

## **BAB II**

### **KAJIAN TEORI**

Penelitian ini membahas tentang prediksi nilai tukar rupiah terhadap dollar Amerika menggunakan model *Fuzzy Backpropagation Neural Network*, dimana bobot dari model *Fuzzy Backpropagation Neural Network* tersebut dioptimasi menggunakan Algoritma Genetika yang kemudian bobot hasil optimasi digunakan pada proses prediksi. Sebelum melakukan proses prediksi ada beberapa hal yang harus diketahui terlebih dahulu, yaitu:

#### **A. Valuta Asing**

Menurut Hady dan Hamdy (2004: 28), Valuta asing (valas) atau *foreign exchange (forex)* adalah mata uang asing yang difungsikan sebagai alat pembayaran untuk membiayai transaksi ekonomi keuangan internasional dan juga mempunyai catatan kurs resmi pada Bank Sentral.

Pasar valuta asing atau sering disebut dengan istilah *foreign exchange market* merupakan pasar di mana transaksi valuta asing dilakukan antar negara. Transaksi dapat dilakukan oleh suatu badan/perusahaan atau secara perorangan dengan berbagai tujuan. Dalam setiap kali melakukan transaksi valuta asing, maka digunakan kurs (nilai tukar) (Kasmir, 2013 : 215).

Nilai tukar mata uang atau yang sering disebut dengan kurs adalah harga satu unit mata uang asing dalam mata uang domestik atau dapat juga dikatakan harga satu unit mata uang domestik terhadap mata uang asing (Firdaus & Ariyanti, 2011: 131).

Dalam perdagangan valas internasional hanya mata uang yang tergolong “*convertible currencies*” yang sering diperdagangkan, sedangkan yang tidak termasuk dalam golongan tersebut jarang diperdagangkan. Berikut ini golongan mata uang yang termasuk dalam *convertible currencies* (Kasmir, 2013 : 216) antara lain: Dollar Amerika Serikat (USD), France Perancis (FRF), Yen Jepang (JPN), France Swiss (SFR), Dollar Australia (AUD), Dollar Canada (CAD), Deutch Mark Jerman (DM), Dollar Singapura (SGD), Dollar Hongkong (HKD), Poundsterling Inggris (GBP).

Di samping dapat dilakukan antar negara transaksi valas juga dapat dilakukan antar bank dengan nasabahnya seperti transaksi uang kertas asing (*bank notes*), *travellers cheque*, giro valas, transfer ke luar negeri atau kegiatan mata uang asing lainnya. Dalam transaksi ini bank menggunakan kurs jual dan kurs beli di mana penggunaan kurs dapat dilakukan sebagai berikut (Kasmir, 2013 : 217):

- a. Kurs jual pada saat bank menjual dan nasabah membeli.

Contoh 2.1: 1 USD = Rp 13.483,00

- b. Kurs beli pada saat bank membeli dan nasabah menjual.

Contoh 2.2: 1 Rp = 0,000075855 USD

## **1. Faktor yang Mempengaruhi Nilai Tukar Rupiah terhadap Dollar Amerika**

Dari beberapa penelitian yang pernah dilakukan ada beberapa faktor yang mempengaruhi nilai tukar rupiah terhadap dollar Amerika. Berikut penelitian yang pernah dilakukan:

Wibowo dan Amir (2008), meneliti faktor-faktor yang mempengaruhi nilai tukar rupiah terhadap dollar Amerika dengan variabel yang diteliti adalah nilai tukar rupiah terhadap dollar Amerika, *Wholesale Price Index* Indonesia dan Amerika, selisih jumlah uang beredar (JUB) Indonesia dan Amerika, selisih PDB *riil* Indonesia dan Amerika, selisih tingkat suku bunga Indonesia dan Amerika, dan selisih neraca perdagangan Indonesia dan Amerika. Penelitian tersebut memberikan hasil bahwa variabel-variabel yang mempengaruhi nilai tukar rupiah terhadap dollar Amerika adalah selisih pendapatan *riil* Indonesia dan Amerika, selisih inflasi Indonesia dan Amerika, selisih tingkat suku bunga Indonesia dan Amerika, serta nilai tukar rupiah terhadap USD satu bulan sebelumnya (*lag -1*).

Jika Wibowo menggunakan data selisih Indonesia dengan Amerika, pada penelitian Triyono (2008) hanya menggunakan data-data dari Indonesia. Triyono meneliti pengaruh jumlah uang beredar (JUB), inflasi, suku bunga Indonesia (SBI), dan impor pada nilai tukar rupiah terhadap dollar Amerika. Penelitian tersebut menghasilkan bahwa jumlah uang beredar (JUB), inflasi, dan suku bunga Indonesia (SBI) secara signifikan mempengaruhi nilai tukar rupiah terhadap dollar Amerika.

Noor (2011), meneliti tentang pengaruh inflasi, suku bunga Indonesia, dan jumlah uang beredar terhadap nilai tukar rupiah terhadap dollar Amerika. Pada penelitian tersebut menunjukkan di mana ketiga faktor yang digunakan secara bersama ditambah faktor perubahan nilai tukar pada periode atau kurun waktu sebelumnya, akan memberikan pengaruh pada perubahan nilai tukar.

Oktavia dkk (2013), meneliti tentang pengaruh jumlah uang beredar (JUB), pendapatan Indonesia, suku bunga domestik, inflasi dan neraca perdagangan terhadap kurs di Indonesia. Pada penelitian tersebut menghasilkan bahwa jumlah uang beredar, pendapatan Indonesia, suku bunga domestik, inflasi dan neraca perdagangan secara bersama-sama berpengaruh signifikan terhadap kurs di Indonesia.

Muchlas dan Alamsyah (2015), meneliti pengaruh inflasi, tingkat suku bunga, jumlah uang beredar (JUB), *Gross Domestic Product (GDP)*, *Balance of Payment* (BOP) terhadap pergerakan kurs IDR/USD. Pada penelitian tersebut membuktikan bahwa secara bersama-sama atau sendiri-sendiri inflasi, tingkat suku bunga, JUB, BOP berpengaruh terhadap pergerakan rupiah terhadap dolar Amerika.

Dari beberapa penelitian diatas maka pada penelitian ini variabel-variabel yang akan digunakan untuk memprediksi nilai tukar rupiah terhadap dollar Amerika adalah nilai tukar (kurs) rupiah terhadap dollar Amerika pada kurun waktu sebelumnya, inflasi di Indonesia, jumlah uang beredar (JUB) di Indonesia, dan tingkat suku bunga Indonesia (SBI).

### **a. Inflasi**

Inflasi adalah suatu keadaan dimana harga-harga barang secara umum mengalami kenaikan terus menerus. Hal ini menunjukkan terjadinya perubahan nilai mata uang. Perubahan nilai mata uang ini berbanding terbalik dengan laju inflasi. Semakin tinggi inflasi maka semakin banyak penurunan nilai mata uang.

### **b. Tingkat Suku Bunga**

Tingkat Suku Bunga adalah harga dari peminjaman uang untuk menggunakan daya belinya dan biasanya dinyatakan dalam persen (%). Suku bunga suatu negara yang rendah akan menyebabkan modal dalam negeri mengalir ke luar negeri dan pada suku bunga yang tinggi akan menyebabkan aliran modal masuk ke dalam negeri. Jika lebih banyak modal yang mengalir ke suatu negara, permintaan atas mata uang negara tersebut akan bertambah dan nilai mata uang tersebut akan menguat.

### **c. Jumlah Uang Beredar**

Terdapat dua jenis jumlah uang beredar, yaitu jumlah uang beredar dalam arti sempit (M1) dan jumlah uang beredar dalam arti luas (M2). Dalam arti sempit jumlah uang beredar meliputi uang kartal dan uang giral. Uang kartal adalah alat pembayaran sah yang dikeluarkan oleh pemerintah suatu negara melalui bank sentral yang berupa uang logam dan uang kertas. Sedangkan uang giral adalah alat pembayaran berupa bilyet giro, cek, dan pemindahan telegrafis yang dikeluarkan oleh bank kepada seseorang atau badan karena mempunyai simpanan rekening di bank yang bersangkutan. Sedangkan dalam arti luas jumlah uang beredar meliputi M1 ditambah dengan uang kuasi. Uang kuasi adalah simpanan masyarakat pada bank umum dalam bentuk deposit berjangka dan tabungan. Jika jumlah uang beredar tinggi maka pemerintah akan menurunkan tingkat bunga dan merangsang investasi keluar negeri sehingga terjadi aliran modal keluar sehingga kurs valuta asing naik (apresiasi).

## B. Analisis *Time Series*

Data *time series* adalah nilai-nilai suatu variabel yang berurutan menurut waktu, dapat berupa dalam jam, hari, minggu, bulan, atau tahun. Contoh dari data *time series* adalah data nilai tukar rupiah terhadap dollar Amerika. Pada data *time series* dapat dilakukan suatu peramalan dengan menganalisa pola hubungan data *time series*. Beberapa teknik yang dapat digunakan dalam peramalan *time series* adalah *Moving Average*, *Regresi*, *Autoregresif Integrated Moving Average* (ARIMA), *Neural Network*, dan Algoritma Genetika.

### 1. Autokorelasi

*Autokorelasi* merupakan korelasi atau hubungan antar pengamatan dari sebuah data *time series* untuk selang waktu (*lag*) yang berlainan.

Menurut Montgomery dkk (2007: 30), Koefisien *autokorelasi* pada *lag k* ( $\rho_k$ ) antara pengamatan  $Y_t$  dan  $Y_{t+k}$  pada populasi dinyatakan dalam bentuk sebagai berikut:

$$\rho_k = \frac{Cov(Y_t, Y_{t+k})}{\sqrt{Var(Y_t)}\sqrt{Var(Y_{t+k})}} = \frac{E(Y_t - \mu)(Y_{t+k} - \mu)}{\sqrt{E(Y_t - \mu)^2}\sqrt{E(Y_{t+k} - \mu)^2}} = \frac{\gamma_k}{\gamma_0} \quad (2.1)$$

dimana  $Var(Y_t) = Var(Y_{t+k}) = \gamma_0$

dengan

$\mu$  = rata-rata

$\gamma_k$  = *autokovariansi* pada *lag-k*

$\rho_k$  = *autokorelasi* pada *lag-k*

$t$  = waktu pengamatan,  $t = 1, 2, 3, \dots$

$Y_t$  = pengamatan pada saat  $t$

$Y_{t+k}$  = pengamatan pada saat  $t+k$

Nilai-nilai  $\rho_k$  pada saat  $k = 1, 2, \dots, n$  disebut fungsi *autokorelasi* (*Autocorrelation Function* atau ACF). Menurut Markridakis (1999: 339), fungsi *autokorelasi* dapat diperkirakan dari fungsi *autokorelasi* sampel yang didefinisikan dengan:

$$r_k = \frac{\sum_{t=1}^{n-k} (Y_t - \bar{Y})(Y_{t+k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2} \quad (2.2)$$

dengan

$r_k$  = *autokorelasi* pada *lag-k*

$k$  = selisih waktu,  $k=1, 2, 3, \dots$

$n$  = jumlah observasi

$Y_t$  = pengamatan pada waktu ke- $t$

$Y_{t+k}$  = pengamatan pada waktu ke  $t+k$

$\bar{Y}$  = rata-rata pengamatan

Nilai *autokorelasi* berkisar antar -1 sampai 1. Jika nilai *autokorelasi* tepat  $\pm 1$  atau mendekati, dapat disimpulkan terdapat hubungan yang tinggi antara data *time series* tersebut dalam *lag* yang berlainan. Jika nilai *autokorelasi* adalah 0, maka tidak terdapat hubungan dari data *time series* tersebut. Untuk mengetahui suatu *autokorelasi* signifikan atau tidak dapat menggunakan suatu pengujian dengan hipotesis sebagai berikut:

$H_0: \rho_k = 0$  (koefisien *autokorelasi* tidak signifikan)

$H_1: \rho_k \neq 0$  (koefisien *autokorelasi* signifikan)

Statistik uji yang digunakan adalah

$$t_{hitung} = \frac{r_k}{SE(r_k)} \quad (2.3)$$

dengan  $SE$  adalah standar *error* yang didefinisikan

$$SE(r_k) = \sqrt{\frac{1 + 2 \sum_{i=1}^{k-1} r_i^2}{n}} \quad (2.4)$$

dengan

$SE(r_k)$  = standar *error* koefisien *autokorelasi* pada *lag*  $k$

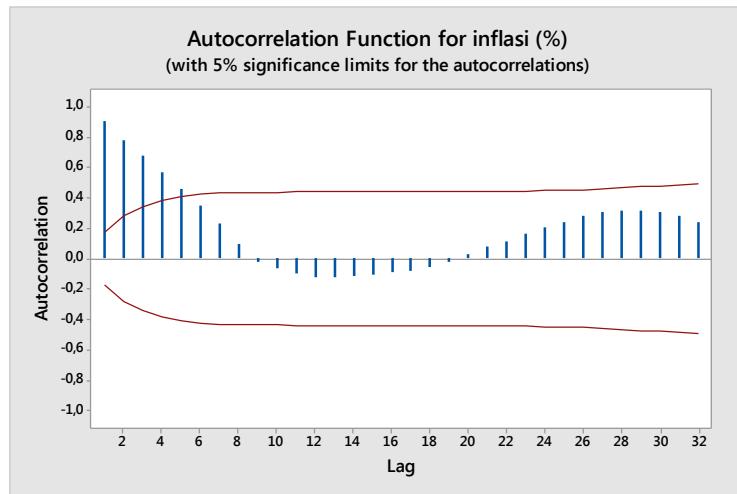
$r_k$  = koefisien korelasi pada *lag*  $k$

$n$  = banyak pengamatan

Kriteria keputusan tolak  $H_0$  jika  $t_{hitung} > t_{\frac{\alpha}{2}, n-1}$ . Signifikansi *autokorelasi* juga dapat dilihat dengan selang kepercayaan  $r$  dengan pusat 0. Selang kepercayaan  $r_k$  dapat dihitung dengan rumus:

$$0 \pm t_{n-1}(\alpha/2) \times SE(r_k) \quad (2.5)$$

Selang kepercayaan  $r_k$  dapat direpresentasikan dalam sebuah plot *autokorelasi* dengan bantuan program Minitab 17. Contoh plot *autokorelasi* dapat dilihat dari Gambar 2.1. Pada gambar tersebut selang kepercayaan direpresentasikan dengan garis merah. Kriteria *autokorelasi* pada suatu *lag* dikatakan signifikan jika nilai *autokorelasi* yang direpresentasikan dengan garis vertikal berwarna biru lebih besar dari selang kepercayaan. Pada Gambar 2.1 *autokorelasi lag* yang signifikan adalah *lag* 1 sampai dengan *lag* 5.



