:

$$y^* = \frac{\sum_{l=1}^M \bar{y}^l w_l}{\sum_{l=1}^M w_l} \quad (2.19)$$

dengan

$M$  = jumlah himpunan *fuzzy*

$\bar{y}$  = pusat himpunan *fuzzy* ke- $l$

$w_l$  = tinggi himpunan *fuzzy* ke- $l$

c) *Maximum* Defuzzifier

Defuzzifikasi ini dengan mengambil salah satu dari nilai-nilai variabel dimana himpunan bagian *fuzzy* memiliki nilai kebenaran maksimum sebagai nilai tegas bagi variabel *output* ( $y^*$ ).

Ada tiga jenis *maximum* defuzzifier, yaitu:

i. *Smallest of Maxima (SOM)*

Solusi tegas diperoleh dengan cara mengambil nilai terkecil dari domain yang memiliki derajat keanggotaan maksimum.

ii. *Largest of Maxima (LOM)*

Solusi tegas diperoleh dengan cara mengambil nilai terbesar dari domain yang memiliki derajat keanggotaan maksimum.

iii. *Mean of Maxima (MOM)*

Solusi tegas diperoleh dengan cara mengambil nilai rata-rata domain yang memiliki derajat keanggotaan maksimum.

$$y^* = \frac{\int_{hgt} y \, dy}{\int_{hgt} dy} \quad (2.20)$$

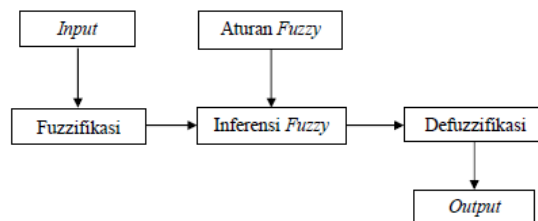
dengan

$y$  = nilai tegas

$hgt$  = himpunan semua titik dimana  $\mu_y$  mencapai nilai maksimumnya.

b. Susunan Sistem *Fuzzy*

Susunan sistem *fuzzy* dapat digambarkan pada diagram berikut:



**Gambar 2.10.** Susunan Sistem *Fuzzy*. (Wang, 1997).

Langkah-langkah dalam sistem *fuzzy* adalah:

1) Menentukan *Input* dan *Output*

*Input* merupakan variabel/data yang akan dimasukkan pada suatu sistem. *Output* merupakan hasil dari keluaran atau kesimpulan dari *input* pada suatu sistem.

2) Fuzzifikasi

Fuzzifikasi yaitu pemetaan dari himpunan tegas (sesuatu yang bernilai pasti) ke

himpunan *fuzzy* (sesuatu yang bernilai samar) dengan suatu fungsi keanggotaan. Dengan kata lain proses fuzzifikasi merupakan proses untuk mengubah variabel non-*fuzzy* menjadi variabel *fuzzy* dengan suatu fungsi keanggotaan. Melalui fungsi keanggotaan yang telah disusun maka nilai masukan tersebut menjadi informasi *fuzzy* yang selanjutnya akan digunakan untuk proses pengolahan secara *fuzzy*.

### 3) Menentukan Aturan *Fuzzy*

Sebuah pengetahuan aturan *fuzzy* dipresentasikan dalam aturan *fuzzy* Jika-Maka. Aturan *fuzzy* Jika-Maka dapat dipresentasikan dalam pernyataan berikut (Wang, 1997)

Jika  $x$  is  $A$ , maka  $y$  is  $B$ , dengan  $A$  dan  $B$  menyatakan himpunan *fuzzy* dan " $x$  is  $A$ " disebut anteseden (premis) sedangkan " $y$  is  $B$ " disebut konsekuen (kesimpulan). Aturan Jika-Maka dapat dipresentasikan pada beberapa variable anteseden dan satu variabel konsekuen dengan operator AND atau operator OR. Aturan dapat dinyatakan sebagai berikut (Wang, 1997).

Jika  $(x_1 \text{ is } A_1) \cdot (x_2 \text{ is } A_2) \cdot \dots \cdot (x_n \text{ is } A_n)$  Maka  $(y \text{ is } B)$ , dengan  $(x_1 \text{ is } A_1)$ ,  $(x_2 \text{ is } A_2) \dots \dots \dots (x_n \text{ is } A_n)$  menyatakan anteseden sedangkan  $(y \text{ is } B)$  menyatakan konsekuen dan " $\cdot$ " menyatakan operator himpunan *fuzzy* (misal AND atau OR).

### 4) Melakukan Inferensi *fuzzy*

Inferensi diperoleh dari kumpulan dan korelasi antar aturan. Metode yang biasa digunakan dalam melakukan inferensi adalah metode Max (*Maximum*).

