

BAB II

TINJAUAN PUSTAKA

2.1 Deret Waktu

Deret waktu (*Time Series*) adalah sekumpulan data pengamatan yang diukur selama kurun waktu tertentu, berdasarkan waktu dengan interval yang sama. Jika waktu dipandang bersifat diskrit (waktu dapat dimodelkan bersifat kontinu), frekuensi pengumpulan selalu sama. Dalam kasus diskrit frekuensi dapat berupa detik, menit, jam, hari, minggu, bulan bahkan tahun (Makridakis, dkk, 1999:3)

Dalam peramalan deret waktu harus memperhatikan tipe atau pola data yang ada. Pola pergerakan data atau variabel dapat diikuti dengan adanya data deret waktu, sehingga data deret waktu dapat digunakan sebagai dasar untuk pembuatan keputusan saat ini. Peramalan keadaan perdagangan dan ekonomi pada masa yang akan datang, perancangan kegiatan untuk masa depan (Anwary, 2011).

Pola data pada data deret waktu dikelompokkan menjadi empat jenis yaitu (Makridakis, 1992) :

1. Pola Horizontal (H) yaitu pola data yang terjadi jika tidak berfluktuasi di sekitar nilai rata-rata yang konstan.
2. Pola Musiman (S) yaitu pola data yang terjadi jika deret waktu dipengaruhi faktor musiman.
3. Pola Siklis (C) yaitu pola data yang terjadi jika data dipengaruhi fluktuasi ekonomi jangka panjang seperti yang berhubungan dengan siklus bisnis.
4. Pola Data Trend (T) yaitu pola data yang terjadi jika terjadi kenaikan ataupun penurunan sekuler jangka panjang pada data.

2.2 Stasioneritas

Dalam pembentukan model analisis deret waktu mempunyai ciri yaitu mengasumsikan bahwa data penelitian dalam keadaan stasioner. Dikatan stasioner jika deret waktu tersebut memiliki tiga kriteria yaitu :

1. Variansnya konstan
2. Nilai Tengah (rata-rata)
3. Pravalensi antar kedua deret waktu hanya tergantung lag antara dua periode tersebut

Berdasarkan rata-rata dan variansi terdapat dua jenis kstasioneran data yaitu data stasioner pada rata-rata dan data stasioner pada variansi. Jika deret waktu bersifat kuat, waktu pengamatan tidak berpengaruh terhadap rata-rata μ , variansi σ^2 dan kovariansi γ_k . Hal tersebut menandakan Z_t akan berfrukturasi disekitar rata-rata sedangkan variansinya tetap dan dapat dikatakan jika deret Z_t stasioner dalam rata-rata dan vaiansi.

2.2.1 Differencing

Jika tidak stasioner pada mean maka perlu diatasi dengan cara didefensiasi (*differencing*) terhadap deret data asli. *Differencing* yaitu proses mencari perbedaan antara data satu periode dengan periode sebelumnya secara beraturan. Proses *differencing* yang sesuai harus memiliki mean yang mendekati nol.

Dalam melakukan *differencing* pada sebuah data runtun waktu, diferensiasi sebuah variabel baru w_t yang merupakan deretan besarnya perubahan pada runtun waktu z_t yaitu

$$w_t = Z_t - Z_{t-1}, \quad t = 2, 3, \dots, n \quad (2.1)$$

Runtun waktu w_t disebut *differencing* pertama dari Z_t . Jika *differencing* pertama tidak menghasilkan runtun waktu yang memiliki mean konstan, maka dideferensikan kembali w_t sebagai *differencing* pertama dari *differencing* pertama.

$$w_t = (Z_t - Z_{t-1}) - (Z_{t-1} - Z_t), \quad t = 3, 4, \dots, n \quad (2.2)$$

Sekarang waktu w_t disebut *differencing* kedua dari Z_t karena merupakan hasil *differencing* kedua dari Z_t . namun pada umumnya *differencing* pertama sudah cukup untuk memperoleh mean yang stasioner.

2.2.2 Transformasi Data

Untuk ketidakstasioneran dalam hal varian dapat dihilangkan dengan cara melakukan transformasi data untuk menstabilkan variansi. Mentransformasikan data dapat digunakan transformasi kuasa (*The Power of Transformation*) dengan λ disebut parameter transformasi. Beberapa nilai λ yang umum digunakan adalah sebagai berikut.

Tabel 2.1 Nilai Lamda dan Transformasi

Nilai λ	Transformasi
-1	$\frac{1}{X_t}$
-0.5	$\frac{1}{\sqrt{X_t}}$
0	$\text{Ln } X_t$
0.5	$\sqrt{X_t}$
1	X_t (tidakada transformasi)

2.3 ACF dan PACF

Dalam menentukan model pada metode data runtun waktu perlu adanya alat untuk mengidentifikasi data yang akan diramalkan yaitu dengan menggunakan Fungsi Autokorelasi / *Autocorreation Function* (ACF) dan Fungsi Autokorelasi Parsial / *Partial Autocorrelation Function* (PACF).

2.3.1 Fungsi Autokorelasi (ACF)

Autokorelasi adalah korelasi antar deret pengamatan suatu deret waktu, sedangkan autokorelasi fungsi yaitu (ACF) adalah plot autokorelasi-korelasi. Pada proses stasioner suatu data time series (Z_t) diperoleh $E(Z_t) = \mu$ dan variansi $\text{Var}(Z_t) = E(Z_t - \mu)^2 = \sigma^2$, yang konstan dan kovarian $\text{Cov}(Z_t, Z_s)$, dimana fungsinya hanya pada perbedaan waktu $|t-s|$.

Diagram ACF dapat digunakan sebagai alat ukur mengidentifikasi kestasioneran data. Jika diagram ACF cenderung turun lambat atau turun secara linier, maka dapat disimpulkan data belum stasioner dalam rata-rata.

Dalam penentuan pola ACF di bagi menjadi dua yaitu Menurun secara eksponensial menuju nol atau yang lebih dikenal dengan sebutan *Dies Down* dan untuk pola kedua yaitu signifikan pada semua lag atau lebih dikenal dengan sebutan *Cut Off Lag*.

2.3.2 Fungsi Parial Autokorelasi (PACF)

Autokorelasi parsial (*Partial Autocorrelation*) adalah korelasi antar deret pengamatan suatu deret waktu dimana digunakan untuk mengukur hubungan keeratan antar pengamatan suatu deret waktu. Tingkat keeratan yang diukur yaitu antara Z_t dan Z_{t+k} , apabila pengaruh dari time lag 1,2,3,..., dan seterusnya sampai k-1 dianggap terpisah.

Dalam penentuan pola PACF di bagi menjadi dua yaitu menurun secara eksponensial menuju nol atau yang lebih dikenal dengan sebutan *Dies Down* dan untuk pola kedua yaitu signifikan pada semua lag atau lebih dikenal dengan sebutan *Cut Off Lag*.

Dalam penentuan model menggunakan pola ACF dn PACF dapat dilihat pada tabel model seperti dibawah ini.

Tabel 2.2 Model ACF dan PACF

Model	Pola ACF	Pola PACF
AR (p)	<i>Dies Down</i>	<i>Cut Off Lag</i>
MA (q)	<i>Cut Off Lag</i>	<i>Cut Off Lag</i>
ARMA (p,q)	<i>Dies Down</i>	<i>Dies Down</i>

2.4 Asumsi *White Noise*

Proses *White Noise* adalah deret variabel acak yang independen, identik dan terdistribusi. Suatu proses $\{\alpha_t\}$ dapat dikatakan *white noise* jika data terdiri dari variabel acak yang tidak berkorelasi dan berdistribusi normal dengan rata-rata konstan $E(\alpha_t) = 0$, variansi konstan $\text{Var}(t) = \sigma_\alpha^2$ dan $\gamma_k = \text{Cov}(\alpha_t, \alpha_{t+k}) = 0$ untuk $k \neq 0$.