**Gambar 2. 1** Contoh plot *autokorelasi*

Selain autokorelasi pada analisis *time series* terdapat pula fungsi *autokorelasi* parsial (PACF). PACF digunakan untuk mengukur tingkat korelasi antara  $Y_t$  dan  $Y_{t+k}$  apabila pengaruh dari time *lag* 1, 2, 3, ...,  $k-1$  dihilangkan. PACF dinotasikan dengan  $\emptyset_{kk}$  dan dirumuskan sebagai berikut (Wei, 2006: 15).

$$\emptyset_{kk} = \frac{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{k-2} & \rho_1 \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{k-3} & \rho_2 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \dots & \rho_1 & \rho_k \end{vmatrix}}{\begin{vmatrix} 1 & \rho_1 & \rho_2 & \dots & \rho_{k-2} & \rho_1 \\ \rho_1 & 1 & \rho_1 & \dots & \rho_{k-3} & \rho_2 \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \rho_{k-1} & \rho_{k-2} & \rho_{k-3} & \dots & \rho_1 & \rho_k \end{vmatrix}}$$

## 2. Proses *white noise*

Sebuah proses  $\{Y_t\}$  disebut *white noise* jika merupakan serangkaian variabel acak yang tidak berkorelasi dan berdistribusi tertentu dengan rata-rata tetap  $E(Y_t) = \mu$  biasanya bernilai 0, variansi konstanta  $Var(Y_t) = \sigma^2$  dan  $Cov(Y_t, Y_{t+k}) = 0$  untuk semua  $k \neq 0$  (Wei, 2006: 16). Dengan demikian proses dari *white noise* ( $Y_t$ ) adalah stasioner dengan fungsi *autokovariansi*:

$$\gamma_k = \begin{cases} \sigma^2 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

fungsi *autokorelasi*

$$\rho_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

dan fungsi *autokorelasi* parsial

$$\phi_{kk} = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

karena menurut definisi  $\rho_0 = \phi_{00} = 1$  untuk semua proses, dalam *autokorelasi* dan *autokorelasi* parsial hanya mengacu pada  $\rho_k$  dan  $\phi_{kk}$  untuk  $k \neq 0$ . Konsep dasar dari proses *white noise* adalah bahwa ACF dan PACF tidak berbeda signifikan dari nol.

### C. Himpunan *Fuzzy*

Himpunan *fuzzy* merupakan perluasan dari himpunan klasik (*crisp*). Pada himpunan *crisp* nilai keanggotaan yang digunakan hanya 1 dan 0. Suatu bilangan dengan nilai keanggotaan 1 menyatakan bahwa bilangan tersebut merupakan anggota dari himpunan, sedangkan bilangan yang memiliki nilai keanggotaan 0 menyatakan bahwa bilangan tersebut bukan merupakan anggota dari himpunan tersebut. Pada himpunan *fuzzy* nilai keanggotaan yang digunakan adalah bilangan *riil* pada rentang  $[0,1]$ . Suatu bilangan akan selalu menjadi anggota himpunan jika memiliki nilai keanggotaan pada rentang  $[0,1]$ .

**Definisi 2.1** (Zimmermann, 1991: 11):

Suatu himpunan *fuzzy*  $A$  dalam himpunan semesta  $U$  dinyatakan sebagai himpunan pasangan berurutan,

$$A = \{(x, \mu_A(x) | x \in U)\} \quad (2.6)$$

dengan  $\mu_A(x)$  adalah derajat keanggotaan  $x$  di himpunan *fuzzy*  $A$  yang terletak pada interval  $[0,1]$ .

Contoh 2.3 Misalkan kurs naik adalah himpunan *fuzzy* pada nilai tukar IDR/USD dengan  $U$  adalah himpunan universal nilai tukar IDR/USD [11.462 13.820], maka himpunan kurs naik dapat dinotasikan sebagai:

kurs naik =  $\{(x, \mu_{\text{kurs naik}}(x) | x \in U)\}$ , dengan

$$\mu_{\text{kurs naik}}(x) = \begin{cases} 0; & x \leq 11.462 \\ \frac{(x - 11.462)}{(13.820 - 11.462)}; & 11.462 \leq x \leq 13.820 \\ 1; & x \geq 13.820 \end{cases}$$

Menurut Kusumadewi dan Purnomo (2013 : 6), himpunan *fuzzy* memiliki 2 atribut, yaitu:

1. Linguistik, yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami, seperti : RENDAH, SEDANG, TINGGI.
2. Numeris, yaitu suatu nilai (angka) yang menunjukkan ukuran dari suatu variabel seperti: 9.231, 10.651, 13.752, dsb.

Ada beberapa hal yang perlu diketahui dalam memahami sistem *fuzzy*, (Kusumadewi & Purnomo, 2013: 6), yaitu :

1. Variabel *fuzzy*

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

Contoh 2.4 variabel *fuzzy* yang digunakan dalam penelitian ini adalah nilai tukar IDR/USD, inflasi, jumlah uang beredar, dan suku bunga Indonesia.

## 2. Semesta pembicaraan

Semesta pembicaraan adalah keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh 2.5 Semesta pembicaraan untuk variabel nilai tukar IDR/USD [5.462 17.820].

## 3. Himpunan *fuzzy*

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

Contoh 2.6 Variabel nilai tukar IDR/USD, terbagi menjadi 3 himpunan *fuzzy*, yaitu : RENDAH, SEDANG, dan TINGGI.

## 4. Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diizinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

Contoh 2.7 domain himpunan *fuzzy*

- a. RENDAH = [5462 11641]
- b. SEDANG = [8551,5 14730,5]
- c. TINGGI = [11641 17820]

## 1. Fungsi Keanggotaan

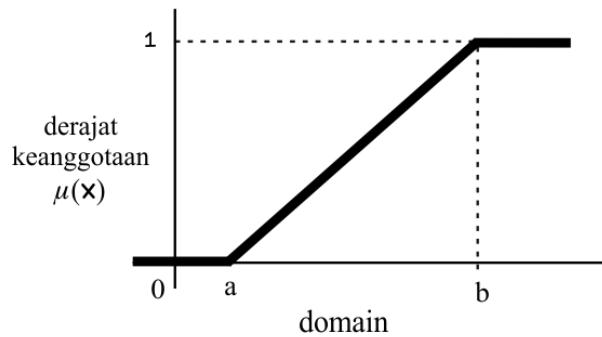
Fungsi keanggotaan (*membership function*) adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* data (himpunan *crisp*) ke dalam nilai (derajat) keanggotaan himpunan *fuzzy* yang memiliki interval antara 0 sampai 1. Salah satu cara yang dapat digunakan untuk mendapatkan derajat keanggotaan adalah dengan melalui pendekatan fungsi. Apabila  $U$  menyatakan himpunan universal dan  $A$  adalah himpunan fungsi *fuzzy* dalam  $U$ , maka derajat keanggotaan himpunan *fuzzy*  $A$  dapat dinyatakan sebagai pasangan terurut dari  $A \in U$  ke bilangan real pada interval  $[0,1]$  (Wang, 1997 : 22). Fungsi keanggotaan yang dapat dibangun dan digunakan untuk merepresentasikan himpunan *crisp* ke dalam derajat keanggotaan himpunan *fuzzy* antara lain (Kusumadewi & Hartati, 2010: 20):

### a. Representasi linear

Pada representasi linear, pemetaan *input* ke derajat anggotanya digambarkan sebagai suatu garis lurus sehingga merupakan bentuk yang paling sederhana. Terdapat 2 keadaan pada himpunan *fuzzy* yang linear, yaitu;

#### 1) Representasi linear naik

Representasi linear naik dimulai dari domain yang memiliki derajat keanggotaan nol [0] dan bergerak ke kanan menuju nilai domain yang memiliki derajat keanggotaan satu [1]. Seperti pada gambar 2.2 berikut:



**Gambar 2. 2** Representasi Linear Naik

fungsi keanggotaan:

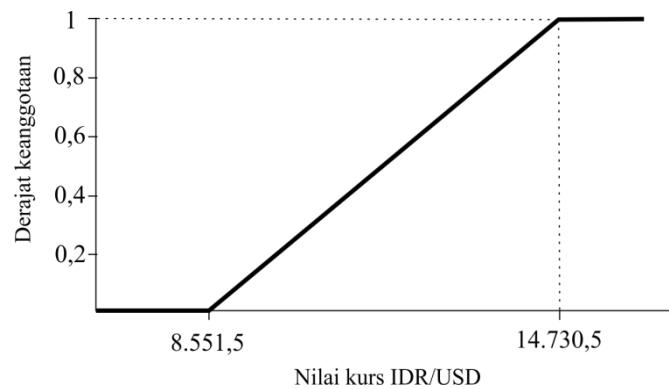
$$\mu(x) = \begin{cases} 0; & x \leq a \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & x \geq b \end{cases} \quad (2.7)$$

Contoh 2.8 Himpunan *fuzzy* nilai kurs turun dengan semesta pembicaraan [5.462 17.820] dan domain [8.551,5 14.730,5] memiliki fungsi keanggotaan

$$\mu_{\text{nilai kurs turun}}(x) = \begin{cases} 0; & x \leq 8.551,5 \\ \frac{(x - 8.551,5)}{(14.730,5 - 8.551,5)}; & 8.551,5 \leq x \leq 14.730,5 \\ 1; & x \geq 14.730,5 \end{cases}$$

Grafik representasi dari fungsi keanggotaan tersebut dapat ditunjukkan pada

Gambar 2.3 berikut:



**Gambar 2. 3** Himpunan *fuzzy* nilai kurs turun

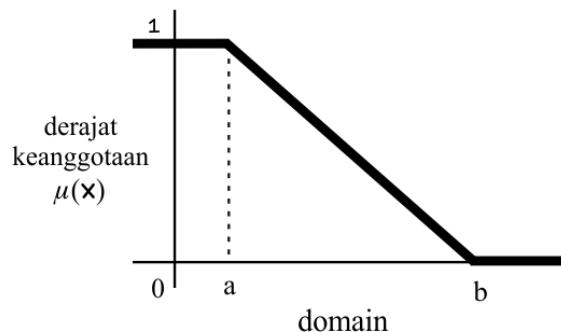
Berdasarkan fungsi keanggotaan pada gambar 2.3 akan ditentukan derajat keanggotaan himpunan *fuzzy* dari nilai kurs IDR/USD sebesar 12.350 dengan perhitungan sebagai berikut:

$$\mu_{\text{nilai kurs turun}}(12.350) = \frac{12.350 - 8.551,5}{14.730,5 - 8.551,5} = 0,6147$$

Dapat disimpulkan bahwa derajat keanggotaan nilai kurs IDR/USD sebesar 12.350 pada himpunan *fuzzy* nilai kurs turun adalah 0,6147. Sehingga nilai kurs IDR/USD sebesar 12.350 merupakan anggota himpunan *fuzzy* nilai kurs turun dengan nilai keanggotan sebesar 0,6147.

## 2) Representasi Linear Turun

Representasi nilai turun merupakan kebalikan dari representasi linear naik yaitu berupa garis lurus yang dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri kemudian bergerak menurun ke nilai domain dengan derajat keanggotaan yang lebih rendah. Seperti pada gambar 2.4 berikut:



**Gambar 2. 4** Representasi Linier Turun

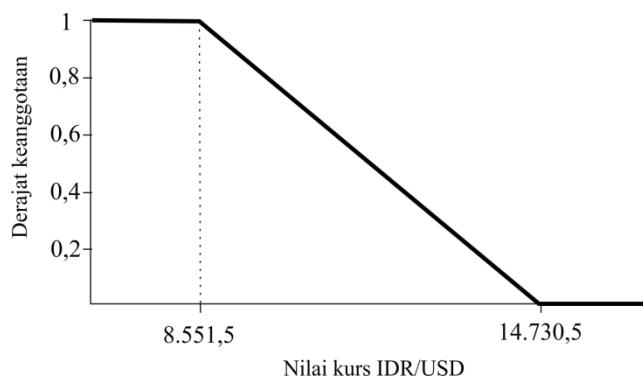
Fungsi keanggotaan:

$$\mu(x) = \begin{cases} (b - x)/(b - a); & a \leq x \leq b \\ 0; & x \geq b \end{cases} \quad (2.8)$$

Contoh 2.9 Himpunan *fuzzy* nilai kurs naik dengan semesta pembicaraan [5.462 17.820] dan domain [8.551,5 14.730,5] memiliki fungsi keanggotaan

$$\mu_{\text{nilai kurs naik}}(x) = \begin{cases} \frac{14.730,5 - x}{14.730,5 - 8.551,5}; & 8.551,5 \leq x \leq 14.730,5 \\ 0; & x \geq 14.730,5 \end{cases}$$

Grafik representasi dari fungsi keanggotaan tersebut dapat ditunjukkan pada Gambar 2.5 berikut:



**Gambar 2.5** Himpunan *fuzzy* nilai kurs naik

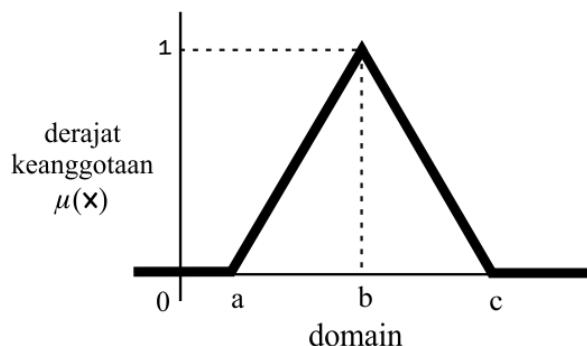
Berdasarkan fungsi keanggotaan pada gambar tersebut akan ditentukan derajat keanggotaan himpunan *fuzzy* dari nilai kurs IDR/USD sebesar 12.350 dengan perhitungan sebagai berikut:

$$\mu_{\text{nilai kurs naik}}(12.350) = \frac{14.730,5 - 12.350}{14.730,5 - 8.551,5} = 0,3853$$

Dapat disimpulkan bahwa derajat keanggotaan nilai kurs IDR/USD sebesar 12.350 pada himpunan *fuzzy* nilai kurs naik adalah 0,3853. Sehingga nilai kurs IDR/USD sebesar 12.350 merupakan anggota himpunan *fuzzy* nilai kurs naik dengan nilai keanggotaan sebesar 0,3853.