## 5) Melakukan Defuzifikasi

Defuzifikasi adalah komponen penting dalam pemodelan system *fuzzy*. Defuzzifikasi digunakan untuk menghasilkan nilai variable solusi yang diinginkan dari suatu daerah konsekuan *fuzzy* (Setiadji, 2009).

### **D. Fuzzy C- Means**

*Fuzzy C-means clustering (FCM)* adalah suatu algoritma *clustering* data dimana setiap titik data termasuk dalam suatu *cluster* pada tingkat yang dispesifikasikan oleh sebuah derajat keanggotaan. Algoritma ini diajukan oleh Bezdek (1973).

Menurut Kusumadewi (2006) *FCM* adalah suatu teknik pengclusteran data yang mana keberadaan tiap-tiap data dalam suatu *cluster* ditentukan oleh nilai keanggotaan. Konsep dasar *FCM*, pertama kali adalah menentukan pusat *cluster* yang akan menandai lokasi rata-rata untuk tiap-tiap *cluster*.

Pada kondisi awal, pusat *cluster* ini masih belum akurat. Tiap-tiap data memiliki derajat keanggotaan untuk tiap-tiap *cluster*. Dengan cara memperbaiki pusat *cluster* dan nilai keanggotaan tiap-tiap data secara berulang, maka akan dapat dilihat bahwa pusat *cluster* akan bergerak menuju lokasi yang tepat.

Menurut Kusumadewi (Zimmerman, 1991), algoritma *FCM* diberikan sebagai berikut:

1. Tentukan:
  - a. Matriks  $X$  berukuran  $n \times m$ , dengan  $n$  = jumlah data yang akan

dicluster; dan m = jumlah variabel (kriteria).

- b. Jumlah *cluster* yang akan dibentuk = C ( $\leq 2$ ).
- c. Pangkat (pembobot) = w ( $\geq 1$ ).
- d. Maksimum iterasi.
- e. Kriteria penghentian =  $\varepsilon$  (nilai positif yang sangat kecil)
- f. Iterasi awal, t = 1, dan  $\Delta = 1$ .

2. Bentuk matriks partisi awal,  $U^0$ , sebagai berikut:

$$U = \begin{bmatrix} \mu_{11}(X_1) & \mu_{12}(X_2) & \dots & \mu_{1n}(X_n) \\ \mu_{21}(X_1) & \mu_{22}(X_2) & \dots & \mu_{2n}(X_n) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \mu_{C1}(X_1) & \mu_{C2}(X_2) & \dots & \mu_{Cn}(X_n) \end{bmatrix}$$

(matriks partisi awal biasanya dipilih secara acak).

3. Hitung pusat *cluster* ( $V$ ), untuk setiap *cluster*

$$V_{ij} = \frac{\sum_{k=1}^n (\mu_{ik})^w \cdot X_{kj}}{\sum_{k=1}^n (\mu_{ik})^w} \quad (2.21)$$

4. Perbaiki derajat keanggotaan setiap data pada setiap *cluster*

(perbaiki matriks partisi), sebagai berikut :

$$\mu_{ik} = \left[ \sum_{j=1}^C \left( \frac{d_{ik}}{d_{jk}} \right)^{2/(w-1)} \right]^{-1} \quad (2.22)$$

dengan:

$$d_{ik} = d(X_k - V_i) = \left[ \sum_{j=1}^m (X_{kj} - V_{ij}) \right]^{1/2}$$

5. Tentukan kriteria berhenti, yaitu perubahan matriks partisi pada iterasi sekarang dengan iterasi sebelumnya, sebagai berikut:

$$\Delta = \| U^t - U^{t-1} \| \quad (2.23)$$

Apabila  $\Delta \leq \varepsilon$  maka iterasi dihentikan, namun apabila  $\Delta \geq \varepsilon$ , maka naikkan iterasi ( $t=t+1$ ) dan kembali ke langkah-3.

### E. Neural Network

Jaringan syaraf tiruan (JST) atau yang biasa disebut *Artificial Neural Network* (ANN) atau *Neural Network* (NN) saja, merupakan sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf pada makhluk hidup. NN berupa suatu model sederhana dari suatu syaraf nyata dalam otak manusia seperti suatu unit *threshold* yang biner.