Dengan demikian proses *white noise* stasioner dengan fungsi autokovariansi

$$\gamma \begin{cases} \sigma_\alpha^2, & \text{jika } k = 0 \\ 0, & \text{jika } k \neq 0 \end{cases} \quad (2.3)$$

Fungsi autokorelasi

$$\rho_k \begin{cases} 1, & \text{jika } k = 0 \\ 0, & \text{jika } k \neq 0 \end{cases} \quad (2.4)$$

Fungsi autokorelasi parsial

$$\Phi_{kk} \begin{cases} 1, & \text{jika } k = 0 \\ 0, & \text{jika } k \neq 0 \end{cases} \quad (2.5)$$

Proses *white noise* dapat diprediksi menggunakan uji autokorelasi residual pada analisis *error*-nya. Uji korelasi residual digunakan untuk mendeteksi

ada tidaknya korelasi residual antar *lag* . langkah-langkah pengujian korelasi residual yaitu :

$H_0 = \rho_1 = \rho_2 = \rho_3 = \dots = 0$ (residual tidak terdapat korelasi)

$H_1: \exists \rho_k \neq 0, k = 1, 2, \dots, K$ (residual terdapat korelasi)

Taraf signifikansi $\alpha = 5\%$

Statistik uji *Ljung Box-Pierce*. Rumus uji *Ljung Box-Pierce* :

$$QL_{LB} = T (T + 2) \sum_{k=1}^k \frac{\hat{\rho}_k^2}{T-K} \quad (2.6)$$

Dengan :

T = banyaknya data

K = banyaknya lag yang diuji

$\hat{\rho}_k$ = dugaan autokorelasi residual periode k

Kriteria keputusan yaitu tolak H_0 jika Q hitung $> \chi_{\alpha, d, f}^2$ tabel, dengan derajat kebebasan K dikurangi banyaknya parameter pada model.

2.5 Model ARIMA Box-Jenkins

Model ARIMA Box-Jenkins dibagi kedalam 3 kelompok, yaitu : *Autoregressive* (AR) , *Moving Average* (MA) , dan model campuran ARIMA (*Autoregressive Moving Average*) yang merupakan campuran dari kedua model.

2.5.1 Model *Autoregressive* (AR)

Model AR (p) adalah model dimana Z_t merupakan fungsi dari data dimasa yag lalu yakni t-1, t-2, ..., t-p . Rumus umum dari model AR adalah :

$$Z_t = \phi_1 Z_{t-1} + \dots + \phi_p Z_{t-p} + a_t \quad (2.7)$$

$$Z_t - \phi_1 Z_{t-1} - \dots - \phi_p Z_{t-p} = a_t \quad (2.8)$$

Maka dapat ditulis dalam bentuk berikut

$$(1 - \phi_1 B - \dots - \phi_p B^p) = a_t \quad (2.9)$$

$$\text{Dimana } BZ_t = Z_{t-1} \quad (2.10)$$

Atau

$$\phi_p(B)Z_t = a_t \quad (2.11)$$

$$\text{Dengan } \phi_p(B) = (1 - \phi_1 B - \dots - \phi_p B^p) \quad (2.12)$$

Model tersebut menyatakan hubungan antara peubah tak bebas Z terhadap himpunan peubah bebas Z_{t-p} ditambah sebuah suku yang menyatakan *error* a , model ini sering kali dinyatakan sebagai model regresi dan dikatakan Z diregresikan terhadap Z_{t-p} .

2.5.2 Model *Moving Average* (MA)

Model MA (q) adalah salah satu cara dalam memuluskan data deret waktu dimana hal tersebut bertujuan untuk menghilangkan atau meminimalisir dampak dari faktor sikliks, musiman serta random hingga akhirnya didapatkan sebuah trend. Jika data yang digunakan adanya faktor musiman membuat data berfrukturasi (naik-turun). Dengan dilakukannya proses *Moving Average* maka diharapkan data menjadi lebih smooth dan frakturasi data lebih kecil.

Model MA (q) adalah model untuk memprediksi Z_t sebagai fungsi dari kesalahan prediksi dimasa lalu (*past forecast error*) dalam

memprediksi Z_t . Secara umum model MA (q) adalah sebagai berikut:

$$Z_t = a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (2.13)$$

$$Z_t = (1 - \theta_1 B^p - \dots - \theta_q a_{t-q}) a_t \quad (2.14)$$

$$\text{Dimana } B a_t = a_{t-1} \quad (2.15)$$

Atau

$$Z_t = \theta_q(B) a_t \quad (2.16)$$

$$\text{Dengan } \theta_q(B) = (1 - \theta_1 B - \dots - \theta_q B^q) \quad (2.17)$$

2.5.3 Model Autoregressive Moving Average (ARMA)

Model ARMA (p, q) adalah kombinasi dari model AR (p) dan MA

(q) yaitu :

$$Z_t = \phi_1 Z_{t-1} + \dots + \phi_p Z_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (2.18)$$

$$Z_t - \phi_1 Z_{t-1} - \dots - \phi_p Z_{t-p} = a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (2.19)$$

Sehingga dapat ditulis sebagai berikut :

$$(1 - \phi_1 B - \dots - \phi_p B^p) Z_t = (1 - \theta_1 B - \dots - \theta_q B^q) a_t \quad (2.20)$$

Atau

$$\phi_p(B) Z_t = \theta_q(B) a_t \quad (2.21)$$

2.5.4 Model Autoregressive Integrated Moving Average (ARIMA)

Model ARIMA adalah model yang tidak mengasumsikan pada pola tertentu pada data yang akan diramalkan dan modelnya tidak memperhatikan variabel bebas yang ada. Model ARIMA merupakan

gabungan dari *Autoregressive* (AR) dan *Moving Average* (MA) dimana model ini dapat mewakili deret waktu yang stasioner dan non stasioner.

Dalam deret ARIMA adanya konstanta tambahan yaitu d yang dikenal dengan konstanta untuk diferensiasi membuat data stasioner, dan untuk konstanta lainnya yaitu p dan q . nilai konstanta p dan q didapatkan dari estimasi parameter ACF dan PACF . Sedangkan untuk nilai d dilakukan dengan *trail error* terhadap nilai p dan q yang sudah didapatkan. Secara umum model ARIMA dirumuskan dengan notasi berikut :

ARIMA (p,d,q)

Model ARIMA dinotasikan dengan ARIMA (p,d,q) dimana apabila $d=0$ dan $q=0$, maka model *Autoregressive* dinotasikan sebagai AR (p). apabila $p=0$ dan $d=0$, maka model *Moving Average* dinotasikan sebagai MA (q).

Model ARIMA dapat dilakukan pada data stasioner maupun data yang tidak stasioner. Karena pada data deret waktu banyak yang tidak stasioner maka perlu dilakukannya proses *differencing* sebanyak d kali sehingga nantinya data tersebut menjadi stasioner. Secara umum rumus dari model ARIMA (p,d,q) adalah sebagai berikut :

$$\phi_p(B)(1-B)^d Z_t = \theta_q(B) a_t \quad (2.22)$$

dengan

$$\phi_p(B) = (1 - \phi_1 B - \dots - \phi_p B^p), AR(p) \quad (2.23)$$

$$\theta_q(B) = (1 - \theta_1 B - \dots - \theta_q B^q), MA(q) \quad (2.24)$$

Dimana :

$(1 - B)^d$ = differencing orde d

a_t = nilai residual pada t

2.6 Model Seasonal Autoregressive Integrated Moving Average (SARIMA)

Model SARIMA pada dasarnya Model ARIMA akan tetapi datanya merupakan data musiman atau seasonal. Secara umum model SARIMA dinotasikan sebagai berikut :

$ARIMA(p, d, q)(P, D, Q)^S$

Dimana :

(p, d, q) = bagian tidak musiman dari model

(P, D, Q) = bagian musiman dari model

S = jumlah periode musiman

Model SARIMA secara umum adalah sebagai berikut :

$$\phi_p(B)\Phi_P(B^S)(1 - B)^d(1 - B^S)^D Z_t = \theta_q(B)\Theta_Q(B^S) a_t \quad (2.25)$$

Dimana

Z_t : data deret waktu dengan rata-rata μ

$\phi_p(B)$: Persamaan Polinomial AR(p)

$\theta_q(B)$: Persamaan Polinomial MA(q)

$\phi_p(B^S)$: Persamaan Polinomial Musiman AR(P)

$\theta_q(B^S)$: Persamaan Polinomial Musiman MA(q)

$(1 - B)^d$: Operator pembeda non musiman

$(1 - B^S)^d$: Operator pembeda musiman dengan periode S

ε_t : Residu deret waktu (dimusimkan *white noise*)