### 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 dipersekutaran  $b$  memiliki perubahan derajat keanggotaan menurun dengan menjauhi 1. Seperti pada gambar 2.6 berikut:



**Gambar 2.6** Kurva segitiga

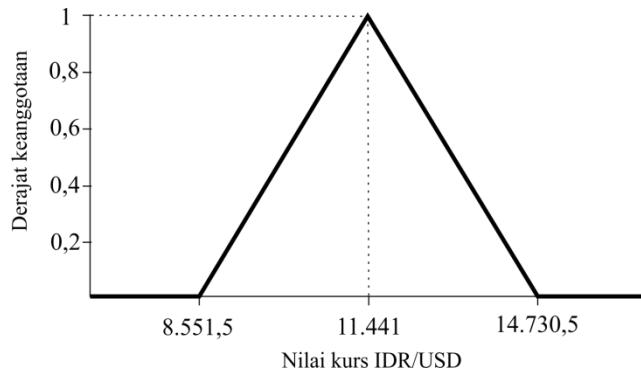
Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq c \\ (x - a)/(b - a); & a \leq x \leq b \\ (c - x)/(c - b); & b \leq x \leq c \end{cases} \quad (2.9)$$

Contoh 2.10 Sebagai contoh himpunan *fuzzy* nilai kurs sedang dengan semesta pembicaraan [5.462 17.820] dan domain [8.551,5 11.641 14.730,5] memiliki fungsi keanggotaan

$$\mu_{\text{nilai kurs sedang}}(x) = \begin{cases} 0; & x \leq 8.551,5 \text{ atau } x \geq 14.730,5 \\ \frac{x - 8.551,5}{11.641 - 8.551,5}; & 8.551,5 \leq x \leq 11.641 \\ \frac{14.730,5 - x}{14.730,5 - 11.641} & 11.641 \leq x \leq 14.730,5 \end{cases}$$

Grafik representasi dari fungsi keanggotaan tersebut dapat ditunjukkan pada Gambar 2.7 berikut:



**Gambar 2. 7** Himpunan *fuzzy* niasi kurs sedang

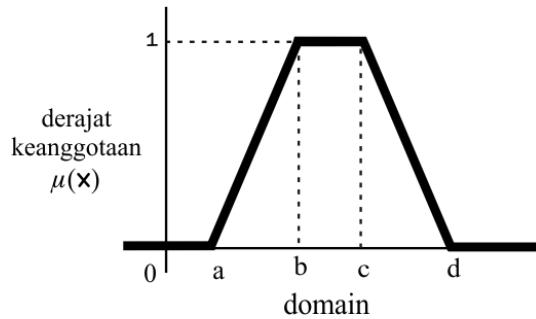
Berdasarkan fungsi keanggotaan pada gambar tersebut akan ditentukan derajat keanggotaan himpunan *fuzzy* dari nilai kurs IDR/USD sebesar 12.350 dengan perhitungan sebagai berikut:

$$\mu_{\text{niasi kurs sedang}}(12.350) = \frac{14.730,5 - 12.350}{14.730,5 - 11.641} = 0,7705$$

Dapat disimpulkan bahwa derajat keanggotaan nilai kurs IDR/USD sebesar 12.350 pada himpunan *fuzzy* niasi kurs sedang adalah 0,7705. Sehingga nilai kurs IDR/USD sebesar 12.350 merupakan anggota himpunan *fuzzy* niasi kurs sedang dengan nilai keanggotaan sebesar 0,7705.

### c. Representasi kurva trapesium

Representasi kurva trapesium merupakan perluasan dari kurva segitiga, yang memiliki lebih dari satu titik yang nilai keanggotanya 1. Seperti pada gambar 2.8 berikut:



**Gambar 2.8** Kurva trapesium

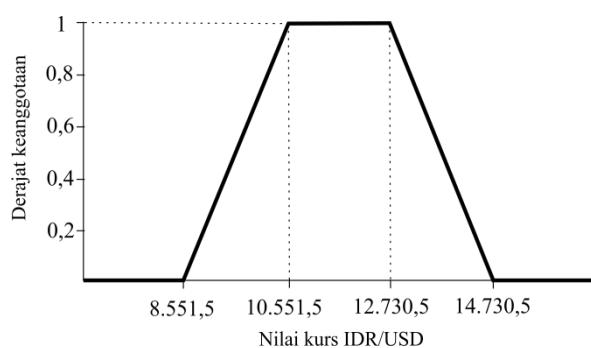
Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \text{ atau } x \geq d \\ (x - a)/(b - a); & a \leq x \leq b \\ 1; & b \leq x \leq c \\ (d - x)/(d - c); & c \leq x \leq d \end{cases} \quad (2.10)$$

Contoh 2.11 Sebagai contoh himpunan *fuzzy* nilai kurs sedang dengan semesta pembicaraan [5.462 17.820] dan domain [8.551,5 10.551,5 12.730,5 14.730,5] memiliki fungsi keanggotaan

$$\mu_{\text{nilai kurs sedang}}(x) = \begin{cases} 0; & x \leq 8.551,5 \text{ atau } x \geq 14.730,5 \\ \frac{x - 8.551,5}{10.551,5 - 8.551,5} & 8.551,5 \leq x \leq 10.551,5 \\ 1; & 10.551,5 \leq x \leq 12.730,5 \\ \frac{14.730,5 - x}{14.730,5 - 12.730,5} & 12.730,5 \leq x \leq 14.730,5 \end{cases}$$

Grafik representasi dari fungsi keanggotaan tersebut dapat ditunjukkan pada Gambar 2.9 berikut:



**Gambar 2.9** Himpunan *fuzzy* nilai kurs sedang

Berdasarkan fungsi keanggotaan pada gambar tersebut akan ditentukan derajat keanggotaan himpunan *fuzzy* dari nilai kurs IDR/USD sebesar 12.350 dengan perhitungan sebagai berikut:

$$\mu_{\text{nilai kurs sedang}}(12.350) = 1$$

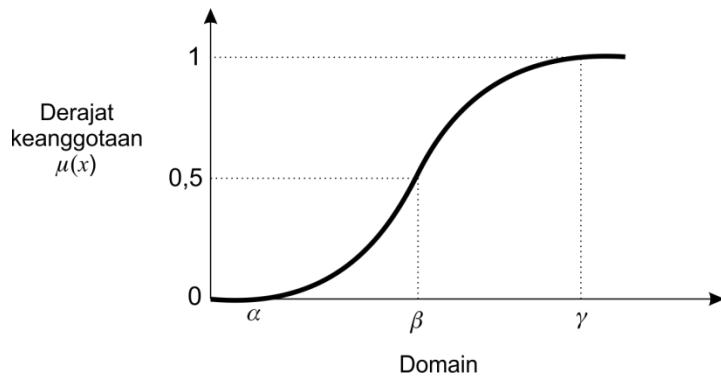
Dapat disimpulkan bahwa derajat keanggotaan nilai kurs IDR/USD 12.350 pada himpunan *fuzzy* nilai kurs sedang adalah 1. Sehingga nilai kurs IDR/USD sebesar 12.350 merupakan anggota himpunan *fuzzy* nilai kurs sedang dengan nilai keanggotaan sebesar 1.

#### **d. Representasi kurva-S**

Kurva-*S* atau *sigmoid* terdiri dari kurva pertumbuhan dan penyusutan yang merupakan kurva berbentuk huruf *S* dan digunakan menghubungkan kenaikan dan penurunan permukaan yang tidak linear. Definisi Kurva-*S* menggunakan 3 parameter , yaitu nilai keanggotaan nol ( $\alpha$ ), nilai keanggotaan satu ( $\gamma$ ), dan titik infleksi ( $\beta$ ) yaitu titik dengan domain yang memiliki derajat keanggotaan sebesar 0,5.

##### **1) Kurva-S untuk Pertumbuhan**

Kurva-*S* untuk pertumbuhan bergerak dari sisi kiri dengan nilai keanggotaan 0 ke sisi kanan dengan nilai keanggotaan 1. Seperti pada gambar 2.10 berikut:



**Gambar 2. 10** Kurva-S untuk Pertumbuhan

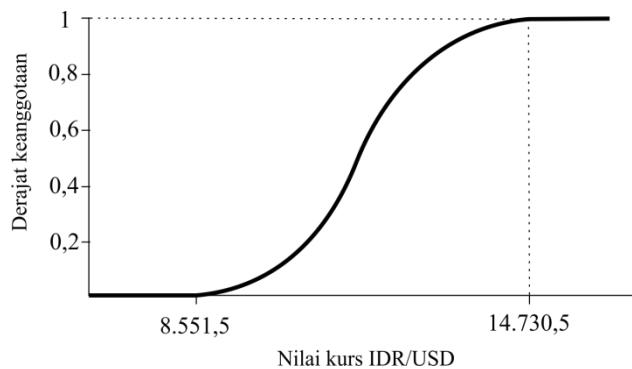
Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 0; & x \leq a \\ 2((x-a)/(y-a))^2; & a \leq x \leq \beta \\ 1 - 2((y-x)/(y-a))^2; & \beta \leq x \leq y \\ 1; & x \geq y \end{cases} \quad (2.11)$$

Contoh 2.12 Sebagai contoh himpunan *fuzzy* nilai kurs turun dengan semesta pembicaraan [5.462 17.820] dan domain [8.551,5 11.641 14.730,5] memiliki fungsi keanggotaan

$$\mu_{\text{nilai kurs turun}}(x) = \begin{cases} 0; & x \leq 8.551,5 \\ 2 \left( \frac{x - 8.551,5}{14.730,5 - 8.551,5} \right)^2 & 8.551,5 \leq x \leq 11.641 \\ 1 - 2 \left( \frac{14.730,5 - x}{14.730,5 - 8.551,5} \right)^2 & 11.641 \leq x \leq 14.730,5 \\ 1 & x \geq 14.730,5 \end{cases}$$

Grafik representasi dari fungsi keanggotaan tersebut dapat ditunjukkan pada Gambar 2.11 berikut:



**Gambar 2. 11** Himpunan *fuzzy* nilai kurs turun

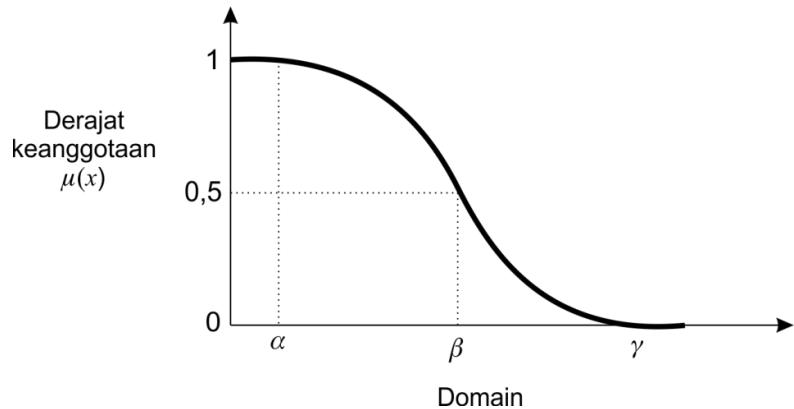
Berdasarkan fungsi keanggotaan pada gambar tersebut akan ditentukan derajat keanggotaan himpunan *fuzzy* dari nilai kurs IDR/USD sebesar 12.350 dengan perhitungan sebagai berikut:

$$\mu_{\text{nilai kurs turun}}[12.350] = 1 - 2 \left( \frac{14.730,5 - x}{14.730,5 - 8.551,5} \right)^2 = 0,7032$$

Dapat disimpulkan bahwa derajat keanggotaan nilai kurs IDR/USD sebesar 12.350 pada himpunan *fuzzy* nilai kurs turun adalah 0,7032. Sehingga nilai kurs IDR/USD sebesar 12.350 merupakan anggota himpunan *fuzzy* nilai kurs turun dengan nilai keanggotaan sebesar 0,7032.