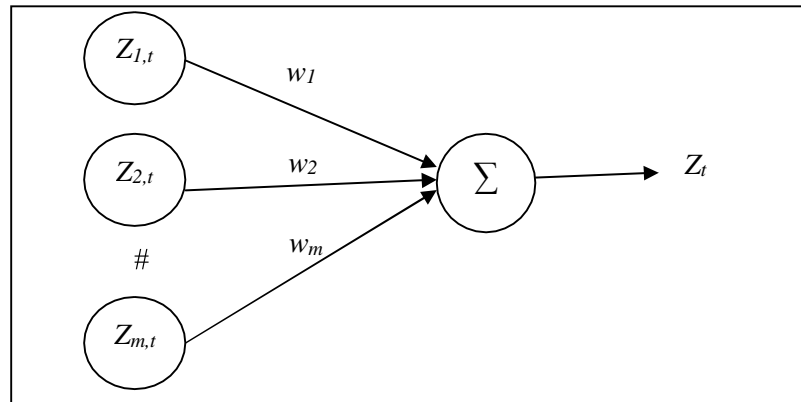
NN merupakan sebuah mesin pembelajaran yang dibangun dari sejumlah elemen pemrosesan sederhana yang disebut neuron atau *node*. Setiap neuron dihubungkan dengan neuron yang lain dengan hubungan komunikasi langsung melalui pola hubungan yang disebut arsitektur jaringan.

Bobot-bobot pada koneksi mewakili besarnya informasi yang digunakan jaringan. Metode yang digunakan untuk menentukan bobot koneksi tersebut dinamakan dengan algoritma pembelajaran. Setiap neuron mempunyai tingkat aktivasi yang merupakan fungsi dari input yang masuk. Aktivasi yang dikirim suatu neuron ke neuron lain berupa sinyal dan hanya dapat mengirim sekali dalam satu waktu, meskipun sinyal tersebut disebarkan pada beberapa neuron yang lain.

Misalkan input  $Z_{1,t}, Z_{2,t}, \dots, Z_{m,t}$  yang bersesuaian dengan sinyal dan masuk ke dalam saluran penghubung. Setiap sinyal yang masuk dikalikan dengan bobot koneksinya yaitu  $w_1, w_2, \dots, w_m$  sebelum masuk ke blok penjumlahan yang berlabel  $\Sigma$ . Kemudian blok penjumlahan akan menjumlahkan semua input terbobot dan menghasilkan sebuah nilai yaitu  $Z_{t\_in}$ .

$$Z_{t\_in} = \sum^m Z_{i,t} \cdot w_i = Z_{1,t} \cdot w_1 + Z_{2,t} \cdot w_2 + \dots + Z_{m,t} \cdot w_m \quad (2.24)$$

Aktivasi  $Z_t$  ditentukan oleh fungsi input jaringannya,  $Z_t = f(Z_{t\_in})$  dengan  $f$  merupakan fungsi aktivasi yang digunakan.



**Gambar 2.11.** Struktur jaringan syaraf tiruan dengan input  $Z_{1,t}, Z_{2,t}, \dots, Z_{m,t}$  dan bobot koneksinya  $w_1, w_2, \dots, w_m$

Secara garis besar *neural network* mempunyai dua tahap pemrosesan informasi, yaitu tahap pelatihan dan tahap pengujian.

1. Tahap Pelatihan

Tahap pelatihan dimulai dengan memasukkan pola-pola pelatihan (data

latih) ke dalam jaringan. Dengan menggunakan pola-pola ini jaringan akan mengubah-ubah bobot yang menjadi penghubung antar *node*. Pada setiap iterasi (*epoch*) dilakukan evaluasi terhadap output jaringan. Tahap ini berlangsung pada beberapa iterasi dan berhenti setelah jaringan menemukan bobot yang sesuai dan nilai eror yang diinginkan telah tercapai atau jumlah iterasi telah mencapai nilai yang ditetapkan. Selanjutnya bobot ini menjadi dasar pengetahuan pada tahap pengujian.

## 2. Tahap Pengujian

Pada tahap ini dilakukan pengujian terhadap suatu pola masukan yang belum pernah dilatihkan sebelumnya (data uji) menggunakan bobot-bobot yang telah dihasilkan pada tahap pelatihan. Diharapkan bobot-bobot hasil pelatihan yang sudah menghasilkan eror minimal juga akan memberikan eror yang kecil pada tahap pengujian. (Warsito, 2009)

## F. FIS Model Sugeno (TSK)

Sistem inferensi *fuzzy* menggunakan metode Sugeno memiliki karakteristik yaitu konsekuen tidak merupakan himpunan *fuzzy*, namun merupakan suatu persamaan linier dengan variabel-variabel sesuai dengan variabel inputnya. Metode ini diperkenalkan oleh **Takagi Sugeno Kang** (TSK) pada 1985. Aturan *fuzzy* metode Sugeno adalah sebagai berikut:

$$\text{If } Z1,t \text{ is } A1 \text{ and } Z2,t \text{ is } A2 \text{ then } f=h(Z1,t, Z2,t) \quad (2.25)$$