2.6.1 Model ARIMA $(P, D, Q)^S$ Musiman Non-Multipikatif Stasioner

a. Model ARIMA $(P, 0, 0)^S$

Suatu proses Z_t mengikuti model ARIMA $(P, 0, 0)^S$ apabila

$$Z_t = \Phi_1 Z_{t-s} + a_t \quad (2.26)$$

b. Model ARIMA $(0, 0, Q)^S$

Suatu proses Z_t mengikuti model ARIMA $(0, 0, Q)^S$ apabila

$$Z_t = a_t - \Theta_1 Z_{t-s} \quad (2.27)$$

c. Model ARIMA $(P, 0, Q)^S$

Suatu proses Z_t mengikuti model ARIMA $(P, 0, Q)^S$ apabila

$$Z_t = \Phi_1 Z_{t-s} + a_t - \Theta_1 Z_{t-s} \quad (2.28)$$

2.6.2 Model ARIMA $(P, D, Q)^S$ Musiman Non-Multipikatif Non-Stasioner

Suatu proses Z_t mengikuti model ARIMA yang non-stasioner dalam rata-rata musiman jika orde D berbeda dengan nol. Model umum untuk ARIMA $(P, D, Q)^S$ adalah

$$\Phi_P(B^S)(1 - B)^D Z_t = \Theta_Q(B^S) a_t \quad (2.29)$$

2.6.3 Model ARIMA $(p, d, q) (P, D, Q)^S$ Musiman Multipikatif Stasioner

a. Model ARIMA $(p, 0, 0)(P, 0, 0)^S$

Suatu proses Z_t mengikuti model ARIMA $(p, 0, 0)(P, 0, 0)^S$ apabila

$$(1 - \phi_1 B)(1 - \Phi_1 B^s) Z_t = a_t \quad (2.30)$$

$$Z_t = \phi_1 Z_{t-1} + \Phi_1 Z_{t-s} - \phi_1 \Phi_1 Z_{t-(s+1)} + a_t \quad (2.31)$$

b. Model ARIMA (0, 0, q)(0, 0, Q)^S

Suatu proses Z_t mengikuti model ARIMA $(p, 0, 0)(P, 0, 0)^S$ apabila

$$(1 - \theta_1 B) Z_t = (1 - \Theta_1 B^s) a_t \quad (2.32)$$

$$Z_t = a_t - \theta_1 a_{t-1} + \Theta_1 a_{t-s} - \theta_1 \Theta_1 a_{t-(s+1)} \quad (2.33)$$

c. Model ARIMA (0, 0, q)(P, 0, 0)^S

Suatu proses Z_t mengikuti model ARIMA $(0, 0, q)(P, 0, 0)^S$ apabila

$$(1 - \Phi_1 B^s) Z_t = (1 - \theta_1 B) a_t \quad (2.34)$$

$$Z_t = \Phi_1 Z_{t-s} + a_t - a_{t-1} - \theta_1 a_{t-1} \quad (2.35)$$

d. Model ARIMA (p, 0, 0)(0, 0, Q)^S

Suatu proses Z_t mengikuti model ARIMA $(p, 0, 0)(0, 0, Q)^S$ apabila

$$(1 - \phi_1 B) Z_t = (1 - \Theta_1 B^s) a_t \quad (2.35)$$

$$Z_t = \phi_1 Z_{t-1} + a_t + \Theta_1 a_{t-s} \quad (2.36)$$

2.6.4 Model ARIMA (p,d,q) (P, D, Q)^S Musiman Non-Stasioner dalam

Rata-rata Non-Musiman

Secara umum bentuk umum model ARIMA Box-Jenkins musiman atau ARIMA $(p,d,q) (P,D,Q)^S$ apabila orde d bukanlah nol maka hal tersebut menunjukkan bahwa proses adalah non-stasioner dalam rata-rata non-musiman. Sehingga secara umum model yang termasuk dalam kelompok model ARIMA dengan orde $(p,d,q) (P, 0, Q)^S$.

2.6.5 Model ARIMA $(p,d,q) (P,D,Q)^S$ Musiman Non-Stasioner dalam Rata-rata Musiman

Secara umum bentuk umum model ARIMA Box-Jenkins musiman atau ARIMA $(p,d,q) (P,D,Q)^S$ apabila orde D bukanlah nol maka hal tersebut menunjukkan bahwa proses adalah non-stasioner dalam rata-rata musiman. Sehingga secara umum model yang termasuk dalam kelompok model ARIMA dengan orde $(p,0,q) (P,D,Q)^S$

2.6.6 Model ARIMA $(p,d,q) (P,D,Q)^S$ Musiman Non-Stasioner dalam Rata-rata Non-Musiman dan Rata-rata Musiman

Secara umum bentuk umum model ARIMA Box-Jenkins musiman yang tidak stasioner dalam rata-rata non-musiman dan rata-rata musiman adalah model ARIMA dengan orde $(p,d,q) (P,D,Q)^S$.

Berdasarkan pendekatan Box-Jenkins dalam melakukan analisis deret waktu dibagi menjadi 4 proses, yaitu :

1. Identifikasi Model

Sebelum melakukan analisis deret waktu, tahapan yang pertama kali dilakukan yaitu identifikasi karakteristik data. Apakah data yang kita gunakan stasioner, musiman, non-musiman atau lainnya. Selanjutnya menentukan nilai p , d , dan q dengan bantuan pola ACF dan PACF. Dimana di tahap ini mencari model yang paling sesuai dengan datanya.

2. Pendugaan Parameter Model

Pendugaan parameter model digunakan untuk menentukan besaran koefisien model. Dimana metode yang biasa digunakan adalah dengan Metode *Maximum Likelihood Estimation* (MLE), *Durbin Watson* dll. Uji signifikansi parameter memiliki hipotesis sebagai berikut :

H_0 = Parameter signifikan

H_1 = Parameter tidak signifikan

Dimana dalam pengambilan keputusan H_0 ditolak jika nilai p-value < $\alpha = 0,05$ yang artinya model signifikan atau layak.

3. Tes Diagnostik

Tes diagnostik dilakukan untuk menyakinkan apakah spesifikasi modelnya telah sesuai. Jika nilai residualnya sudah white noise maka model sudah baik. Serta perlu dilakukannya uji asumsi untuk mengetahui kenormalan residual model.

4. Peramalan

Model peramalan dapat dikatakan baik jika hasil peramalan memiliki nilai ketepatan yang tinggi. Sehingga hasil peramalan tersebut dapat digunakan di kemudian hari.

2.7 Ketetapan Model Terbaik

Dalam suatu proses analisis runtun waktu, banyak model yang dapat mewakili keadaan data. Untuk menentukan model terbaik dapat digunakan perhitungan model residual yang sesuai berdasarkan kesalahan peramalan.

Kriteria yang digunakan yaitu menggunakan nilai *Akaike's Information Criterion* (AIC).

AIC digunakan untuk menentukan model yang optimum dari suatu data observasi. Dalam membandingkan dua buah regresi atau lebih, maka model yang mempunyai nilai AIC terkecil merupakan model yang lebih baik. Rumus untuk menentukan nilai AIC dinyatakan dengan persamaan:

$$AIC = n \log \left(\frac{RSS}{n} \right) + 2k \quad (2.37)$$

Dimana RSS adalah Residual Sum of Square (Jumlah Kuadrat Residual) yang dinyatakan dengan persamaan :

$$RSS = \sum_{i=1}^n \hat{e}_i^2 \quad (2.38)$$

2.8 Ketetapan Model Peramalan

Dalam memastikan bahwa suatu model yang dibangun dengan prosedur dan langkah yang benar akan cocok dengan data yang ada secara tepat. Salah satu kriteria untuk menilai ketetapan hasil peramalan yaitu menggunakan nilai *Symmetric Mean Absolute Percentage Error* (SMAPE)

SMAPE digunakan untuk mengukur kesalahan nilai dugaan model yang dinyatakan dalam bentuk persentase rata-rata *absolute* kesalahan. Rumus untuk menentukan nilai SMAPE dinyatakan dengan persamaan.

$$SMAPE = \left(\frac{1}{n} \sum_{i=1}^n \frac{|Z_t - \hat{Z}_t|}{\frac{|Z_t + \hat{Z}_t|}{2}} \right) \times 100 \quad (2.39)$$

Dengan

Z_t = data aktual output sample ke- t , $t = 1, 2, \dots, T$

\hat{Z}_t = data hasil ramalan out sample ke- t

2.9 Uji Normalitas

Dalam peramalan menggunakan metode ARIMA atau SARIMA perlu adanya uji normalitas. Uji ini digunakan untuk mengetahui apakah populasi data berdistribusi normal atau tidak. Analisis menggunakan metode parametric maka persyaratan normalitasnya harus terpenuhi. Uji normalitas dapat dilakukan dengan melihat grafik normalitas dimana data dikatakan berdistribusi normal jika reidu berada disekitar garis normal. Dapat juga menggunakan Uji *Jarque Bera*, dimana uji tersebut mempunyai hipotesis sebagai berikut :

H_0 = Residual model berdistribui normal

H_1 = Residual model tidak berdistribui normal

Kriteria penolakan :

H_0 ditolak jika p-value < α

H_0 diterima jika p-value > α

2.10 Jaringan Syaraf Tiruan

Menurut Sinaga (2012:2) Jaringan Syaraf Tiruan adalah salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran otak manusia tersebut. Jaringan syaraf tiruan (JST) atau yang biasa disebut *Artificial Neural Network* (ANN) atau *Neural Network* (NN) saja,

merupakan sistem pemrosesan informasi yang memiliki karakteristik mirip dengan jaringan syaraf pada makhluk hidup. *Neural network* berupa suatu model sederhana dari suatu syaraf nyata dalam otak manusia seperti suatu unit threshold yang biner. Jaringan Syaraf Tiruan tercipta sebagai suatu generalisasi model matematis dari pemahaman manusia (*human cognition*) yang didasarkan atas asumsi sebagai berikut.

1. Pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron.
2. Isyarat mengalir di antara sel syaraf atau neuron melalui penghubung.
3. Setiap sambungan penghubung memiliki bobot yang bersesuaian.
4. Setiap sel syaraf akan merupakan fungsi aktivasi terhadap isyarat hasil penjumlahan berbobot yang masuk kepadanya untuk menentukan isyarat keluarannya.

Suatu jaringan saraf tiruan memproses sejumlah besar informasi secara paralel dan terdistribusi, hal ini terinspirasi oleh model kerja otak biologis. Sistem syaraf buatan adalah suatu struktur pemroses informasi yang terdistribusi dan bekerja secara paralel, yang terdiri atas elemen pemroses (yang memiliki memori lokal dan beroperasi dengan informasi lokal) yang diinterkoneksi bersama dengan alur sinyal searah yang disebut koneksi. Setiap elemen pemroses memiliki 9 koneksi keluaran tunggal yang bercabang (*fan out*) ke sejumlah koneksi kolateral yang diinginkan (setiap koneksi membawa sinyal yang sama dari keluaran elemen pemroses tersebut). Keluaran dari elemen pemroses tersebut dapat merupakan sebarang jenis persamaan matematis yang diinginkan. Seluruh proses yang berlangsung pada setiap elemen pemroses harus benar-benar dilakukan secara

lokal, yaitu keluaran hanya bergantung pada nilai masukan pada saat itu yang diperoleh melalui koneksi dan nilai yang tersimpan dalam memori lokal (Pakaja, dkk, 2012: 23).

Menurut Buckley dkk (1995: 265) jaringan syaraf dan sistem *fuzzy* memiliki beberapa kesamaan. Jika sudah tidak ada model matematika dari masalah yang diberikan, maka keduanya dapat digunakan untuk memecahkan masalah (misalnya *pattern recognition*, *regression* atau *density estimation*). Jaringan syaraf hanya dapat ikut berperan jika masalah yang ada diungkapkan oleh contoh yang diamati (dengan jumlah yang cukup). Observasi ini digunakan untuk pelatihan secara *black box*. Di satu sisi tidak ada pengetahuan tentang masalah ini perlu diberikan. Di sisi lain, bagaimanapun, adalah tidak mudah untuk mengekstrak aturan yang mudah dipahami dari struktur jaringan syaraf tersebut. Sebaliknya, sistem kabur menuntut aturan linguistik sebagai pengganti contoh pembelajaran sebagai pengetahuan sebelumnya. Selanjutnya variabel input dan output harus dijelaskan secara linguistic atau bahasa (Nauck, et al., 1996: 295).

Jika pengetahuan tidak lengkap, salah atau bertentangan, maka sistem *fuzzy* harus disetel (*tuned*). Karena tidak ada pendekatan formal untuk itu, tuning dilakukan dengan cara heuristik. Hal ini biasanya sangat memakan waktu dan rawan 10 kesalahan.

Tabel 2.3 Perbedaan antara jaringan syaraf dan sistem *fuzzy*

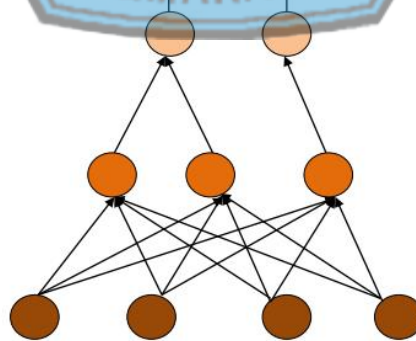
<i>Neural Network</i>	<i>Fuzzy System</i>
Tidak memerlukan model matematika	Memerlukan model matematika
Proses learning dari awal	Pegetahuan apriori merupakan hal yang penting
Terdapat beberapa algoritma pembelajaran	Tidak mampu untuk belajar
Perilaku black-box	Interpretasi dan implementasi sederhana

Antara jaringan syaraf dan jaringan syaraf *fuzzy system* memiliki banyak perbedaan diantaranya bobot koneksi, propagasi dan aktivasi. Meskipun memiliki banyak persamaan akan tetapi jaringan syaraf *fuzzy* memiliki karakteristik sebagai berikut :

1. Sebuah sistem *neuro-fuzzy* berbasis dasar sistem *fuzzy* dilatih dengan menggunakan metode pembelajaran berbasis-data yang berasal dari teori jaringan syaraf. Heuristik ini hanya memperhitungkan informasi lokal akun untuk menyebabkan perubahan lokal dalam sistem *fuzzy* mendasar.
2. Hal ini dapat direpresentasikan sebagai seperangkat aturan *fuzzy* setiap saat proses pembelajaran, yaitu, sebelum, selama dan sesudah. Dengan demikian sistem dapat diinisialisasi dengan atau tanpa pengetahuan sebelumnya dalam hal aturan *fuzzy*.

3. Prosedur pembelajaran dengan terpaksa untuk memastikan sifat semantik sistem *fuzzy* yang mendasarinya.
4. Sebuah sistem *neuro-fuzzy* mendekati n-dimensi suatu fungsi yang tidak diketahui yang sebagian diwakili oleh contoh-contoh pelatihan. Aturan *fuzzy* sehingga dapat diartikan sebagai prototipe yang jelas dari data pelatihan.
5. Sebuah sistem *neuro-fuzzy* direpresentasikan sebagai jaringan syaraf *feedforward* tiga lapis khusus seperti yang ditunjukkan pada Gambar 2.1.
 - a. Lapisan pertama sesuai dengan variabel input.
 - b. Lapisan kedua melambangkan aturan *fuzzy*.
 - c. Lapisan ketiga merupakan variabel output.
 - d. Set-set *fuzzy* dikonversi sebagai (*fuzzy*) bobot koneksi.

Beberapa pendekatan juga menggunakan lima lapisan dimana set-set *fuzzy* dikodekan dalam masing-masing unit lapisan kedua dan keempat. Namun, model ini dapat diubah ke dalam sebuah arsitektur tiga lapis.



Gambar 2.1 Arsitektur System Neuro *Fuzzy*

6. Pada dasarnya seseorang dapat membedakan antara tiga jenis jaringan syaraf *fuzzy*, yaitu, koperasi, bersamaan dan hibrida FNNs (Nauck, et al., 1997: 160).

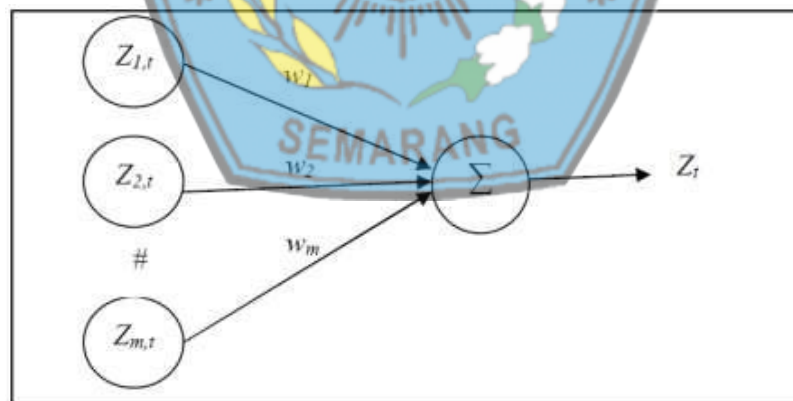
Neural network merupakan sebuah mesin pembelajaran yang dibangun dari sejumlah elemen pemrosesan sederhana yang disebut neuron atau node. Setiap neuron dihubungkan dengan neuron yang lain dengan hubungankomunikasi langsung melalui pola hubungan yang disebut arsitektur jaringan. Bobot-bobot pada koneksi mewakili besarnya informasi yang digunakan jaringan. Metode yang digunakan untuk menentukan bobot koneksi tersebut dinamakandengan algoritma pembelajaran. Setiap neuron mempunyai tingkat aktivasi yangmerupakan fungsi dari input yang masuk padanya (Warsito, 2009: 29).