## 2) Kurva-S untuk Penyusutan

Kurva-S untuk penyusutan bergerak dari sisi kanan dengan nilai keanggotaan 1 ke sisi kiri dengan nilai keanggotaan 0. Seperti pada gambar 2.12 berikut ini:



**Gambar 2. 12** Kurva – S untuk penyusutan

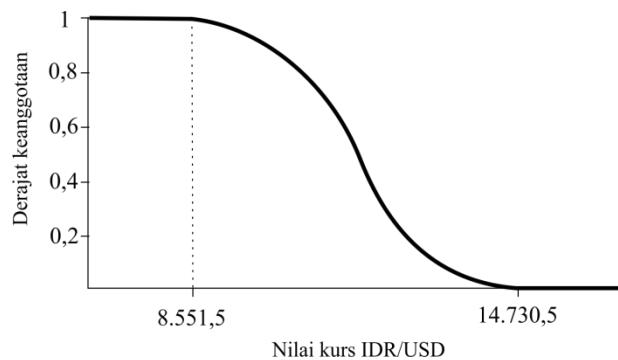
Fungsi keanggotaan:

$$\mu(x) = \begin{cases} 1; & x \leq a \\ 1 - 2((x - \alpha)/(\gamma - \alpha))^2; & a \leq x \leq \beta \\ 2((\gamma - x)/(\gamma - \alpha))^2; & \beta \leq x \leq \gamma \\ 0; & x \geq \gamma \end{cases} \quad (2.12)$$

Contoh 2.13 Sebagai contoh himpunan *fuzzy* nilai kurs naik dengan semesta pembicaraan [5.462 17.820] dan domain [8.551,5 11.641 14.730,5] memiliki fungsi keanggotaan:

$$\mu_{\text{nilai kurs naik}}(x) = \begin{cases} 1; & x \leq 8.551,5 \\ 1 - 2 \left( \frac{x - 8.551,5}{14.730,5 - 8.551,5} \right)^2 & 8.551,5 \leq x \leq 11.641 \\ 2 \left( \frac{14.730,5 - x}{14.730,5 - 8.551,5} \right)^2 & 11.641 \leq x \leq 14.730,5 \\ 0; & x \geq 14.730,5 \end{cases}$$

Grafik representasi dari fungsi keanggotaan tersebut dapat ditunjukkan pada Gambar 2.13 berikut:



**Gambar 2. 13** Himpunan *fuzzy* nilai kurs naik

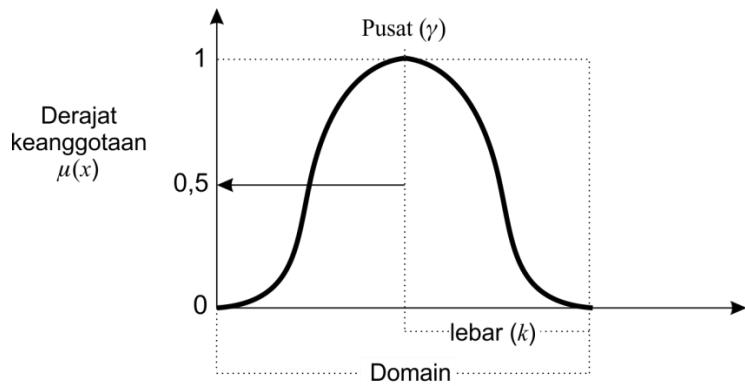
Berdasarkan fungsi keanggotaan pada gambar tersebut akan ditentukan derajat keanggotaan himpunan *fuzzy* dari nilai kurs IDR/USD sebesar 12.350 dengan perhitungan sebagai berikut:

$$\mu_{\text{nilai kurs naik}}[12.350] = 2 \left( \frac{14.730,5 - 12.350}{14.730,5 - 8.551,5} \right)^2 = 0,2968$$

Dapat disimpulkan bahwa derajat keanggotaan nilai kurs IDR/USD sebesar 12.350 pada himpunan *fuzzy* nilai kurs naik adalah 0,2968. Sehingga nilai kurs IDR/USD sebesar 12.350 merupakan anggota himpunan *fuzzy* nilai kurs naik dengan nilai keanggotaan sebesar 0,2968.

#### e. Representasi Kurva Gauss

Kurva gauss menggunakan parameter ( $\gamma$ ) untuk menunjukkan nilai domain pada pusat kurva dan ( $k$ ) sebagai lebar kurva seperti yang ditunjukkan pada gambar 2.14 berikut:



**Gambar 2. 14** Kurva gauss

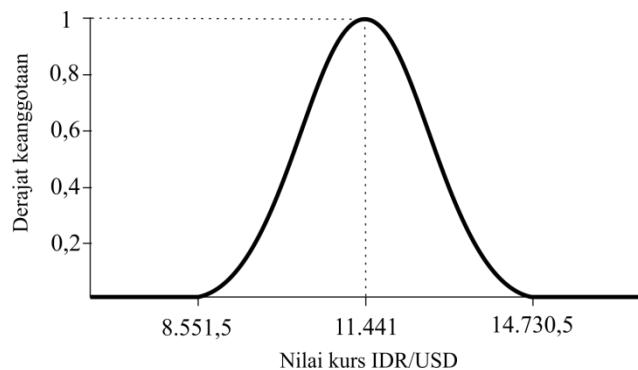
fungsi keanggotaan:

$$\mu(x) = e^{\frac{-(x-\gamma)^2}{2k^2}} \quad (2.13)$$

Contoh 2.14 Sebagai contoh himpunan kurs sedang dengan semesta pembicaraan  $[5.462 \quad 17.820]$  dan domain  $[8.551,5 \quad 14.730,5]$  memiliki fungsi keanggotaan:

$$\mu_{\text{kurs sedang}}(x) = e^{\frac{-(x-11.641)^2}{2(3.089,5)^2}}$$

Grafik representasi dari fungsi keanggotaan tersebut dapat ditunjukkan pada Gambar 2.15 berikut:



**Gambar 2. 15** Himpunan *fuzzy* kurs sedang

Berdasarkan fungsi keanggotaan pada gambar tersebut akan ditentukan derajat keanggotaan himpunan *fuzzy* dari nilai kurs IDR/USD sebesar 12.350 dengan perhitungan sebagai berikut:

$$\mu_{\text{kurs sedang}}(12.350) = e^{\frac{-(12.350-11.641)^2}{2(3.089,5)^2}} = 0,9740$$

Dapat disimpulkan bahwa derajat keanggotaan nilai kurs IDR/USD sebesar 12.350 pada himpunan *fuzzy* kurs sedang adalah 0,9740. Sehingga nilai kurs IDR/USD sebesar 12.350 merupakan anggota himpunan *fuzzy* kurs sedang dengan nilai keanggotaan sebesar 0,9740.

## 2. Operator *Fuzzy*

Seperti halnya himpunan konvensional, ada beberapa operasi yang didefinisikan secara khusus untuk mengkombinasikan dan memodifikasi himpunan *fuzzy*. Nilai keanggotaan sebagai hasil dari operasi 2 himpunan sering dikenal dengan nama *fire strength* atau  $\alpha$ -predikat. Ada 3 operator dasar yang diciptakan oleh Zadeh, (Kusumadewi, 2003:175) yaitu:

### a. Operator AND

Operator ini berhubungan dengan operasi interaksi pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator AND diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2.14)$$

Contoh 2.15 Misalkan derajat keanggotan nilai tukar IDR/USD 2.350 pada himpunan *fuzzy* kurs naik adalah 0,77 dan derajat keanggotan nilai tukar IDR/USD 2.350 pada himpunan *fuzzy* kurs turun adalah 0,11, maka

$$\begin{aligned}
\mu_{\text{kurs naik} \cap \text{kurs turun}}(2.350) &= \min(\mu_{\text{kurs naik}}(2.350), \mu_{\text{kurs turun}}(2.350)) \\
&= \min(0,77, 0,11) \\
&= 0,11
\end{aligned}$$

Dapat disimpulkan bahwa derajat keanggotaan nilai tukar IDR/USD 2350 pada irisan himpunan *fuzzy* kurs naik dan himpunan *fuzzy* kurs turun adalah sebesar 0,11.

### b. Operator OR

Operator ini berhubungan dengan operasi union pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator OR diperoleh dengan mengambil nilai keanggotaan terkecil antar elemen pada himpunan-himpunan yang bersangkutan.

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (2.15)$$

Contoh 2.16 Misalkan derajat keanggotan nilai tukar IDR/USD 2.350 pada himpunan *fuzzy* kurs naik adalah 0,77 dan derajat keanggotan nilai tukar IDR/USD 2.350 pada himpunan *fuzzy* kurs turun adalah 0,11, maka

$$\begin{aligned}
\mu_{\text{kurs naik} \cup \text{kurs turun}}(2.350) &= \max(\mu_{\text{kurs naik}}(2.350), \mu_{\text{kurs turun}}(2.350)) \\
&= \max(0,77, 0,11) \\
&= 0,77
\end{aligned}$$

Dapat disimpulkan bahwa derajat keanggotaan nilai tukar IDR/USD 2.350 pada gabungan himpunan *fuzzy* kurs naik dan himpunan *fuzzy* kurs turun adalah sebesar 0,77.

### c. Operator NOT

Operator ini berhubungan dengan operasi komplemen pada himpunan.  $\alpha$ -predikat sebagai hasil operasi dengan operator NOT diperoleh dengan mengurangkan nilai keanggotaan elemen pada himpunan yang bersangkutan dari 1.

$$\mu_{\tilde{A}'}(x) = 1 - \mu_{\tilde{A}}(x) \quad (2.16)$$

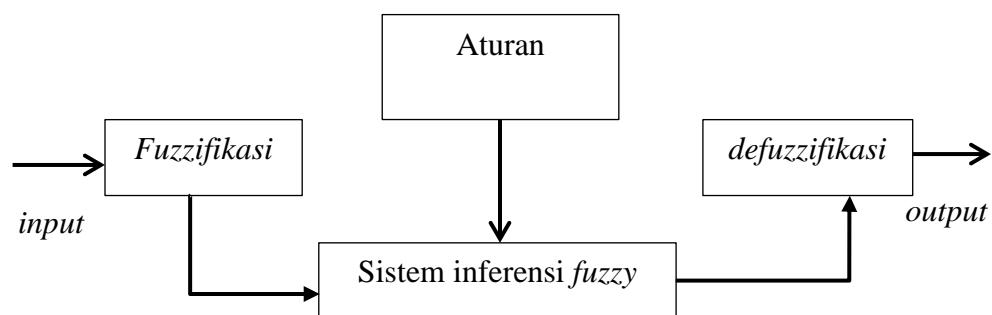
Contoh 2.17 Misalkan derajat keanggotaan nilai tukar IDR/USD 2.350 pada himpunan *fuzzy* kurs naik adalah 0,77, maka komplemen derajat keanggotaan nilai tukar IDR/USD pada himpunan *fuzzy* kurs naik

$$\begin{aligned} \mu_{\text{kurs naik}'}(2.350) &= 1 - \mu_{\text{kurs naik}}(2.350) \\ &= 1 - 0,77 \\ &= 0,23 \end{aligned}$$

Sehingga komplemen derajat keanggotaan nilai tukar EUR/USD pada himpunan *fuzzy* kurs naik adalah sebesar 0,23.

### D. Sistem *Fuzzy*

Menurut Wang (1997 : 7), sistem *fuzzy* terdiri dari 3 tahapan, yaitu *fuzzifikasi*, sistem inferensi *fuzzy*, dan *defuzzifikasi*. Sistem *fuzzy* tersebut dapat dibuat bagan seperti gambar 2.16 berikut:



**Gambar 2. 16** Pembentukan sistem *fuzzy*

### **1. *Fuzzifikasi***

*Fuzzifikasi* adalah proses untuk mengubah bilangan *crisp* menjadi derajat keanggotaan *fuzzy*. Pengubahan bilangan *crisp* ini berdasarkan fungsi keanggotaan yang digunakan pada himpunan *fuzzy*.

### **2. *Sistem inferensi fuzzy***

Sistem inferensi *fuzzy* adalah proses penalaran untuk mendapatkan *output* *fuzzy* dari *input fuzzy*.

### **3. *Defuzzifikasi***

*Defuzzifikasi* adalah proses pemetaan dari derajat keanggotaan himpunan *fuzzy* (yang merupakan *output* dari inferensi *fuzzy*) ke himpunan bilangan *crisp*. Fungsi yang digunakan untuk *defuzzifikasi* menyesuaikan dengan fungsi keanggotaan yang digunakan dalam proses *fuzzifikasi*.

## **E. *Neural Network***

*Neural Network* atau jaringan syaraf tiruan adalah suatu kumpulan hubungan dari elemen pemrosesan sederhana, unit atau *node* yang tidak lepas dari basis neuron biologis (Gurney, 2010:1). Menurut Siang (2009: 23), neuron adalah unit pemroses informasi yang menjadi dasar dalam pengoperasian *Neural Network*.

Struktur jaringan syaraf tiruan terdiri dari tiga lapisan sebagai berikut (Siang, 2009: 24):

### **1. Lapisan *input* (*input layer*)**

Lapisan *input* adalah tempat nilai-nilai variabel *input* yang akan diproses pada tahap selanjutnya.

### **2. Lapisan tersembunyi (*hidden layer*)**

Lapisan tersenbunyi terletak antara lapisan *input* dan *output*. Pada lapisan ini data masukan diproses untuk dijadikan keluaran.

### 3. Lapisan *output* (*output layer*)

Lapisan *output* merupakan tempat hasil dari pemrosesan selama jaringan dibentuk.

Konsep dasar *Neural Network* adalah menghubungkan antara variabel-variabel *input* dengan satu atau lebih variabel *output*. Lapisan *input* dan lapisan *output* dihubungkan melalui lapisan tersenbunyi dan ditransformasi menggunakan fungsi aktivasi. Namun tidak semua jaringan memiliki lapisan tersenbunyi. Ada tidaknya lapisan tersenbunyi ditentukan oleh arsitektur *Neural Network* yang digunakan.

Pada pembentukan model prediksi dengan *Neural Network* terdapat beberapa komponen penting yang harus diperhatikan, yaitu:

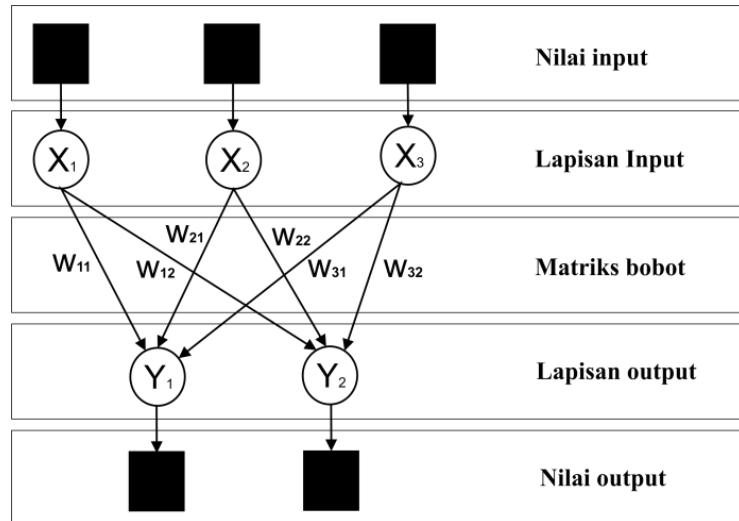
## 1. Arsitektur *Neural Network*

Arsitektur *Neural Network* menggambarkan bagaimana pola perjalanan sinyal/data dari lapisan *input* ke lapisan *output* pada *Neural Network*. Menurut Fausett (1994: 12), ada beberapa arsitektur jaringan yang sering dipakai dalam *NN* antara lain:

### a. Jaringan dengan Lapisan Tunggal (*Single Layer Net*)

Jaringan dengan lapisan tunggal hanya memiliki satu lapisan *input* dan *output* dengan bobot-bobot terhubung. Jaringan ini hanya menerima *input* kemudian

secara langsung akan mengolahnya menjadi *output* tanpa harus melalui lapisan tersembunyi (gambar 2.17) (Kusumadewi, 2003: 212).