Ada dua model untuk sistem inferensi *fuzzy* dengan menggunakan metode

Sugeno, yaitu model Sugeno orde 0 dan model Sugeno orde 1, sebagai berikut:

### 1. Model *Fuzzy Sugeno Orde 0*

Secara umum bentuk model *fuzzy Sugeno* orde 0 adalah:

$$\text{If } (Z1,t \text{ is } A1) \bullet (Z2,t \text{ is } A2) \bullet (Z3,t \text{ is } A3) \circ \dots \circ (Zm,t \text{ is } Am) \text{ then } f=k \quad (2.26)$$

dengan  $A_m$  adalah himpunan *fuzzy* ke- $m$  sebagai anteseden,  $\circ$  adalah operator *fuzzy* (seperti *AND* atau *OR*), dan  $k$  adalah suatu konstanta (tegas) sebagai konsekuen.

### 2. Model *fuzzy Segeno Orde 1*

Secara umum bentuk *fuzzy sugeno* orde 1 adalah:

$$\text{If } (Z1,t \text{ is } A1) \bullet (Z2,t \text{ is } A2) \circ \dots \circ (Zm,t \text{ is } Am) \text{ then } f=p1 Z1,t + \dots + pm Zm,t + Z \quad (2.27)$$

dengan  $A_m$  adalah himpunan *fuzzy* ke- $m$  sebagai anteseden,  $\circ$  adalah operator *fuzzy* (seperti *AND* atau *OR*),  $p_m$  adalah suatu konstanta (tegas) ke- $m$  dan  $q$  juga merupakan konstanta dalam konsekuen. (Kusumadewi, 2006).

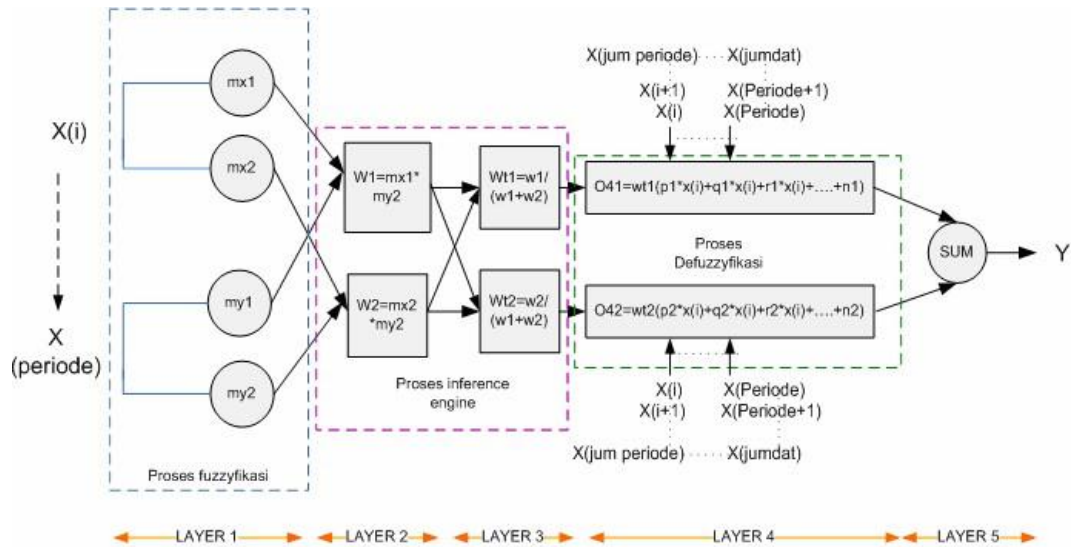
### G. *Adaptive Neuro Fuzzy Inference System (ANFIS)*

*Adaptive Neuro Fuzzy Inference Systems (ANFIS)* merupakan gabungan dari *Artificial Neural Network (ANN)* dan *Fuzzy Inference Systems (FIS)*. ANFIS adalah jaringan adaptif yang berbasis pada sistem inferensi *fuzzy* (Dewi, 2012). Dan menurut Kusumadewi (2002), ANFIS adalah suatu metode yang mana dalam melakukan penyetelan aturan digunakan algoritma pembelajaran terhadap sekumpulan data. Pada ANFIS juga memungkinkan aturan - aturan

untuk beradaptasi.

Data yang digunakan untuk proses pembelajaran (*training*) terdiri dari data input, parameter ANFIS, dan data test yang berada pada periode *training* ANFIS. *Training* pada ANFIS dilatih dengan algoritma pelatihan *hybrid* yang terdiri atas dua langkah, yaitu langkah maju dan langkah balik (Widodo, 2005). Pada langkah maju, parameter premis tetap, sedangkan parameter konsekuensi diidentifikasi dengan metode LSE (*Least Square Estimator*). Pada langkah mundur sinyal *error* antara *output* yang diinginkan dan *output* aktual dirambatkan mundur menggunakan EBP (*Error Backpropagation*) sedangkan parameter premis diperbarui dengan metode penurunan gradient (*Gradient Descent*) (Dewi, 2012).