Pada dasarnya seseorang dapat membedakan antara tiga jenis jaringan syaraf *fuzzy*, yaitu, koperasi, bersamaan dan hibrida FNNs (Nauck, et al., 1997: 160). (http://www.scholarpedia.org/article/Fuzzy_neural_network, 2014) *Neural network* merupakan sebuah mesin pembelajaran yang dibangun dari sejumlah elemen pemrosesan sederhana yang disebut *neuron* atau *node*. Setiap neuron dihubungkan dengan neuron yang lain dengan hubungankomunikasi 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 dinamakandengan algoritma pembelajaran. Setiap neuron mempunyai tingkat aktivasi yangmerupakan fungsi dari input yang masuk padanya (Warsito, 2009: 29).

Menurut Warsito (2009: 30) aktivasi yang dikirim suatu *neuron* ke *neuron* lain berupa sinyal dan hanya dapat mengirim sekali dalam satuwaktu, meskipun sinyal tersebut disebarakan pada beberapa *neuron* yang lain. Seperti Gambar 2.2 yaitu struktur jaringan syaraf tiruan, 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 Σ . Kemudian blok penjumlahan akan menjumlahkan semua inputterbobot dan menghasilkan sebuah nilai yaitu Z_t .

$$Z_t \text{ in} = \sum_i^m Z_{i,t} \cdot w_i = Z_{t,1} \cdot w_1 + Z_{t,1} \cdot w_2 + \dots + Z_{m,1} \cdot w_m \quad (2.40)$$

Aktifasi Z_t ditentukan oleh fungsi input jaringannya, $Z_t = f(Z_t \text{ in})$ dengan merupakan fungsi aktivasi yang digunakan.



Gambar 2.2 Struktur Jaringan Syaraf Tiruan

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

1. Tahap Pelatihan

Pada tahap pelatihan dimulai dengan memasukan pola-pola pelatihan kedalam 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.

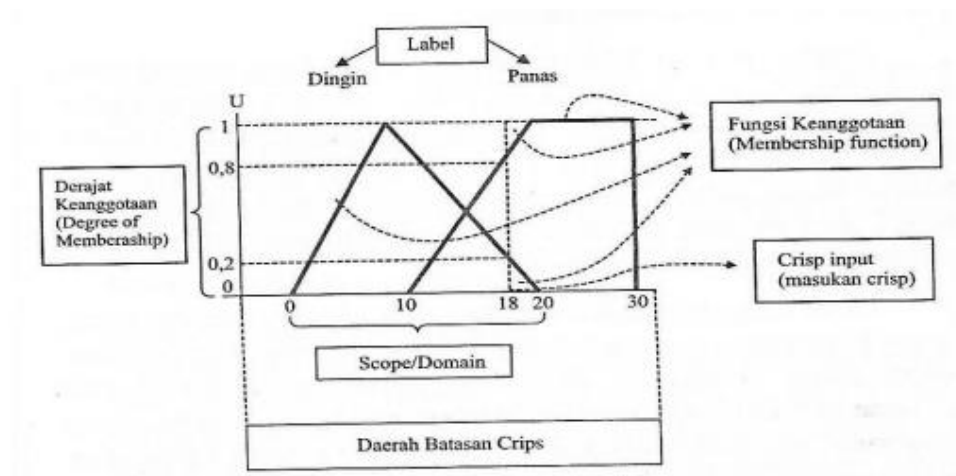
2. Tahap Pengujian

Ditahap 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.

2.11 Logika Fuzzy

Pada tahun 1965 Lotfi A. Zadeh dari University of California, Barkeley memperkenalkan konsep logika *fuzzy* (logika samar). Menurut teori logika *fuzzy* sistem *fuzzy* yaitu sistem yang dibangun dengan definisi, cara kerja, dan deskripsi.

Sistem *fuzzy* dapat dipahami dengan pengenalan konsep dasar yang berhubungan dengan logika *fuzzy* (Setiadji, 2009). Berikut adalah konsep dasar pada logika sistem *fuzzy* :



Gambar 2.3 Kerangka Sistem Fuzzy

Keterangan :

- Derajat keanggotaan : derajat dimana nilai crisp compatible dengan fungsi keanggotaan (dari 0 sampai 1), juga mengacu sebagai tingkat keanggotaan, nilai kebenaran atau masukan *fuzzy*.
- Label : nama deskriptif yang digunakan untuk mengidentifikasi suatu fungsi keanggotaan. Contoh dalam Gambar 2.3 pada temperatur terdapat lebel dingin dan panas.
- Fungsi keanggotaan : mendefinisikan himpunan *fuzzy* dengan memetakan masukan crisp dari domainnya ke derajat keanggotaan.
Masukan crisp : masukan yang tegas dan tertentu.
- Lingkup/domain : lebar daerah awal keanggotaan. Contoh dalam Gambar 2.3 pada temperatur terdapat domain dari 0 sampai 30
- Daerah batasan crisp : jangkauan seluruh nilai yang mungkin dapat diaplikasi pada sistem variabel.

2.11.1 Teori Himpunan *Fuzzy*

Berbeda dengan teori himpunan klasik yang menyatakan suatu objek adalah anggota (ditandai dengan angka 1) atau bukan anggota (ditandai dengan angka 0) dari suatu himpunan dengan batas keanggotaan yang jelas/tegas (*crisp*), teori himpunan *fuzzy* memungkinkan derajat keanggotaan suatu objek dalam himpunan untuk menyatakan peralihan keanggotaan secara bertahap dalam rentang antara 0 sampai 1 atau ditulis $[0,1]$ (Nasution, 2012: 4).

Jika memiliki nilai 1 maka suatu item tersebut menjadi anggota suatu himpunan tertentu sedangkan untuk 0 tidak menjadi anggota suatu himpunan tertentu. Misalkan E suatu himpunan sederhana dan A suatu himpunan bagianya $A \in E$. Pengertian keanggotaan ini dapat pula dinyatakan melalui konsep fungsi karakteristik μ_A , dimana harga $\mu_A(X)$ menyatakan apakah $x \in A$ atau $x \notin A$. (Setiadji, 2009)

$$\mu_A(X) = \begin{cases} 1, & \text{jika } x \in A \\ 0, & \text{jika } x \notin A \end{cases} \quad (2.41)$$

Menurut Kusumadewi, dkk (2006: 18) definisi himpunan *fuzzy* (*fuzzy set*) adalah sekumpulan obyek x dengan masing-masing obyek memiliki nilai keanggotaan (*membership function*) “ μ ” atau disebut juga dengan nilai kebenaran. Jika $Z_{i,t}$ adalah sekumpulan obyek, $Z_{i,t} = (Z_{1,t}, Z_{2,t}, \dots, Z_{m,t})$ dan anggotanya dinyatakan dengan Z maka himpunan *fuzzy* dari A di dalam Z adalah himpunan dengan sepasang anggota atau dapat dinyatakan sebagai berikut.

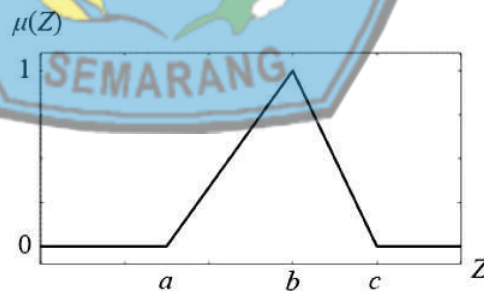
$$F = \{\{Z, \mu_F(Z)\} | Z \in Z_{i,t}\} \quad (2.42)$$

Dengan F adalah notasi himpunan *fuzzy*, $\mu_F(x)$ adalah derajat keanggotaan dari Z (nilai antara 0 sampai 1).