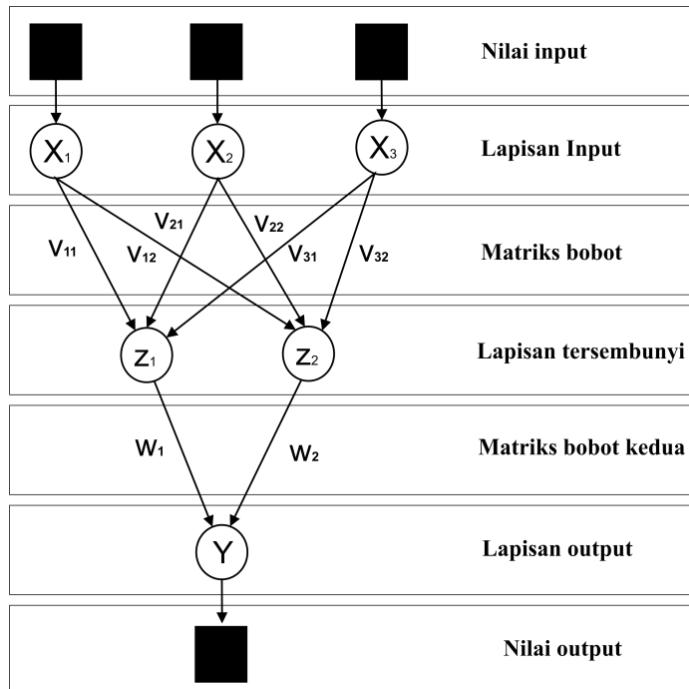


**Gambar 2. 17** Jaringan dengan lapisan tunggal

Pada gambar tersebut lapisan *input* memiliki 3 neuron yaitu  $X_1$ ,  $X_2$ , dan  $X_3$ . Sedangkan pada lapisan *output* memiliki 2 neuron yaitu  $Y_1$  dan  $Y_2$ . Setiap neuron pada lapisan *input* dan lapisan *output* saling terhubung oleh bobot yang bersesuaian ( $w$ ).

### b. Jaringan dengan Banyak Lapisan (*Multilayer Net*)

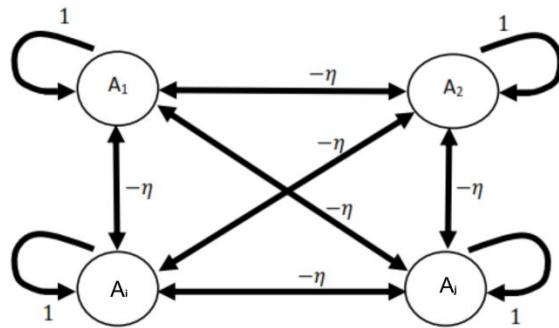
Menurut Kusumadewi (2003: 212), jaringan dengan banyak lapisan memiliki satu atau lebih lapisan yang terletak diantara lapisan *input* dan *output*. Jaringan dengan banyak lapisan dapat menyelesaikan masalah yang lebih kompleks dibandingkan dengan lapisan tunggal, meskipun kadangkala proses pelatihan lebih kompleks dan lama (Siang, 2009: 25). Contoh arsitektur jaringan dengan banyak lapisan yang memiliki satu lapisan tersembunyi dan 1 neuron pada lapisan *output* dapat dilihat pada gambar 2.18 berikut:



**Gambar 2. 18** Jaringan dengan banyak lapisan

**c. Jaringan dengan Lapisan Kompetitif (*Competitive Layer Net*)**

Jaringan syaraf dengan lapisan kompetitif memiliki bentuk yang berbeda dari jaringan syaraf tunggal dan banyak lapisan. Jaringan syaraf ini pada lapisan kompetitif neuron-neuron nya saling terhubung. Sinyal yang melalui lapisan neuron bergerak dalam dua arah dengan menghasilkan perulangan (*loop*) pada jaringan secara terus menerus sampai mencapai titik keseimbangan. Jaringan ini tetap pada titik keseimbangan sampai terjadi perubahan *input* dan mencari titik keseimbangan yang baru. *Recurrent network* dapat diperluas dengan menambah lapisan tersembunyi (Kusumadewi & Hartati, 2010: 75). Jaringan dengan lapisan kompetitif terlihat seperti gambar 2.19.



**Gambar 2. 19** Jaringan dengan lapisan kompetitif

## 2. Fungsi Aktivasi

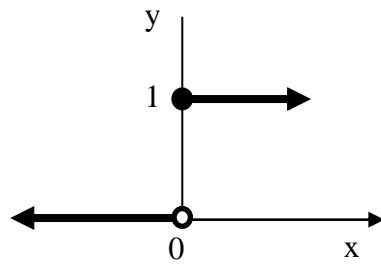
Fungsi aktivasi terdapat pada lapisan tersembunyi dan lapisan *output*. Fungsi aktivasi ini digunakan untuk menghitung sinyal/data keluaran suatu neuron pada lapisan tersembunyi dan lapisan *output* berdasarkan nilai-nilai variabel *input* pada lapisan *input* dan bobot yang menghubungkan setiap neuron pada lapisan yang berbeda.

Menurut Kusumadewi (2004: 51), terdapat beberapa fungsi aktivasi yang sering digunakan dalam jaringan syaraf tiruan, yaitu:

### a. Fungsi Undak Biner (*Hard Limit*)

Fungsi undak biner sering digunakan pada jaringan dengan lapisan tunggal untuk mengkonversi *input* dari suatu variabel yang bernilai *kontinu* ke suatu *output* biner (0 atau 1). Fungsi undak biner dirumuskan sebagai:

$$y = \begin{cases} 0, & \text{jika } x < 0 \\ 1, & \text{jika } x \geq 0 \end{cases} \quad (2.17)$$

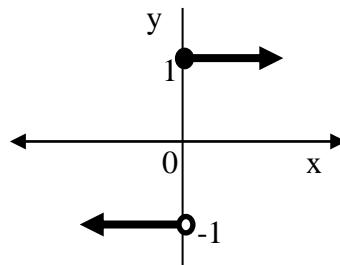


**Gambar 2. 20** Fungsi undak biner

**b. Fungsi Bipolar (*Symetric Hard Limit*)**

Fungsi bipolar hampir sama dengan fungsi undak biner, hanya saja *output* yang dihasilkan berupa 1 atau -1. Fungsi bipolar dirumuskan sebagai :

$$y = \begin{cases} -1, & \text{jika } x < 0 \\ 1, & \text{jika } x \geq 0 \end{cases} \quad (2.18)$$

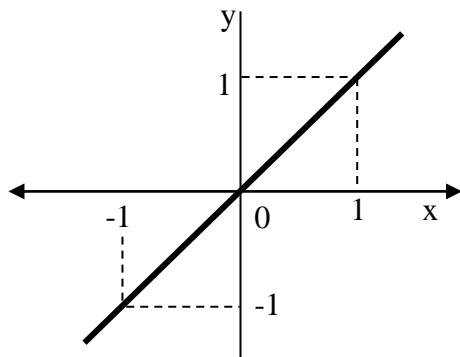


**Gambar 2. 21** Fungsi bipolar

**c. Fungsi Linear (Identitas)**

Fungsi linier memiliki nilai *output* yang sama dengan nilai *inputnya*. Fungsi linier dirumuskan sebagai :

$$y = x \quad (2.19)$$

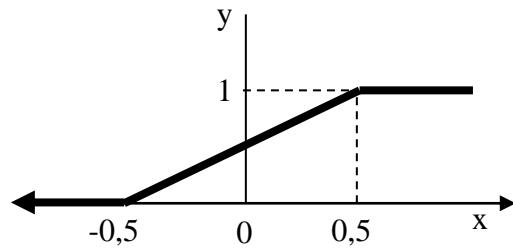


**Gambar 2. 22** Fungsi linier

**d. Fungsi *Saturating Linear***

Fungsi *saturating linear* dirumuskan sebagai berikut:

$$y = \begin{cases} 1; & \text{jika } x \geq 0,5 \\ x + 0,5; & \text{jika } -0,5 \leq x \leq 0,5 \\ 0; & \text{jika } x \leq -0,5 \end{cases} \quad (2.20)$$

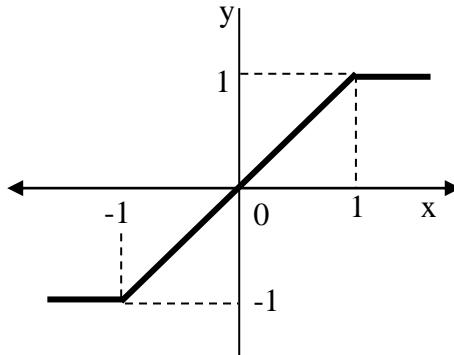


**Gambar 2. 23** Fungsi *saturating linear*

**e. Fungsi *Symmetric Saturating Linear***

Fungsi *symetric saturating linear* dirumuskan sebagai:

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



**Gambar 2. 24** Fungsi *symetric saturating linear*

**f. Fungsi *Sigmoid Biner***

Fungsi ini digunakan untuk jaringan syaraf yang menggunakan algoritma pembelajaran *backpropagation*. Fungsi *sigmoid biner* memiliki nilai pada interval 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.

Fungsi sigmoid biner dirumuskan sebagai:

$$y = f(x) = \frac{1}{1 + e^{-x}} \quad (2.22)$$

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

**g. Fungsi *Sigmoid Bipolar***

Fungsi *sigmoid bipolar* hampir sama dengan fungsi sigmoid biner, hanya saja *output* nya dari fungsi ini memiliki range antara 1 sampai -1. Fungsi sigmoid bipolar dirumuskan sebagai:

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

dengan:  $f'(x) = \frac{1}{2}[1 + f(x)][1 - f(x)]$

### 3. Algoritma Pembelajaran

Menurut Suyanto (2014: 178), belajar dalam *Neural Network* didefinisikan sebagai suatu proses dimana parameter-parameter bebas *Neural Network* diadaptasi melalui suatu proses perangsangan berkelanjutan oleh lingkungan dimana jaringan berada. Jenis belajar ditentukan oleh pola dimana pengubahan parameter dilakukan.

Proses pembelajaran merupakan bagian terpenting dari konsep jaringan syaraf tiruan. Proses pembelajaran bertujuan untuk mengubah bobot awal menjadi bobot akhir yang tepat dan sesuai dengan pola data yang diolah. Menurut Kusumadewi (2003: 220), pada saat proses pembelajaran nilai bobot mengalami perubahan. Nilai bobot akan bertambah, jika informasi yang diberikan oleh neuron yang bersangkutan tersampaikan, sebaliknya jika informasi tidak disampaikan oleh suatu neuron ke neuron yang lain, maka nilai bobot yang menghubungkan keduanya akan dikurangi.

Menurut Puspitaningrum (2006: 23), terdapat 3 metode pembelajaran, yaitu:

#### a. Pembelajaran Terawasi (*Supervised Learning*)

Pada metode ini, setiap pola yang diberikan kedalam jaringan syaraf tiruan telah diketahui *output*nya. Selisih antara pola *output* aktual (*output* yang dihasilkan) dengan pola *output* yang dikehendaki (*output* target) yang disebut *error* digunakan untuk mengoreksi bobot jaringan syaraf tiruan sehingga jaringan syaraf tiruan mampu menghasilkan *output* sedekat mungkin dengan pola target yang telah diketahui oleh jaringan syaraf tiruan. Contoh algoritma jaringan syaraf

tiruan yang menggunakan metode ini adalah: *Hebbian, Perceptron, ADALINE, Boltzman, Hopfield, dan Backpropagation.*

**b. Pembelajaran tak terawasi (*unsupervised learning*)**

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

**c. Pembelajaran hibrid (*hybrid learning*)**

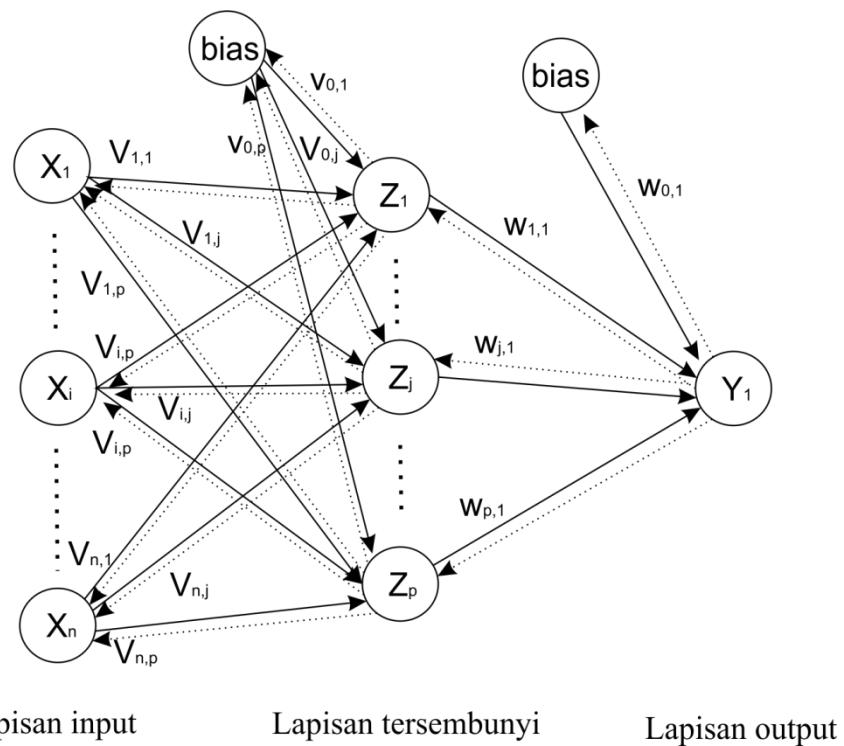
Merupakan kombinasi dari metode pembelajaran *supervised learning* dan *unsupervised learning*. Sebagian dari bobot-bobotnya ditentukan melalui pembelajaran terawasi dan sebagian lainnya melalui pembelajaran tak terawasi. Contoh algoritma jaringan syaraf tiruan yang menggunakan metode ini adalah algoritma *Radial Basis Function (RBF)*.