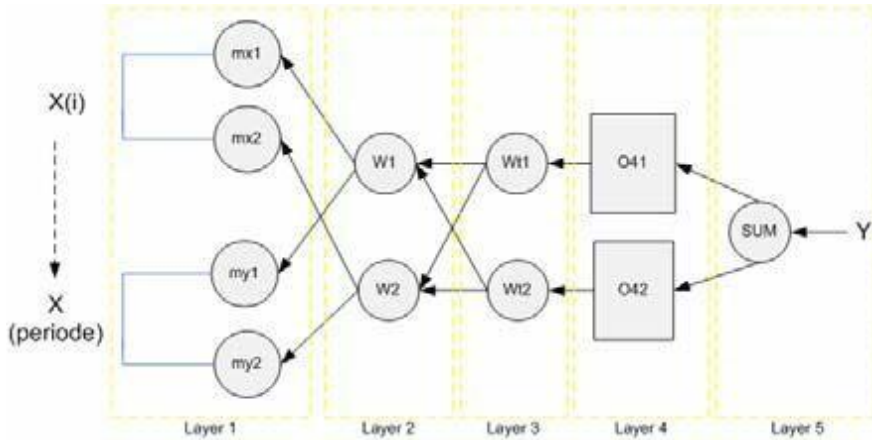
Arsitektur ANFIS dengan model Sugeno terdiri dari 5 (lima) lapis, dan setiap lapis terdapat *node*. Terdapat dua macam *node* yaitu *node* adaptif (bersimbol kotak) dan *node* tetap (bersimbol lingkaran). Proses langkah maju dapat kita lihat pada gambar 2.12.



**Gambar 2.12** Diagram langkah maju ANFIS

Pada layer pertama, data input pada masing masing periode akan dilakukan proses fuzzifikasi yaitu proses untuk memetakan inputan data kedalam himpunan *fuzzy* sesuai dengan klasifikasi yang dipilih. Kemudian pada layer kedua dan ketiga dilakukan proses *inference engine*. Pada layer keempat dilakukan proses defuzzifikasi dilakukan perhitungan mentransformasi hasil *fuzzy* ke bentuk keluaran yang *crisp*. Pada layer ini dilakukan perhitungan LSE untuk mendapatkan nilai parameter konsekuensi. Pada layer kelima dilakukan proses *summary* dari dua output pada layer keempat.

Untuk proses langkah mundur dapat kita lihat pada gambar 2.13.



**Gambar 2.13** Diagram langkah mundur ANFIS

Setelah dilakukan perhitungan alur maju, akan dilanjutkan dengan alur mundur untuk menghitung nilai *error* dari tiap layer dengan algoritma EBP (*Error Backpropagation*).

## H. Akurasi Peramalan

Untuk mengecek besar kesalahan peramalan, dapat diketahui dengan menghitung selisih antara nilai asli dengan nilai peramalannya, yang dikenal dengan nama *error* atau galat. Berikut ini adalah cara pengukuran yang digunakan untuk mengetahui besarnya kesalahan yang dihasilkan oleh suatu model peramalan:

- a. *Mean Absolute Percentage Error* (MAPE) dihitung dengan menggunakan kesalahan absolut pada tiap periode dibagi dengan nilai observasi yang nyata untuk periode itu. Selanjutnya, melakukan rata-rata kesalahan persentase absolut tersebut. Pendekatan ini berguna ketika ukuran atau besar variabel ramalan itu penting dalam mengevaluasi ketepatan ramalan. Metode MAPE

digunakan jika nilai  $Y$  positif. MAPE juga dapat. MAPE dapat dihitung dengan rumus sebagai berikut (Hanke dan Winchern, 2005):

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|Y - \hat{Y}|}{Y} \cdot 100\% \quad (2.49)$$

dengan,

$n$  = banyaknya data

$Y$  = data yang sebenarnya

$\hat{Y}$  = data hasil peramalan

Metode ini baik untuk memprediksi metode peramalan sehingga menghasilkan kesalahan ramalan yang relatif kecil. Fungsi kedua ukuran ketepatan peramalan tersebut adalah sebagai berikut:

- a) Membandingkan ketepatan dua atau lebih metode yang berbeda.
- b) Sebagai alat ukur apakah teknik yang diambil dapat dipercaya atau tidak.
- c) Membantu mencari sebuah metode yang optimal

## 1. Peramalan menggunakan ANFIS

Sistem Neuro Fuzzy berstruktur ANFIS (Adaptive Neuro Fuzzy Inference Sistem atau biasa disebut juga Adaptive Network based Fuzzy Inference Sistem) termasuk dalam kelas jaringan neural namun berdasarkan fungsinya sama dengan Fuzzy Inference System. Pada sistem Neuro Fuzzy, proses belajar pada neural network dengan sejumlah pasangan data yang berguna untuk memperbaharui parameter-parameter Fuzzy Inference System (Fariza, 2007).