2.11.2 Fungsi Keanggotaan *Fuzzy*

Fungsi keanggotaan (*membership function*) adalah suatu fungsi yang menunjukkan pemetaan titik-titik *input* data ke dalam nilai keanggotaannya. Ada beberapa fungsi yang dapat digunakan melalui pendekatan fungsi untuk mendapatkan nilai keanggotaan, seperti *Triangular*, *Trapezoidal*, *Gaussian*, dan *Generalized Bell* (Widodo & Handayanto, 2012:61)

1. Fungsi Keanggotaan *Triangular*

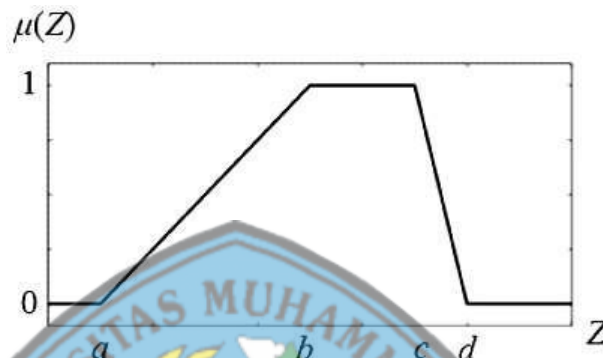


Gambar 2.4 Kurva Fungsi Keanggotaan *Triangular*

Fungsi keanggotaan *triangular* yang ditunjukkan pada Gambar 2.4 terbentuk oleh tiga parameter: a , b dan c sebagai berikut.

$$\mu(Z) = \begin{cases} 0; Z \leq a \text{ atau } Z \geq c \\ \frac{(z-a)}{(b-a)}; a \leq Z \leq b \\ \frac{(b-Z)}{(c-b)}; b \leq Z \leq c \end{cases} \quad (2.43)$$

2. Fungsi keanggotaan *Trapezoidal*

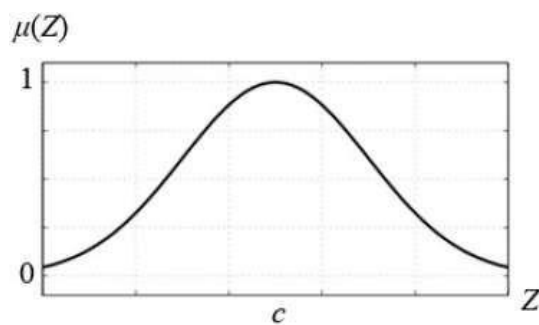


Gambar 2.5 Kurva Fungsi Keanggotaan *Trapezoidal*

Fungsi keanggotaan *trapezoidal* yang ditunjukkan pada Gambar 2.5 terbentuk oleh empat parameter: a , b , c , dan d , sebagai berikut.

$$\mu(Z) = \begin{cases} 0; Z \leq a \text{ atau } Z \geq d \\ \frac{(z-a)}{(b-a)}; a \leq Z \leq b \\ 1; b \leq Z \leq c \\ \frac{(b-Z)}{(c-b)}; c \leq Z \leq d \end{cases} \quad (2.44)$$

3. Fungsi Keanggotaan *Gaussian*

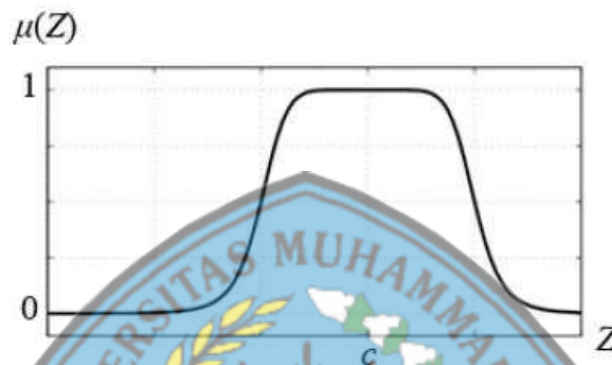


Gambar 2.6 Kurva Fungsi Keanggotaan *Gaussian*

Fungsi keanggotaan *Gaussian* yang ditunjukkan pada Gambar 2.6 terbentuk oleh dua parameter: σ dan c , sebagai berikut.

$$\mu(Z) = e^{-\frac{1}{2}\left(\frac{z-c}{\sigma}\right)^2} \quad (2.45)$$

4. Fungsi Keanggotaan *Generalized Bell*



Gambar 2.7 Kurva Fungsi Keanggotaan *Generalized Bell*

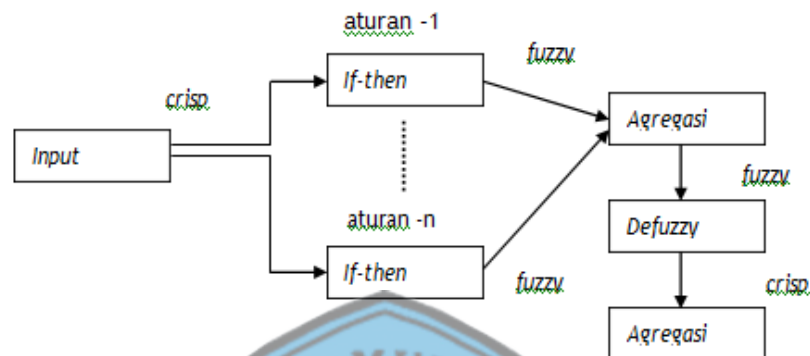
Fungsi keanggotaan *generalized bell* yang ditunjukkan pada Gambar 2.7 terbentuk oleh tiga parameter: a , b , dan c , sebagai berikut.

$$\mu(Z) = \frac{1}{1 + \left|\frac{z-c}{\sigma}\right|^{2b}} \quad (2.46)$$

2.11.3 Sistem Inferensi *Fuzzy* (*Fuzzy Inference System*)

Sistem Sistem Inferensi *Fuzzy* (*Fuzzy Inference System* atau FIS) merupakan suatu kerangka komputasi yang didasarkan pada teori himpunan *fuzzy*, aturan *fuzzy* berbentuk *if-then*, dan penalaran *fuzzy*. Sistem inferensi *fuzzy* dijelaskan pada Gambar 2.8. FIS menerima *inputcrisp*. *Input* ini kemudian dikirim ke basis pengetahuan yang berisi n aturan *fuzzy* dalam bentuk *if-then*. *Fire strength* (bobot) akan dicari pada setiap aturan. Apabila jumlah aturan lebih dari

satu, maka akan dilakukan agregasi dari semua aturan. Selanjutnya, pada hasil agregasi akan dilakukan *defuzzy* untuk mendapatkan nilai *crisp* sebagai keluaran sistem (Kusumadewi, dkk, 2006: 27).



Gambar 2.8. Diagram Blok Sistem Inferensi Fuzzy

Menurut Jang (1993: 665) sistem inferensi *fuzzy* terdiri dari 5 (lima) bagian sebagai berikut.

1. Basis aturan (*rule base*), terdiri dari sejumlah aturan *fuzzy if-then*.
2. Basis data (*database*) yang mendefinisikan fungsi keanggotaan dari himpunan *fuzzy* yang digunakan dalam aturan *fuzzy*, biasanya basis
3. aturan dan basis data digabung dan disebut basis pengetahuan (*knowledge base*).
4. Satuan pengambilan keputusan (*decision making unit*) yang membentuk operasi inferensi pada aturan (*rule*).
5. Antar muka fuzzifikasi (*fuzzification interface*) yang mengubah input ke dalam derajat yang sesuai dengan nilai linguistik (*linguistik value*).
6. Antar muka defuzzifikasi (*defuzzification interface*) yang mengubah hasil *fuzzy* inferensi ke bentuk output yang kompak.

2.11.4 Fuzzy C-Means (FCM)

Fuzzy C-Means (FCM) adalah suatu teknik pengklasteran data yang mana keberadaan tiap data dalam suatu *cluster* ditentukan oleh nilai keanggotaan. Konsep FCM pertama kali adalah menentukan pusat *cluster* yang akan menandai lokasi rata-rata untuk tiap *cluster*. Pada kondisi awal pusat *cluster* ini masih belum akurat. Tiap-tiap data memiliki derajat keanggotaan untuk 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 (Kusumadewi, dkk, 2006: 23).

Algoritma *Fuzzy C-Means* diberikan sebagai berikut.