Pada skripsi ini arsitektur yang digunakan adalah arsitektur dengan banyak lapisan, sedangkan fungsi aktivasi yang digunakan adalah *sigmoid biner* dengan metode pembelajaran yang digunakan adalah metode pembelajaran terawasi dengan algoritma *backpropagation*.

## F. Algoritma *Backpropagation*

Menurut Kusumadewi (2003: 236), algoritma *backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward*). Untuk mendapatkan *error* ini, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu. Menurut Kusumadewi (2004: 93), pada saat perambatan maju, neuron-neuron diaktifkan dengan menggunakan fungsi aktivasi yang dapat dideferensiasikan, seperti fungsi: *sigmoid biner*, *sigmoid bipolar*, atau *linier*.

Arsitektur jaringan *backpropagation* dengan 1 neuron pada lapisan *output* ditunjukkan pada gambar 2.25 berikut:



**Gambar 2. 25** Jaringan *Backpropagation* dengan 1 *output*

Prosedur Algoritma pelatihan *Backpropagation* adalah sebagai berikut:

**Langkah 1** : Menentukan nilai dari maksimum *epoch* (*epoch* adalah pengulangan yang terjadi pada proses pelatihan di dalam *neural network* dalam memperbaiki *error*), maksimum *error*, dan laju pembelajaran ( $\alpha$ ). Laju pembelajaran merupakan sebuah parameter pembelajaran di dalam jaringan saraf tiruan *backpropagation* yang digunakan untuk mempercepat proses pelatihan dan bernilai antara 0 sampai 1.

**Langkah 2:** Inisialisasi bobot input ( $v_{i,p}$ ), bobot bias input( $v_{0,p}$ ), bobot lapisan tersembunyi ( $w_{j,k}$ ), dan bobot bias lapisan tersembunyi ( $w_{0,k}$ ) (ambil bobot awal dengan nilai random yang cukup kecil)

**Fase I : *Feed forward***

**Langkah 3:** Tiap unit masukan ( $x_i, i = 1, 2, \dots, n$ ) menerima sinyal dan meneruskannya ke unit selanjutnya yaitu lapisan tersembunyi

**Langkah 4:** Menghitung semua keluaran pada lapisan tersembunyi ( $Z_j, j = 1, 2, \dots, p$ ) dengan persamaan:

$$Z_{net_j} = v_{0j} + \sum_{i=1}^n x_i v_{ij} \quad (2.24)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output* nya:

$$Z_j = f(Z_{net_j}) \quad (2.25)$$

Dan kirimkan sinyal tersebut ke semua unit lapisan atasnya (unit-unit *output*).

Langkah ini dilakukan sebanyak jumlah lapisan tersembunyi.

**Langkah 5:** Menghitung semua keluaran jaringan di lapisan *output*, pada skripsi ini lapisan *outpt* hanya memiliki 1 neuron ( $Y_k, k = 1$ ) dengan persamaan:

$$Y_{net_k} = w_{ok} + \sum_{j=1}^p z_j w_{jk} \quad (2.26)$$

Gunakan fungsi aktivasi untuk menghitung sinyal *output* nya:

$$Y_k = f(Y_{net_k}) \quad (2.27)$$

**Langkah 6:** Menghitung nilai *error* menggunakan persamaan

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2$$

Dengan

- $y_t$  : data sebenarnya, pada algoritma *backpropagation* ini nilai  $y_t$  diisi dengan nilai target ( $T_k$ ) pada waktu  $t$
- $\hat{y}_t$  : hasil peramalan pada waktu  $t$ , pada algoritma backpropagation ini nilai  $\hat{y}_t$  diisi dengan nilai  $Y_k$  pada waktu  $t$
- $n$  : banyak pengamatan

Jika nilai *error* lebih besar dari nilai *error* maksimum yang ditentukan maka dilanjutkan ke fase *backpropagation*, jika nilai *error* lebih kecil dari nilai *error* maksimum yang ditentukan maka perhitungan dihentikan

### Fase II : *Backpropagation*

**Langkah 7:** Menghitung faktor  $\delta$  unit keluaran berdasarkan kesalahan di setiap unit keluaran ( $Y_k, k = 1$ )

$$\delta_k = (T_k - Y_k)f'(Y_{net_k}) \quad (2.28)$$

$\delta_k$  merupakan unit kesalahan yang akan dipakai dalam perubahan bobot layar di bawahnya (langkah 7)

Kemudian menghitung koreksi bobot ( $\Delta w_{jk}$ ) yang nantinya akan digunakan untuk memperbaiki  $w_{jk}$  dengan laju pembelajaran  $\alpha$

$$\Delta w_{jk} = \alpha \cdot \delta_k \cdot z_j \quad (2.29)$$

Kemudian hitung juga koreksi bias ( $\Delta w_{ok}$ ) yang nantinya akan digunakan untuk memperbaiki nilai  $w_{ok}$

$$\Delta w_{ok} = \alpha \delta_k \quad (2.30)$$

**Langkah 8:** Menghitung faktor  $\delta$  unit tersembunyi berdasarkan kesalahan di setiap unit tersembunyi ( $z_j, j = 1, 2, \dots, p$ )

$$\delta_{net_j} = \sum_{k=1}^m \delta_k \cdot w_{jk} \quad (2.31)$$

Faktor  $\delta$  unit tersembunyi :

$$\delta_j = \delta_{net_j} f'(Z_{net_j}) \quad (2.32)$$

Kemudian menghitung koreksi bobot ( $\Delta v_{ij}$ ) yang nantinya akan digunakan untuk memperbaiki nilai  $v_{ij}$

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i \quad (2.33)$$

Kemudian hitung juga koreksi bias ( $\Delta v_{oj}$ ) yang nantinya akan digunakan untuk memperbaiki nilai  $v_{oj}$

$$\Delta v_{oj} = \alpha \cdot \delta_j \quad (2.34)$$

### Fase III : Perubahan bobot

**Langkah 9:** Tiap-tiap unit *output* ( $Y_k, k = 1$ ) memperbaiki bobotnya ( $j = 1, 2, \dots, p$ )

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \quad (2.35)$$

$$w_{ok}(\text{baru}) = w_{ok}(\text{lama}) + \Delta w_{ok} \quad (2.36)$$

Tiap-tiap unit tersembunyi ( $Z_j, j = 1, 2, 3, \dots, p$ ) memperbaiki bobotnya ( $i = 1, 2, 3, \dots, n$ )

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij} \quad (2.37)$$

$$v_{0j}(\text{baru}) = v_{0j}(\text{lama}) + \Delta v_{0j} \quad (2.38)$$

**Langkah 10:** Bobot baru yang diperoleh dipergunakan untuk fase *Feed forward*.

Langkah 4 hingga 10 akan selalu dilakukan hingga nilai *error* kurang dari nilai *error* maksimum yang ditentuan, atau nilai *epoch* mencapai nilai *epoch* maksimal yang ditentukan.

Menurut Khairani (2014: 158), algoritma *backpropagation* memiliki kelemahan yang sangat menonjol, yaitu memerlukan waktu pembelajaran yang cukup lama. Untuk mempercepat pembelajaran pada *backpropagation* dapat digunakan perbaikan dengan teknik heuristik.

## 1. Perbaikan dengan Teknik Heuristik

Teknik ini digunakan untuk mempercepat dalam proses pelatihan *backpropagation*. Teknik heuristik ini merupakan pengembangan dari suatu analisis kinerja pada algoritma *steepest (gradient) descent standard* (Kusumadewi, 2004: 150). *Steepest (gradient) descent standard* adalah metode *gradient* sederhana yang menggunakan vektor gradien untuk menentukan arah pencarian pada setiap iterasi. Kemudian, dari arah tersebut akan ditentukan besar ukuran langkahnya. Metode *steepest descent* digunakan untuk mencari minimum

suatu fungsi, yakni dengan menggunakan nilai negatif dari gradien fungsi disuatu titik. Digunakan nilai negatif dari gradien karena gradien memberikan nilai kenaikan yang semakin besar. Dengan nilai negatif dari gradien maka akan diperoleh nilai penurunan yang semakin besar. Salah satu teknik heuristik yang sering digunakan adalah *gradient descent* dengan *momentum* dan *adaptive learning rate (traingdx)*.

**a. Gradient Descent dengan Momentum dan Adaptive Learning Rate (traingdx)**

Teknik ini memperbaiki bobot berdasarkan *gradient descent* dengan laju pembelajaran yang bersifat *adaptive* (*adaptive learning rate*) dan menggunakan momentum (*mc*). Menurut Khairani (2014: 161), *adaptive learning rate* merupakan pendekatan atau metode yang bertujuan untuk meningkatkan efektifitas dari parameter tingkat pembelajaran atau *learning rate*, dimana tingkat pembelajaran merupakan parameter yang berfungsi untuk meningkatkan kecepatan belajar dari jaringan *backpropagation*. Sedangkan momentum (*mc*) adalah suatu konstanta yang mempengaruhi perubahan bobot dan bernilai diantara 0 dan 1. Bila *mc* = 0 maka perubahan bobot akan sama dengan perubahan bobot sebelumnya. Pada teknik ini ada beberapa parameter yang harus diperhatikan, yaitu:

1) Maksimum *epoch*

Maksimum *epoch* adalah jumlah *epoch* maksimum yang boleh dilakukan selama proses pelatihan. Iterasi akan dihentikan apabila nilai *epoch* melebihi maksimum *epoch*. Perintah maksimum *epoch* dalam matlab ditulis dengan :

```
net.trainParam.epochs=MaxEpoch
```

Nilai *default* untuk maksimum *epoch* adalah 10.

2) Kinerja tujuan

Kinerja tujuan adalah target nilai fungsi kerja. Iterasi akan dihentikan apabila nilai fungsi kurang dari atau sama dengan kinerja tujuan. Perintah kinerja tujuan dalam matlab ditulis dengan :

```
net.trainParam.goal=TargetError
```

Nilai *default* untuk kinerja tujuan adalah 0.

3) *Learning rate* (laju pembelajaran)

Semakin besar *learning rate* akan berimplikasi pada semakin besar langkah pembelajaran. Nilai *learning rate* yang cukup kecil menjamin penurunan gradient terlaksana dengan baik, namun ini berakibat bertambahnya jumlah iterasi. Pada umumnya besarnya nilai laju pembelajaran tersebut dipilih mulai 0,001 sampai 1 selama proses pelatihan.

Perintah di MATLAB :

```
net.trainParam.Ir = LearningRate.
```

Nilai *default* untuk *learning rate* adalah 0,01.

4) Rasio kenaikan *learning rate*

Rasio ini berguna sebagai faktor pengali untuk menaikkan *learning rate* yang ada terlalu rendah untuk mencapai kekonvergenan. Perintah kenaikan *learning rate* dalam matlab ditulis dengan :

```
net.trainParam.lr_inc=IncLearningRate
```

Nilai *default* untuk kenaikan *learning rate* adalah 1.05.

### 5) Rasio untuk menurunkan *learning rate*

Rasio yang berguna sebagai faktor pengali untuk menurunkan *learning rate* apabila *learning rate* yang ada terlalu tinggi atau menuju ke ketidakstabilan.

Perintah di MATLAB :

```
net.trainParam.Ir_dec =DecLearningRate
```

Nilai *default* untuk rasio penurunan *learning rate* adalah 0,7.

### 6) Maksimum kenaikan kerja

Maksimum kenaikan kerja adalah nilai maksimum kenaikan *error* yang diijinkan, antara *error* saat ini dan *error* sebelumnya. Perintah di MATLAB:

```
net.trainParam.max_perf_inc =MaxPerfInc
```

Nilai *default* untuk maksimum kenaikan kinerja adalah 1,04.

### 7) Gradien minimum

Gradien minimum adalah akar dari jumlah kuadrat semua gradien (bobot input, bobot lapisan, bobot bias) terkecil yang diperbolehkan. Iterasi akan dihentikan apabila nilai akar jumlah kuadrat semua gradien ini kurang dari gradien minimum. Perintah di MATLAB:

```
Net.trainParam.min_grad = MinGradien
```

Nilai *default* untuk gradien minimum adalah  $10^{-10}$

### 8) Momentum

Momentum adalah perubahan bobot yang baru dengan dasar bobot sebelumnya. Besarnya momentum antara 0 sampai 1. Apabila besarnya momentum = 0 maka perubahan bobot hanya akan dipengaruhi oleh gradiennya. Sedangkan, apabila besarnya momentum = 1 maka perubahan bobot akan sama

dengan perubahan bobot sebelumnya. Perintah di MATLAB :

```
net.trainParam.mc = Momentum
```

Nilai *default* untuk momentum adalah 0,9.

- 9) Jumlah *epoch* yang akan ditunjukkan kemajuannya

Parameter ini menunjukkan berapa jumlah *epoch* yang berselang yang akan ditunjukkan kemajuannya. Perintah di MATLAB:

```
net.trainParam.show =EpochShow
```

Nilai *default* untuk jumlah *epoch* yang akan ditunjukkan adalah 25.