Sebagai contoh, untuk model fuzzy Sugeno Orde-Satu, aturan yang umum dengan dua aturan fuzzy IF THEN sebagai berikut:

aturan 1 : IF x is  $A_1$  AND y is  $B_1$ , THEN  $f_1 = p_1x + q_1y + r_1$ ;

aturan 2 : IF x is  $A_2$  AND y is  $B_2$ , THEN  $f_2 = p_2x + q_2y + r_2$ ;

dengan x dan y adalah masukan tegas pada node ke i,  $A_i$  dan  $B_i$  adalah label linguistik (rendah, sedang, tinggi, dan lain-lain) yang dinyatakan dengan fungsi keanggotaan yang sesuai, sedangkan  $p_i$ ,  $q_i$ , dan  $r_i$  adalah parameter consequent ( $i = 1$  atau  $2$ ) (Rosyadi, 2011).

Data yang digunakan untuk proses pembelajaran (training) terdiri dari data masukan, parameter ANFIS, dan data test yang berada pada periode training ANFIS yang kemudian dilakukan proses pembelajaran terhadap data-data tersebut sehingga nantinya diperoleh output berupa hasil prediksi.

Training dengan ANFIS menggunakan algoritma belajar hybrid, dimana dilakukan penggabungan metode Least-Squares Estimator (LSE) untuk menghitung nilai konsekuen pada alur maju dan menggunakan *Error Backpropagation* (EBP) dan

gradient descent pada alur mundur untuk menghitung *error* yang terjadi pada tiap layer (Fariza, 2007).

ANFIS terdiri dari lima layer. Pada layer pertama terdiri dari proses fuzzifikasi dimana data masukan dan target dipetakan dalam derajat keanggotaannya. Pada layer kedua dan ketiga dilakukan proses inferensi yang digunakan untuk menentukan rule fuzzy menggunakan inferensi Sugeno dimana hasilnya akan diproses pada perhitungan selanjutnya. Pada layer 4 dilakukan proses pencarian nilai consequent dengan menggunakan LSE. Pada layer 5 dilakukan proses summary dari dua keluaran pada layer 4.

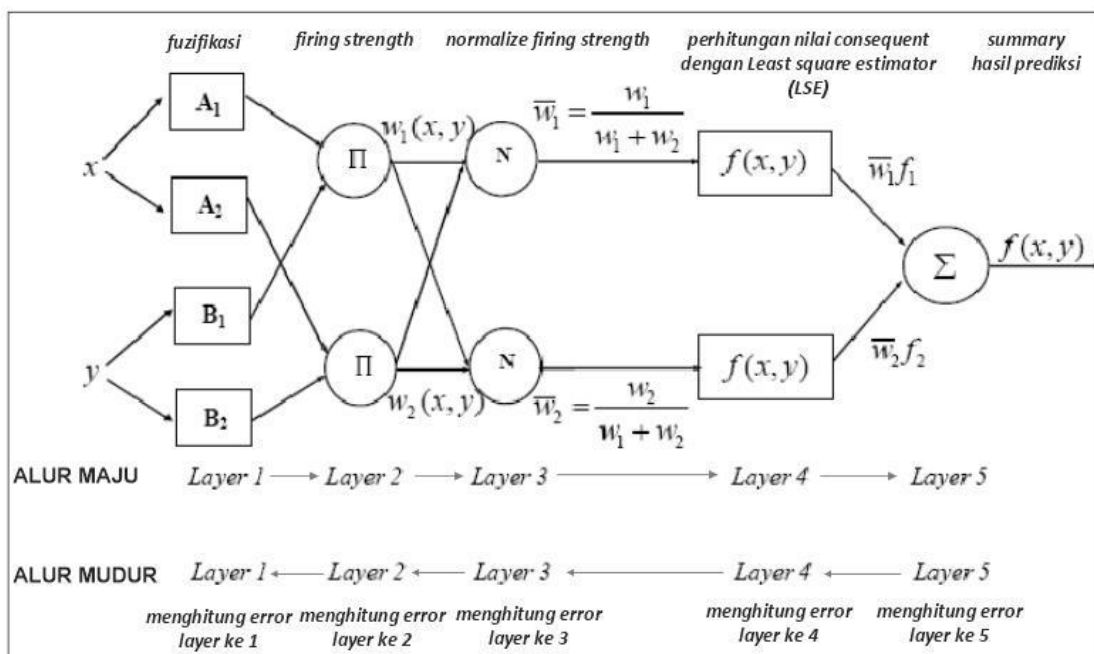
Pada ANFIS, Fuzzy Inference System (FIS) terletak pada layer 1, 2, 3 dan 4 dimana FIS adalah sebagai penentu hidden node yang terdapat pada sistem neural network (Fariza, 2007).