- 1) Tentukan:
 - a. matriks Z berukuran $n \times m$, dengan n = jumlah data yang akan diklaster dan m = jumlah variabel (kriteria),
 - b. jumlah *cluster* yang dibentuk = $C (\geq 2)$,
 - c. pangkat (pembobot) = $w (> 1)$,
 - d. maksimum iterasi,
 - e. kriteria penghentian = ξ (nilai positif yang sangat kecil),
 - f. iterasi awal, $t = 1$ dan $\Delta = 1$.
- 2) Bentuk matriks partisi awal U^0 adalah sebagai berikut.

$$U = \begin{bmatrix} \mu_{11}(Z_{1,t}) & \mu_{12}(Z_{2,t}) & \dots & \mu_{1n}(Z_{n,t}) \\ \mu_{21}(Z_{1,t}) & \mu_{22}(Z_{2,t}) & \dots & \mu_{2n}(Z_{n,t}) \\ \vdots & \vdots & & \vdots \\ \mu_{c1}(Z_{1,t}) & \mu_{c2}(Z_{c,t}) & \dots & \mu_{cn}(Z_{n,t}) \end{bmatrix} \quad (2.47)$$

(matrik partisi awal biasanya dipilih secara acak)

3) Hitung pusat *cluster* V untuk setiap *cluster*.

$$v_{ij} = \frac{\sum_{k=1}^n (\mu_{ij})^w \cdot Z_{kj}}{\sum_{k=1}^n (\mu_{ij})^w} \quad (2.48)$$

4) Perbaiki derajat keanggotaan setiap data pada setiap *cluster* (perbaiki matrik partisi) sebagai berikut

$$\mu_{ik} = \left[\sum_{j=1}^c \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(w-1)} \right]^{-1} \quad (2.49)$$

dengan

$$d(Z_{k,t} - v_i) = \left[\sum_{j=1}^m (Z_{k,t} - v_i) \right]^{1/2} \quad (2.50)$$

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

$$\Delta = \|U^t - U^{t-1}\| \quad (2.51)$$

Apabila $\Delta \leq \xi$ maka iterasi dihentikan, namun apabila $\Delta > \xi$ maka iterasi ($t=t+1$) dan kemudian ke langkah 3.

2.12 Adaptive Neuro Fuzzy Inference System (ANFIS)

Model *fuzzy* dapat digunakan sebagai pengganti dari banyak lapisan. Dalam hal ini sistem dapat dibagi menjadi dua grup, yaitu satu grup berupa jaringan syaraf dengan bobot-bobot *fuzzy* dan fungsi aktivasi *fuzzy*, dan grup kedua berupa jaringan syaraf dengan input yang di-*fuzzy*-kan pada lapisan pertama atau kedua, namun bobot-bobot pada jaringan syaraf tersebut tidak di-*fuzzy*-kan. Menurut Kusumadewi dkk (2006: 23), *Neuro Fuzzy* termasuk kelompok kedua.

ANFIS (*Adaptive Neuro Fuzzy Inference System* atau *Adaptive Networkbased Fuzzy Inference System*) adalah arsitektur yang secara fungsional samadengan *fuzzy rule base* model Sugeno. Arsitektur ANFIS juga sama dengan jaringan syaraf dengan fungsi radial dengan sedikit batasan tertentu. Bisa dikatakan bahwa ANFIS adalah suatu metode yang mana dalam 23 melakukan penyetelan aturan digunakan algoritma pembelajaran terhadap sekumpulan data. Pada ANFIS juga memungkinkan aturan-aturan untuk beradaptasi (Kusumadewi, dkk, 2006: 42).

Agar jaringan dengan fungsi basis radial ekuivalen dengan *fuzzy* berbasis aturan model Sugeno orde 1 ini, diperlukan batasan sebagai berikut.

- a. Keduanya harus memiliki metode *agregasi* yang sama (rata-rata terbobot atau penjumlahan terbobot) untuk menurunkan semua *output*.
- b. Jumlah fungsi aktivasi harus sama dengan jumlah aturan *fuzzy* (*if-then*).
- c. Jika ada beberapa input pada basis aturannya, maka tiap fungsi aktivasi harus sama dengan fungsi keanggotaan tiap-tiap inputnya.
- d. Fungsi aktivasi dan aturan-aturan *fuzzy* harus memiliki fungsi yang sama untuk *neuron-neuron* dan aturan-aturan yang ada di sisi *output*.

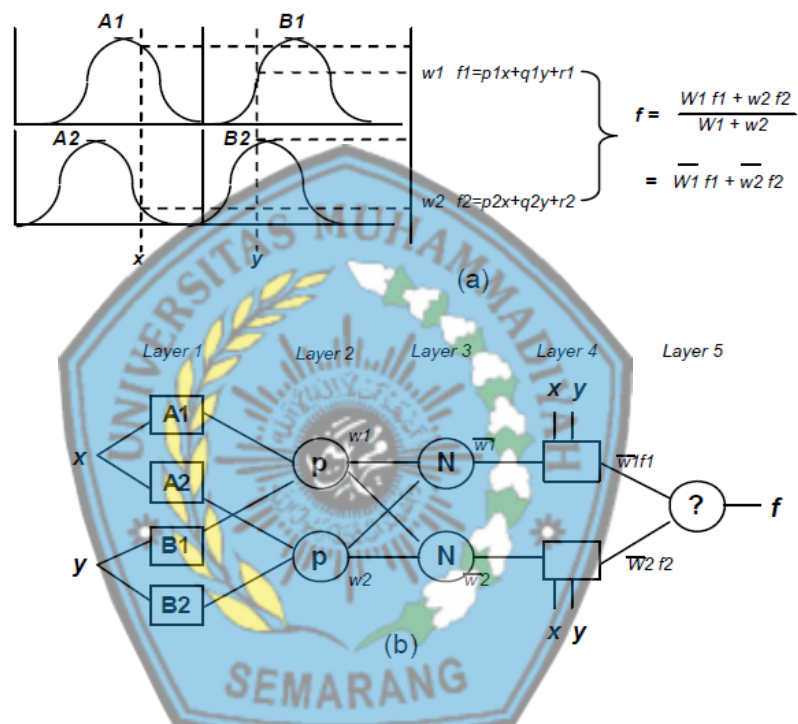
2.12.1 Arsitektur Jaringan ANFIS

ANFIS dikembangkan oleh J.S.R Jang pada tahun 1992. Menurut Jang kelas adaptive network secara fungsional ekuivalen dengan *fuzzy inference system*. ANFIS adalah arsitektur yang secara fungsional sama dengan *fuzzy rule base* model Sugeno orde satu (Jang

et al. 1997). Jika diasumsikan *fuzzy inference system* mempunyai dua input x dan y serta mempunyai satu output z , maka menurut model Sugeno orde satu, ada dua aturan sebagai berikut :

Rule 1 : If x is A_1 and y is B_1 , then $f_1 = p_1x + q_1y + r$

Rule 2 : If x is A_2 and y is B_2 , then $f_2 = p_2x + q_2y + r$

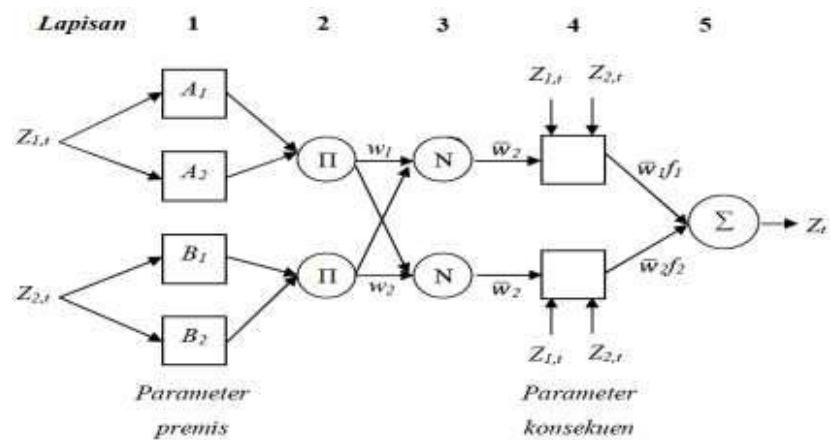


Gambar 2.9 (a) Model Sugeno orde-1 dengan 2 input dan 2 rule ;

(b) Arsitektur ANFIS (J.S.R.Jang)

ANFIS (*Adaptif Neuro Fuzzy Inference System*) adalah metode jaringan neural yang fungsinya sama dengan sistem inferensi *fuzzy*. Pada ANFIS, proses belajar pada jaringan *neural* dengan sejumlah pasangan data berguna untuk memperbaharui parameter-parameter sistem inferensi *fuzzy* (Fariza, dkk, 2007: 77). Jaringan ANFIS yang ditunjukkan pada Gambar 2.10 terdiri dari lapisan- lapisan sebagai

berikut (Jang, Sun & Mizutani, 1997: 70).



Gambar 2.10 Arsitektur Jaringan ANFIS

Lapisan 1 :

Lapisan ini merupakan lapisan *fuzzifikasi*. Pada lapisan ini tiap neuron adaptif terhadap parameter suatu aktivasi. Output dari tiap neuron berupa derajat keanggotaan yang diberikan oleh fungsi keanggotaan input. Misalkan fungsi keanggotaan *Generalized Bell* diberikan sebagai.

$$\mu(Z) = \frac{1}{1 + \left| \frac{Z-c}{a} \right|^{2b}} \quad (2.52)$$

Dengan Z adalah input, dalam hal ini $Z = \{Z_{1,t}, Z_{2,t}\}$ dan $\{a, b, \text{ dan } c\}$ adalah parameter-parameter, biasanya $b = 1$. Jika nilai parameter-parameter ini berubah, maka bentuk kurva yang terjadi akan ikut berubah. Parameter-parameter ini biasanya disebut dengan nama parameter premis.