Langkah-langkah teknik *traingdx* ini adalah:

- 1) Menghitung bobot baru pada lapisan *output* dengan persamaan 2.35 dengan  $\Delta w_{jk}$  diperoleh dengan persamaan berikut

$$\Delta w_{jk} = \alpha \cdot \delta_k \cdot z_j \quad (2.39)$$

untuk *epoch* = 1, dan

$$\Delta w_{jk} = mc * \Delta w_{jk}(\text{epoch sebelumnya}) + \alpha \cdot \delta_k \cdot z_j \quad (2.40)$$

untuk *epoch* > 1

- 2) Menghitung bobot bias baru pada lapisan *output* dengan persamaan 2.36 dengan  $\Delta w_{ok}$  diperoleh dengan persamaan berikut

$$\Delta w_{ok} = \alpha \delta_k \quad (2.41)$$

untuk *epoch* = 1, dan

$$\Delta w_{ok} = mc * \Delta w_{ok}(\text{epoch sebelumnya}) + \alpha \delta_k \quad (2.42)$$

untuk *epoch* > 1

- 3) Menghitung bobot baru pada lapisan tersembunyi dengan persamaan 2.37 dengan  $\Delta v_{ij}$  diperoleh dengan persamaan berikut

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i \quad (2.43)$$

untuk  $epoch = 1$ , dan

$$\Delta v_{ij} = mc * \Delta v_{ij} (epoch sebelumnya) + \alpha \cdot \delta_j \cdot x_i \quad (2.44)$$

untuk  $epoch > 1$

- 4) Menghitung bobot bias baru pada lapisan tersembunyi dengan persamaan 2.38 dengan  $\Delta v_{0j}$  diperoleh dengan persamaan berikut

$$\Delta v_{0j} = \alpha \cdot \delta_j \quad (2.45)$$

untuk  $epoch = 1$ , dan

$$\Delta v_{0j} = mc * \Delta v_{0j} (epoch sebelumnya) + \alpha \cdot \delta_j \quad (2.46)$$

untuk  $epoch > 1$

- 5) Menghitung kinerja jaringan baru ( $perf2$ ) dengan menggunakan bobot-bobot tersebut.
- 6) Membandingkan kinerja jaringan baru ( $perf2$ ) dengan kinerja jaringan sebelumnya ( $perf$ ).
- 7) Jika  $perf2/perf > max\_perf\_inc$  maka laju pembelajaran ( $\alpha$ ) =  $\alpha * Ir\_dec$ .

$$\Delta w_{jk} = \alpha \cdot \delta_k \cdot z_j$$

$$\Delta w_{ok} = \alpha \delta_k$$

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i$$

$$\Delta v_{0j} = \alpha \cdot \delta_j$$

- 8) Jika  $perf2/perf < max\_perf\_inc$  maka laju pembelajaran ( $\alpha$ ) =  $\alpha * Ir\_inc$ .

$$\Delta w_{jk} = \alpha \cdot \delta_k \cdot z_j$$

$$\Delta w_{ok} = \alpha \delta_k$$

$$\Delta v_{ij} = \alpha \cdot \delta_j \cdot x_i$$

$$\Delta v_{0j} = \alpha \cdot \delta_j$$

- 9) Jika  $perf2 / perf = max\_perf\_inc$  maka bobot baru diterima sebagai bobot sekarang.

## G. Ukuran Akurasi Ketepatan Model Peramalan

Model peramalan yang baik adalah model yang memiliki nilai akurasi yang tinggi. Akurasi adalah tingkat kedekatan nilai peramalan terhadap nilai yang sebenarnya. Ada beberapa parameter yang dapat digunakan untuk mengukur tingkat akurasi model peramalan, yaitu:

### 1. **Mean Square Error (MSE)**

*Mean Square Error* (MSE) merupakan rata-rata dari kuadrat selisih data asli dengan data peramalan yang dirumuskan (Hanke & Wichern, 2005 : 80)

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \quad (2.47)$$

Dengan

$y_t$  : data sebenarnya pada waktu t

$\hat{y}_t$  : hasil peramalan pada waktu t

$n$  : banyak pengamatan

semakin kecil nilai MSE menunjukkan semakin kecilnya nilai selisih data peramalan dan data asli. Nilai selisih yang semakin kecil menunjukkan nilai peramalan yang semakin mendekati nilai asli sehingga hasil peramalan akan semakin akurat.

## 2. **Mean Absolute Percentage Error (MAPE)**

MAPE merupakan persentase nilai rata-rata *Absolute Error* dari kesalahan meramal tanpa menghiraukan tanda positif atau negatif yang dirumuskan (Hanke & Wichern, 2005 : 80)

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|Y_t - \hat{Y}_t|}{Y_t} \times 100\% \quad (2.48)$$

dengan

$Y_t$  = nilai pengamatan ke-t

$\hat{Y}_t$  = nilai peramalan pada waktu ke-t

$n$  = banyaknya pengamatan

semakin kecil nilai MAPE menunjukkan semakin kecilnya nilai selisih data peramalan dan data asli. Nilai selisih yang semakin kecil menunjukkan nilai peramalan yang semakin mendekati nilai asli sehingga hasil peramalan akan semakin akurat.

## H. Algoritma Genetika

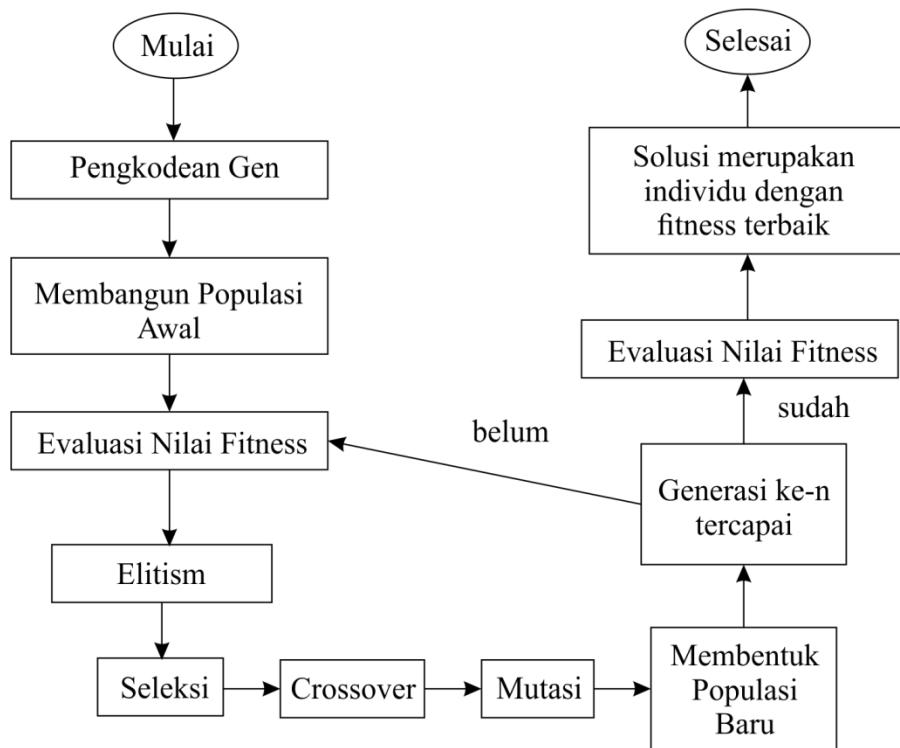
Menurut Wati (2011: 162), algoritma genetika merupakan sebuah metode untuk menyelesaikan masalah optimasi dengan meniru proses seleksi alam, proses yang menyebabkan evolusi biologis. Dalam proses evolusi hanya individu-individu yang kuat yang akan bertahan dalam proses seleksi alam. Konsep dasar algoritma genetika adalah untuk memperoleh individu terbaik sebagai solusi dari permasalahan melalui proses seleksi, *crossover*, dan mutasi pada algoritma genetika.

Menurut Suyanto (2005: 3) algoritma genetika sangat efisien dan berguna untuk menyelesaikan masalah dengan karakteristik sebagai berikut:

- a. ruang masalah sangat besar, kompleks, dan sulit dipahami,
- b. kurang atau bahkan tidak ada pengetahuan yang memadai untuk mempresentasikan masalah ke dalam ruang pencarian yang lebih sempit,
- c. tidak tersedianya analisis matematika yang memadai,
- d. ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi,
- e. solusi yang diharapkan tidak harus paling optimal, tetapi solusi yang cukup ‘bagus’ atau bisa diterima,
- f. terdapat batasan waktu, misalnya dalam *real systems* atau sistem waktu nyata.

## **1. Skema Algoritma Genetika**

Prinsip algoritma genetika dapat digambarkan sebagai berikut:



**Gambar 2. 26** Bagan algoritma genetika

Pada gambar 2.26 diatas Algoritma Genetika dimulai dengan menentukan nilai-nilai pada gen dimana kumpulan gen akan membentuk kromosom, himpunan kromosom akan memebentuk individu, dan himpunan individu membentuk populasi awal. Setiap individu pada populasi akan dievaluasi nilai *fitness*nya. Individu dengan *fitness* terbaik akan disimpan kemudian individu dalam populasi akan diseleksi, individu yang terseleksi kemudian *dicrossover* dan mutasi untuk memperoleh individu baru. Individu baru yang terbentuk digunakan untuk membentuk populasi baru pada generasi selanjutnya. Langkah-langkah tersebut akan diulang-ulang hingga diperoleh nilai solusi optimal atau setalah tercapai generasi *ke-n*.

## 2. Komponen-Komponen Algoritma Genetika

Komponen-komponen penting dalam algoritma genetika adalah sebagai berikut:

### a. Penyandian Gen (Pengkodean)

Menurut Fanggidae dan Lado (2015: 2), *Genotype* (Gen) adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam satu kesatuan gen yang dinamakan kromosom. Setiap gen memiliki suatu nilai yang disebut allele dimana nilai pada gen ini diperoleh berdasarkan teknik pengkodean yang digunakan. Menurut Syarif (2014 : 10), terdapat beberapa pengkodean yang sering digunakan yaitu:

- 1) Pengkodean biner, yaitu pengkodean dengan memberikan bilangan 1 atau 0 pada setiap gen.

Contoh 2.18 bilangan biner : 110100, 101010, 011010

- 2) Pengkodean permutasi, yaitu setiap gen memiliki nilai bilangan bulat yang berbeda. Pengkodean ini sering digunakan pada persoalan yang memperhatikan urutan seperti pada kasus *Travelling Salesman Problem*.

Contoh 2.19 pengkodean permutasi : 2, 4, 5, 7, 6

- 3) Pengkodean bilangan *riil*, yaitu suatu pengkodean dengan nilai setiap gen berisi nilai masing-masing variabel keputusan dari persoalan yang akan diselesaikan. Pengkodean bilangan *riil* memberikan penghematan memori dan memiliki waktu komputasi yang lebih cepat dibanding pengkodean biner (Wati, 2011: 180). Oleh karena itu pada penelitian ini menggunakan pengkodean bilangan *riil*.

Contoh 2.20 bilangan *riil*: 1,212, -3,12, 22,1

**b. Membangkitkan Populasi Awal (*Spanning*)**

Populasi adalah kumpulan dari beberapa individu, dimana setiap individu tersusun dari beberapa gen. Pada algoritma genetika individu menyatakan salah satu solusi yang mungkin dari permasalahan yang akan diselesaikan. Ada beberapa cara membangkitkan populasi awal, yaitu:

1) *Random Generator*

Pembangkitan populasi awal dengan random generator adalah dengan melibatkan pembangkit bilangan acak untuk mengisi nilai gen pada setiap individu pada populasi.

Contoh 2.21 Suatu populasi dengan 4 individu dimana setiap individu memiliki 6 buah gen dan pengkodean yang digunakan pada gen adalah bilangan *riil*. Dengan teknik random generator menggunakan MATLAB dengan perintah:

```
randn(4, 6)
```

diperoleh populasi awal sebagai berikut:

```
0.5377  0.3188  3.5784  0.7254 -0.1241  0.6715  
1.8339 -1.3077  2.7694 -0.0631  1.4897 -1.2075  
-2.2588 -0.4336 -1.3499  0.7147  1.4090  0.7172  
0.8622  0.3426  3.0349 -0.2050  1.4172  1.6302
```

2) Pendekatan Tertentu (Memasukkan Nilai Tertentu ke dalam Gen)

Cara ini dilakukan dengan memasukkan nilai tertentu ke dalam gen pada populasi awal yang dibentuk.

Contoh 2.22 Suatu populasi awal akan dibentuk dari bobot-bobot yang telah diperoleh dari beberapa percobaan jaringan syraf tiruan sebagai berikut:

Bobot percobaan 1

0.4889 -0.3034 0.8884 -0.8095 0.3252 -1.7115 0.3192 -0.0301

Bobot percobaan 2

1.0347 0.2939 -1.1471 -2.9443 -0.7549 -0.1022 0.3129 -0.1649

Bobot percobaan 3

0.7269 -0.7873 -1.0689 1.4384 1.3703 -0.2414 -0.8649 0.6277

maka bobot-bobot yang telah diperoleh tersebut langsung dimasukkan ke dalam gen pada populasi awal. Sehingga populasi awal yang terbentuk adalah:

0.4889 -0.3034 0.8884 -0.8095 0.3252 -1.7115 0.3192 -0.0301

1.0347 0.2939 -1.1471 -2.9443 -0.7549 -0.1022 0.3129 -0.1649

0.7269 -0.7873 -1.0689 1.4384 1.3703 -0.2414 -0.8649 0.6277

### c. Evaluasi Nilai *Fitness* (*Fitness Value*)

Nilai *fitness* digunakan untuk mengetahui baik tidaknya suatu individu. Tujuan dari algoritma genetika adalah mencari individu dengan nilai *fitness* yang tinggi. Nilai *fitness* diperoleh dengan suatu fungsi. Fungsi yang digunakan disesuaikan dengan kasus yang akan diselesaikan. Pada kasus optimasi dimana solusi yang dicari adalah nilai maksimum dari fungsi *y* maka nilai fitnes dapat

diperoleh dengan fungsi  $y$  sehingga  $f=y$  dimana  $f$  adalah fungsi nilai *fitness*. Sedangkan pada kasus optimasi dimana solusi yang dicari adalah nilai minimum dari fungsi  $y$  maka fungsi  $y$  tidak dapat langsung digunakan karena pada algoritma genetika individu dengan nilai *fitness* kecil akan hilang. Pada kasus ini nilai *fitness* dapat dicari dengan persamaan

$$f = \frac{1}{y} \quad (2.49)$$

Dengan persamaan tersebut, semakin kecil nilai  $y$  maka nilai *fitness* akan semakin tinggi. Namun dengan persamaan tersebut nilai *fitness* tidak akan terdefinisi jika nilai  $y=0$ . Oleh karena itu diberikan suatu nilai  $\alpha$  yang memiliki nilai cukup kecil dan bervariasi sesuai dengan masalah yang akan diselesaikan.

Sehingga persamaan nilai *fitness* menjadi

$$f = \frac{1}{y + \alpha} \quad (2.50)$$

#### **d. *Elitism***

*Elitism* bertujuan untuk menyimpan atau mempertahankan individu dengan nilai *fitness* terbaik. Nilai *fitness* terbaik perlu dipertahankan karena proses seleksi dilakukan secara acak sehingga tidak ada jaminan bahwa individu dengan *fitness* terbaik akan selalu terpilih. Jika nilai *fitness* terbaik terpilih, mungkin nilai *fitness* tersebut dapat rusak karena proses pindah silang dan mutasi.