Setelah perhitungan alur maju dilakukan perhitungan alur mundur untuk menghitung nilai *error* tiap layer dan mengubah nilai parameter masukan menggunakan gradient descent. Proses perhitungan di atas akan berulang terus menerus sampai nilai *error* memenuhi nilai *error* maksimum yang telah ditentukan. Alur proses dari sebuah sistem ANFIS yang terdiri dari lima layer digambarkan pada Gambar 2.14. Berikut ini adalah algoritma Adaptive Neuro Fuzzy Inference System yang digunakan untuk memprediksi data runtun waktu (Mordjaoui, et. al, 2011):

1. Melakukan inisialisasi terhadap parameter ANFIS, yaitu laju pembelajaran ( $\eta$ ), momentum ( $mc$ ), batasan kesalahan ( $err$ ), dan maksimum iterasi (Max Epoch).

2. Tahap pertama yang dilakukan adalah lajur maju yang berisi beberapa tahap untuk mencari nilai consequent dari aturan yang dibuat dan melakukan penjumlahan terhadap semua masukan pada layer terakhir. Adapun tahapan lajur maju adalah sebagai berikut:

- a. Setiap node  $i$  pada layer satu merupakan node adaptive dengan fungsi node sebagai berikut:



**Gambar 2.14.** Diagram ANFIS (Rosyadi, 2011)

dengan

$x$  atau  $y$  : input dari node  $i$

$A_i$  atau  $B_i$  : sebuah label linguistik yang terhubung dengan node  $i$ .

$O_{1,i}$  : derajat keanggotaan sebuah himpunan fuzzy

- b. Setiap node  $i$  pada layer kedua berupa node tetap yang keluarannya adalah hasil dari masukan. Operator yang digunakan adalah operator AND. Tiap-tiap node

merepresentasikan  $\alpha$  predikat dari aturan ke- $i$ . Keluaran dari layer ini disebut dengan firing strength.

$$O_{2,i} = w_i = \mu A_i(x)\mu B_i(y), i = 1,2 \quad (2.28)$$

- c. Tiap- tiap *node* pada lapisan ketiga berupa *node* tetap yang merupakan hasil perhitungan rasio dari  $w$  predikat dari aturan ke- $i$  terhadap jumlah dari keseluruhan  $w$  predikat. Hasil keluaran dari layer ini dikenal dengan nama kuat penyulutan ternormalisasi (*normalized firing strength*).

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1+w_2}, i = 1,2 \quad (2.29)$$

- d. Tiap-tiap node pada layer keempat merupakan node adaptive terhadap suatu keluaran.

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i(p_i x + q_i y + r_i) = 1,2 \quad (2.30)$$

Dengan  $\bar{w}_i$  adalah *normalized firing strength* pada layer ketiga dan  $\{p_i, q_i, r_i\}$  adalah parameter-parameter pada node tersebut yang dinamakan *consequent parameters*.

- e. Menentukan konsekuen parameters dengan menggunakan *recursive least-squares estimator* (LSE resahamif). Berikut ini adalah langkah untuk menentukan nilai konsekuen dengan menggunakan LSE resahamif:

- i. Buat matrix A dengan ukuran  $n \times n$  yang berisi nilai dari keluaran pada layer keempat dan nilai  $n$  merupakan jumlah parameter keluaran pada layer keempat.

ii. Buat matrix Y dengan ukuran n x 1 yang berisi nilai dari target prediksi.

$$P_n = (A_n^T A_n)^{-1} \quad (2.31)$$

$$\theta_n = A_n A_n^T Y_n \quad (2.32)$$

iii. Melakukan pengulangan dari n+1 sampai data terakhir untuk mendapatkan nilai consequent.

$$P_{n+1} = P_n - \frac{P_n a_{n+1} a_{n+1}^T P_n}{1 + a_{n+1}^T P_n a_{n+1}} \quad (2.33)$$

$$\theta_{n+1} = \theta_n + P_{n+1} a_{n+1} (Y_{n+1} - a_{n+1}^T \theta_n) \quad (2.34)$$

f. Pada layer kelima memiliki sebuah node yang tetap yang mempunyai tugas untuk menjumlahkan nilai dari semua masukan.

$$O_{5,1} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.35)$$

g. Ketika nilai dari parameter *consequent* telah ditetapkan, maka nilai output juga dapat ditetapkan sebagai persamaan linear yang merupakan kombinasi dari parameter consequent. Nilai simbol arsitektur dinotasikan dengan  $f$ .

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_1 f \\ &= \bar{w}_1 (p_1 x + q_1 y + r_1) + \bar{w}_2 (p_1 x + q_1 y + r_1) \end{aligned} \quad (2.36)$$

4. Setelah tahap lajur maju selesai, maka selanjutnya dilakukan tahap laju mundur dengan menggunakan *Error Backpropagation* (EBP) untuk mengecek setiap *error* pada setiap layer dan menggunakan gradient descent untuk mengubah nilai

parameter masukan pada layer pertama. EBP menggunakan metode ordered derivative untuk mencari *error* pada setiap layer.

a. Menghitung nilai *error* pada layer kelima.

$$\varepsilon_{5,1} = \frac{\partial E_p}{\partial O_{4,1}} = -2 (y_p - y_p^*) \quad (2.37)$$

dengan

$y_p$  = target prediksi

$y_p^*$  = hasil prediksi

b. Menghitung nilai *error* pada layer keempat.

$$\varepsilon_{4,j} = \left( \frac{\partial E_p}{\partial O_{5,i}} \right) \left( \frac{\partial f_{5,i}}{\partial O_{4,j}} \right) = \varepsilon_{5,i} \left( \frac{\partial f_{5,i}}{\partial O_{4,j}} \right) = \varepsilon_{5,i}(1) \quad (2.38)$$

c. Menghitung nilai *error* pada layer ketiga.

$$\varepsilon_{3,j} = \left( \frac{\partial E_p}{\partial O_{5,i}} \right) \left( \frac{\partial f_{5,i}}{\partial O_{4,j}} \right) \left( \frac{\partial f_{4,j}}{\partial O_{3,j}} \right) = \varepsilon_{4,j} \left( \frac{\partial f_{4,j}}{\partial O_{3,j}} \right) \quad (2.39)$$

d. Menghitung nilai *error* pada layer kedua.

$$\varepsilon_{2,j} = \sum_{k=1}^n \left( \frac{\partial^+ E}{\partial O_{3,j}} \right) \left( \frac{\partial f_{3,j}}{\partial O_{2,k}} \right) = \sum_{k=1}^n \varepsilon_{3,j} \left( \frac{\partial f_{3,j}}{\partial O_{2,k}} \right) \quad (2.40)$$

e. Menghitung nilai *error* pada layer pertama.

$$\varepsilon_{1,ij} = \sum_{k=1}^m \left( \frac{\partial^+ E}{\partial O_{2,j}} \right) \left( \frac{\partial f_{2,k}}{\partial O_{1,ij}} \right) = \sum_{k=1}^n \varepsilon_{2,j} \left( \frac{\partial f_{2,k}}{\partial O_{1,ij}} \right) \quad (2.41)$$

f. Mengitung nilai *error* antara layer pertama dengan parameter masukan.

$$\varepsilon_{a,ij} = \sum_{k=1}^m \left( \frac{\partial^+ E}{\partial o_{1,ik}} \right) \left( \frac{\partial f_{1,ik}}{\partial o_{a,ij}} \right) \quad (2.42)$$

$$\varepsilon_{c,ij} = \sum_{k=1}^m \left( \frac{\partial^+ E}{\partial o_{1,ik}} \right) \left( \frac{\partial f_{1,ik}}{\partial o_{c,ij}} \right) \quad (2.43)$$

g. Mengubah nilai parameter masukan pada layer pertama dengan menggunakan gradient descent.

$$\Delta a_{ij} = \eta \varepsilon_{a,ij} x_i \quad (2.44)$$

$$\Delta c_{ij} = \eta \varepsilon_{c,ij} x_i \quad (2.45)$$

$$a_{ij} = a_{ij}^{lama} + \Delta a_{ij} \quad (2.46)$$

$$c_{ij} = c_{ij}^{lama} + \Delta c_{ij} \quad (2.47)$$

dengan  $a$  adalah mean,  $c$  adalah standar deviasi, dan  $\eta$  adalah laju pembelajaran

5. Menghitung jumlah kuadrat *error* (SSE) pada layer ke L data ke-p,  $1 \leq p \leq N$ .

$$E_p = \sum_{k=1}^{N(L)} (d_k^p - x_{L,k}^p)^2 \quad (2.48)$$

6. Ulangi proses iterasi hingga nilai epoch < Max Epoch dan Ep > batasan

7. kesalahan (err).