### e. Seleksi (*Selection*)

Seleksi merupakan proses untuk memilih individu (induk) yang akan digunakan pada proses pindah silang dan mutasi. Terdapat beberapa teknik seleksi yang sering digunakan, yaitu:

#### 1) *Roulette wheel selection*

Teknik ini diilustrasikan sebagai teknik pemutaran cakram rolet, *probabilitas* keterpilihan individu berdasarkan rasio nilai fitnessnya dengan total nilai fitness dalam populasi (Arkeman dkk, 2012: 21) Langkah-langkah dalam teknik *roulette wheel selection*:

- a) Dihitung nilai *fitness* dari masing-masing individu pada populasi.
- b) Dihitung total nilai *fitness*.
- c) Dihitung *probabilitas* nilai *fitness* masing-masing individu
- d) Dihitung nilai *probabilitas* kumulatif
- e) Membangkitkan bilangan acak pada interval  $[0,1]$
- f) Dari bilangan random yang dihasilkan, ditentukan individu mana yang terpilih dalam proses seleksi.

Pada seleksi ini individu dengan nilai *fitness* tinggi memiliki kemungkinan besar untuk terpilih.

Contoh 2.23 Pada 6 individu akan dilakukan proses seleksi dengan *roulette wheel selection* dimana nilai *fitness* individu 1 hingga 6 secara berurutan sebagai berikut 0,3112; 0,5285; 0,1656; 0,6020; 0,2630; 0,6541. Diperoleh total nilai *fitness* 2,5244. Selanjutnya dihitung nilai *probabilitas* masing-masing individu dan *probabilitas* kumulatifnya seperti pada tabel 2.1 berikut:

**Tabel 2. 1** Probabilitas nilai *fitness*

individu	Nilai <i>fitness</i>	Probabilitas <i>fitness</i>	Probabilitas kumulatif <i>fitness</i>
1	0,3112	0,1233	0,1233
2	0,5285	0,2094	0,3327
3	0,1656	0,0656	0,3983
4	0,6020	0,2385	0,6368
5	0,2630	0,1042	0,741
6	0,6541	0,2591	1

Dengan bantuan MATLAB diperoleh nilai acak 0,6892 untuk menentukan individu terpilih. Nilai acak ini terdapat pada *probabilitas kumulatif fitness* individu 5 sehingga individu 5 yang terpilih dalam proses seleksi.

## 2) **Rank-Based Fitness Assignment (Seleksi Ranking)**

Pada seleksi *ranking* individu diurutkan menurut nilai objektifnya. Nilai *fitness* dari tiap-tiap individu hanya tergantung pada posisi individu tersebut dalam urutan, dan tidak dipengaruhi oleh nilai objektifnya. Cara kerja metode seleksi *ranking* adalah sebagai berikut:

- a) Dihitung nilai *fitness* dari masing-masing individu ( $f_i$ , dimana  $i$  adalah individu ke-1 s/d ke- $n$ ).
- b) Nilai *fitness* diurutkan dari nilai yang terkecil hingga paling besar.
- c) Setelah diurutkan, individu terburuk diberi nilai *fitness* baru sebesar 1, individu kedua terburuk diberi nilai 2, dan seterusnya. Individu terbaik diberi nilai *fitness* baru sebesar  $n$  dimana  $n$  adalah banyak individu dalam suatu populasi.
- d) Dihitung total *fitness* semua individu
- e) Dihitung *probabilitas* masing-masing individu
- f) Dihitung nilai *probabilitas kumulatif*

- g) Dibangkitkan bilangan random antara 0 sampai 1
- h) Dari bilangan random yang dihasilkan, ditentukan individu mana yang terpilih dalam proses seleksi

Pada seleksi *ranking* memberikan keragaman yang lebih banyak sehingga mengurangi kemungkinan solusi menuju lokal optimal.

Contoh 2.24 Dari 6 individu dengan nilai *fitness* individu 1 hingga 6 secara berurutan sebagai berikut 0,3112; 0,5285; 0,1656; 0,6020; 0,2630; 0,6541 akan dilakukan proses seleksi. Pertama individu diurutkan berdasarkan nilai *fitness* dari terkecil ke terbesar yang selanjutnya diberi nilai *fitness* baru dan dihitung nilai *probabilitas* masing masing individu sebagaimana tersaji pada tabel 2.2 berikut:

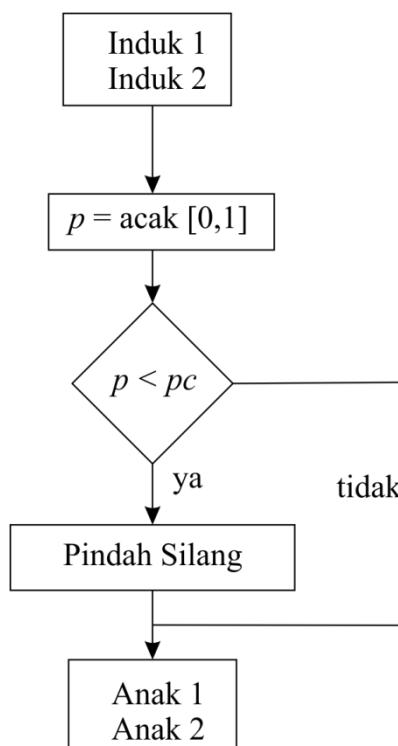
**Tabel 2. 2** Nilai *fitness* dan *probabilitas* nilai *fitness*

Nilai <i>fitness</i> lama yang diurutkan	individu	Nilai <i>fitness</i> baru	<i>Probabilitas</i> <i>fitness</i> baru individu	<i>Probabilitas</i> kumulatif
0,1656	3	1	0,047619048	0,047619048
0,263	5	2	0,095238095	0,142857143
0,3112	1	3	0,142857143	0,285714286
0,5285	2	4	0,19047619	0,476190476
0,602	4	5	0,238095238	0,714285714
0,6541	6	6	0,285714286	1

Kemudian dengan MATLAB diperoleh nilai acak 0,6892 untuk menentukan individu terpilih. Nilai acak ini terdapat pada *probabilitas* kumulatif *fitness* individu 4 sehingga individu 4 yang terpilih dalam proses seleksi.

#### f. Pindah Silang (Crossover)

Pindah silang (*crossover*) adalah operator utama atau operator primer dalam algoritma genetika yang bekerja pada sepasang kromosom induk untuk menghasilkan kromosom anak dengan cara menukar beberapa gen yang dimiliki masing-masing kromosom induk (Arkeman dkk, 2012: 22). Operasi ini tidak selalu dilakukan pada setiap induk yang ada. Setiap induk diberikan nilai *probabilitas* ( $p$ ) secara acak. Induk yang memiliki *probabilitas* kurang dari *probabilitas crossover* ( $pc$ ) yang telah ditentukan (biasanya antara 0,6 s/d 0,95) akan mengalami pindah silang. Jika pindah silang tidak dilakukan, maka nilai dari induk akan diturunkan kepada keturunannya. Secara skematis proses pindah silang dapat digambarkan sebagai berikut:



**Gambar 2.27** Sistematis proses pindah silang

Pada kasus dengan teknik pengkodean bilangan *rill* teknik pindah silang yang digunakan adalah pindah silang aritmatika (Syarif, 2014: 39). Pindah silang aritmatika dilakukan dengan menentukan bilang  $r$  secara acak antara 0 hingga 1. Selain itu ditentukan juga 2 bilangan acak  $k$  untuk menentukan posisi gen yang akan dipindah silangkan. Nilai  $k$  berada pada rentang  $[1, n]$  dimana  $n$  adalah banyaknya gen pada individu. Nilai pada anak diperoleh dengan persamaan sebagai berikut:

$$x'_1(k) = r \cdot x_1(k) + (1 - r) \cdot x_2(k) \quad (2.51)$$

$$x'_2(k) = r \cdot x_2(k) + (1 - r) \cdot x_1(k) \quad (2.52)$$

dengan

$x'_1$  = nilai gen pada anak 1

$x'_2$  = nilai gen pada anak 2

$r$  = nilai acak  $[0, 1]$

$k$  = posisi gen yang dilakukan pindah silang

$x_1$  = nilai gen pada induk 1 yang akan dipindah silangkan

$x_2$  = nilai gen pada induk 2 yang akan dipindah silangkan

Contoh 2.25 Dua buah induk dengan nilai gen sebagai berikut:

$$\begin{array}{ccccccc} 1,71 & 0,23 & -2,14 & -0,84 & 1,35 & -1,07 \\ 0,96 & 0,12 & 1,44 & -1,96 & -0,20 & -1,21 \end{array}$$

akan dilakukan pindah silang aritmatika dengan nilai  $r = 0,6$  nilai  $k = 2$  dan  $4$ .

Maka gen yang dipindah silangkan adalah gen ke-2, ke-3, dan ke-4. Dengan persamaan 2.51 dan 2.52 diperoleh nilai gen anak sebagai berikut:

$$x'_1(2) = 0,6(0,23) + (1 - 0,6)(0,12) = 0,19$$

$$x'_2(2) = 0,6(0,12) + (1 - 0,6)(0,23) = 0,17$$

$$x'_1(3) = 0,6(-2,14) + (1 - 0,6)(1,44) = -0,71$$

$$x'_2(3) = 0,6(1,44) + (1 - 0,6)(-2,14) = 0,01$$

$$x'_1(4) = 0,6(-0,84) + (1 - 0,6)(-1,96) = -1,29$$

$$x'_2(4) = 0,6(-1,96) + (1 - 0,6)(-0,84) = -1,51$$

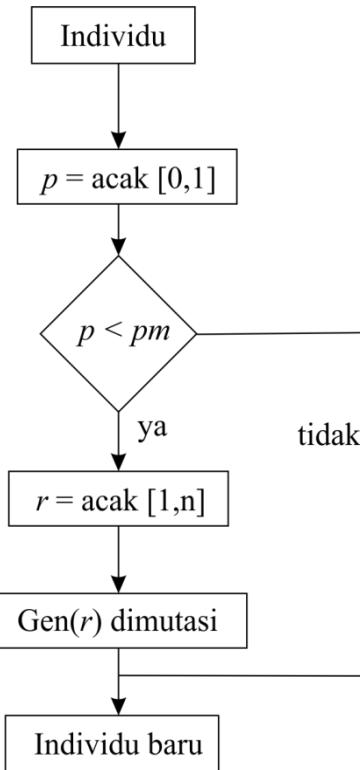
Sehingga anak yang terbentuk adalah

1,71	<b>0,19</b>	<b>-0,71</b>	<b>-1,29</b>	1,35	-1,07
0,96	<b>0,17</b>	<b>0,01</b>	<b>-1,51</b>	-0,20	-1,21

#### **g. Mutasi**

Mutasi merupakan operator sekunder atau operator pendukung dalam algoritma genetika yang berperan mengubah struktur kromosom secara spontan (Arkeman dkk, 2012: 23). Mutasi bertujuan untuk memperoleh individu-individu baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik.

Pada mutasi ditentukan suatu nilai *probabilitas* mutasi (*pm*) dimana *probabilitas* ini yang menentukan ada tidaknya mutasi pada gen. Selain nilai *pm* pada mutasi, dibangkitkan sebuah bilangan acak *p* untuk menentukan *probabilitas* mutasi individu. Jika nilai *p* < *pm* maka pada individu tersebut dilakukan mutasi. Untuk menentukan gen yang akan dimutasi maka dibangkitkan nilai acak *r* pada rentang [0, n] dimana n adalah banyak gen pada individu. Secara skematis proses mutasi dapat digambarkan sebagai berikut.



**Gambar 2. 28** Sistematika proses mutasi

Terdapat dua teknik mutasi pada bilangan *riil* yang sering digunakan, yaitu:

**1) *Random mutation***

Teknik ini mengganti gen yang termutasi dengan nilai acak.

$$a(k) = \text{bilangan acak} \quad (2.53)$$

dimana  $a(k)$  adalah gen ke- $k$ .

Contoh 2.26 Mutasi akan dilakukan pada individu berikut:

$$1,71 \quad 0,19 \quad -0,71 \quad -1,29 \quad 1,35 \quad -1,07$$

ditentukan nilai *probabilitas* mutasi sebesar 0,07. Dengan membangkitkan bilangan acak diperoleh *probabilitas* mutasi individu sebesar 0,021. Sehingga proses mutasi dapat dilakukan. Selanjutnya diperoleh nilai acak  $r$  sebesar 4 maka gen ke-4 yang mengalami mutasi, sehingga nilai pada gen ke-4 diganti dengan nilai acak sebagai berikut

1,71      0,19      -0,71      **1,37**      1,35      -1,07

## 2) *Shift mutation*

Teknik ini menggeser nilai gen termutasi sebesar  $\varepsilon$ , dimana  $\varepsilon$  adalah bilangan kecil yang ditentukan.

$$a(k) = a(k) + \varepsilon \quad (2.54)$$

Contoh 2.27 Mutasi akan dilakukan pada individu berikut:

1,71      0,19      -0,71      -1,29      1,35      -1,07

ditentukan nilai *probabilitas* mutasi sebesar 0,07 dan nilai  $\varepsilon$  sebesar 0,02. Dengan membangkitkan bilangan acak diperoleh *probabilitas* mutasi individu sebesar 0,001. Sehingga mutasi dapat dilakukan. Diperoleh bilangan acak  $r$  sebesar 3, maka gen ke-3 mengalami mutasi, sehingga diperoleh individu baru

1,71      0,19      **-0,69**      -1,29      1,35      -1,07

## h. Pembentukan Populasi Baru

Setelah diperoleh individu baru dari proses pindah silang dan mutasi. Maka individu-individu tersebut digunakan untuk membentuk populasi baru. Setelah populasi baru terbentuk, dilakukan pengulangan langkah-langkah evaluasi nilai *fitness*, proses seleksi, proses pindah silang, proses mutasi pada populasi baru untuk membentuk populasi baru selanjutnya. Proses ini berlangsung hingga diperoleh solusi yang optimal atau mencapai generasi yang telah ditentukan.