Lapisan 2:

Lapisan ini berupa neuron tetap (diberi simbol Π) merupakan

hasil kali dari semua masukan, sebagai berikut.

$$w_i = \mu_{Ai} \cdot \mu_{Bi} \quad (2.53)$$

Biasanya digunakan operator *AND*. Hasil perhitungan ini disebut *firing strength* dari sebuah aturan. Tiap neuron merepresentasikan aturan ke-*i*.

Lapisan 3:

Tiap neuron pada lapisan ini berupa neuron tetap (diberi simbol *N*) merupakan hasil perhitungan rasio dari *firing strength* ke-*i* (w_i) terhadap jumlah dari keseluruhan *firing strength* pada lapisan kedua, sebagai berikut:

$$\bar{w}_i = \frac{w_1}{w_1 + w_2}; i = 1, 2 \quad (2.54)$$

Hasil perhitungan ini disebut *normalized firing strength*.

Lapisan 4:

Lapisan ini berupa neuron yang merupakan neuron adaptif terhadap suatu output, sebagai berikut.

$$\bar{w}_i f_i = \bar{w}_i (p_i Z_{1,t} + q_i Z_{2,t} + r_i) \quad (2.55)$$

dengan adalah *normalized firing strength* pada lapisan ketiga dan p_i, q_i dan r_i adalah parameter-parameter pada neuron tersebut. Parameter-parameter ini biasa disebut parameter konsekuen.

Lapisan 5:

Lapisan ini berupa neuron tunggal (diberi symbol Σ) merupakan hasil penjumlahan seluruh output dari lapisan keempat,

sebagai berikut.

$$\sum_i \bar{w}_i f_i = \frac{\sum w_i f_i}{\sum w_i} \quad (2.56)$$

2.12.2 Algoritma Pembelajaran *Hybrid*

Pada saat parameter *premis* ditemukan keluaran keseluruhan akan merupakan kombinasi linier dari konsekuensi parameter, yaitu:

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \quad (2.57) \\ &= \bar{w}_1 (p_1 Z_{1,t} + q_1 Z_{2,t} + r_1) + \bar{w}_2 (p_2 Z_{1,t} + q_2 Z_{2,t} + r_2) \\ &= (\bar{w}_1 Z_{1,t}) p_1 + (\bar{w}_1 Z_{2,t}) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 Z_{1,t}) p_2 + (\bar{w}_2 Z_{2,t}) q_2 \\ &\quad + (\bar{w}_2) r_2 \end{aligned}$$

Adalah linier terhadap parameter p_1, q_1, r_1, p_2, q_2 dan r_2 . Algoritma hibrida akan mengatur parameter-parameter p_i, q_i dan r_i konsekuensi dan secara maju (*forward*) dan akan mengatur parameter-parameter premis a, b , dan c secara mundur (*backward*). Pada langkah maju, input jaringan akan merambat maju sampai pada lapisan keempat. Parameter-parameter konsekuensi akan diidentifikasi dengan menggunakan *least-square*. Sedangkan pada langkah mundur, eror sinyal akan merambat mundur dan parameter-parameter premis akan diperbaiki dengan menggunakan metode *gradient descent*.

Prosedur pembelajaran *Hybrid* metode ANFIS dapat dilihat pada Tabel 2.4 (Jang, Sun & Mizutani 1997: 78).

Tabel 2.4 Prosedur Pembelajaran *Hybrid* Metode ANFIS

	Arah Maju	Arah Mundur
Parameter Premis	Tetap	Gradient descent
Parameter Konsekuen	Least-squares estimator	Tetap
Sinyal	Keluaran neuron	Sinyal eror

2.13 *Hybrid* SARIMA-ANFIS

Hybrid adalah kombinasi dari dua atau lebih system dalam satu fungsi, dimana dalam hal ini adalah kombinasi antara SARIMA dengan ANFIS. Sudah banyak peneliti yang menggunakan metode *Hybrid* dikarenakan hasil yang diharapkan dapat saling melengkapi karena pada dasarnya dalam dunia nyata jarang ditemukan kejadian deret waktu yang murni linier ataupun murni non-linier (Zheng & Zhong, 2011). Secara umum kombinasi deret waktu dituliskan sebagai berikut :

$$y_t = L_t + E_t \quad (2.58)$$

Dimana L_t adalah komponen linier dan E_t adalah komponen non-linier . terdapat dua komponen yang harus diestimasi dari data, yaitu model SARIMA digunakan untuk menyelesaikan kasus linier, dimana residual dari model linier yang masih mengandung informasi hubungan non-linier. Dapat dituliskan sebagai berikut :

$$e_t = y_t + \hat{L}_t \quad (2.59)$$

Dimana \hat{L}_t adalah hasil nilai ramalan SARIMA pada waktu t . persamaan nilai residual untuk model ANFIS apat dituliskan sebagai berikut:

$$e_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \varepsilon_t \quad (2.60)$$

Diamana f adalah fungsi non linier yang dijelaskan oleh ANFIS dan ε_t adalah *error* yang acak, sehingga fungsi kombinasi untuk memperdiksi adalah sebagai berikut :

$$\hat{y}_t = \hat{L}_t + \hat{E}_t \quad (2.61)$$

2.14 Curah Hujan

Curah hujan adalah butir-butir air atau kristal es yang jatuh/keluar dari awan atau kelompok awan. Jika curahan dimaksud dapat mencapai permukaan bumi disebut sebagai hujan, jika setelah keluar dari dasar awan tetapi tidak jatuh sampai ke permukaan bumi disebut *virga*. Butiran air yang dapat keluar dari permukaan bumi harus memiliki garis tengah ≥ 200 mikrometer (1 mikrometer sebanding dengan 0,001 cm), kurang dari ukuran diameter tersebut butir-butir air dimaksud habis menguap di atmosfer sebelum mampu mencapai permukaan bumi. Banyaknya curah hujan yang mencapai permukaan bumi atau tanah selama selang waktu tertentu dapat diukur dengan jalan mengukur tinggi air hujan dengan cara tertentu. Hasil dari pengukuran dinamakan curah hujan, yaitu tanpa mengingat macam atau bentuk pada saat mencapai permukaan bumi dan tidak memperhitungkan endapan yang meresap ke dalam tanah, hilang karena penguapan, ataupun mengalir (Swarinoto & Sugiyono, 2011).

Menurut BMKG **Curah Hujan (mm)** adalah ketinggian air hujan yang terkumpul dalam penakar hujan pada tempat yang datar, tidak menyerap, tidak meresap dan tidak mengalir. Unsur hujan 1 (satu) milimeter artinya dalam luasan satu meter persegi pada tempat yang datar tertampung air hujan setinggi satu milimeter atau tertampung air hujan sebanyak satu liter.

Zona Musim (ZOM) adalah daerah yang pola hujan rata-ratanya memiliki perbedaan yang jelas antara periode musim kemarau dan periode musim hujan. Wilayah ZOM tidak selalu sama dengan luas daerah administrasi pemerintahan. Dengan demikian satu kabupaten/ kota dapat saja terdiri dari beberapa ZOM dan sebaliknya satu ZOM dapat terdiri dari beberapa kabupaten.

Dasarian adalah rentang waktu selama 10 (sepuluh) hari. Dalam satu bulan dibagi menjadi 3 (tiga) dasarian, yaitu :

- a. Dasarian I : tanggal 1 sampai dengan 10
- b. Dasarian II : tanggal 11 sampai dengan 20
- c. Dasarian III : tanggal 21 sampai dengan akhir bulan

Sifat Hujan adalah perbandingan antara jumlah curah hujan selama rentang waktu yang ditetapkan (satu periode musim hujan atau satu periode musim kemarau) dengan jumlah curah hujan normalnya (rata-rata selama 30 tahun periode 1981 - 2010). Sifat hujan dibagi menjadi 3 (tiga) kategori, yaitu :

- a. Di Atas Normal (AN), jika nilai curah hujan lebih dari 115% terhadap rata-ratanya
- b. Normal (N), jika nilai curah hujan antara 85% - 115% terhadap rata-ratanya

- c. Di Bawah Normal (BN), jika nilai curah hujan kurang dari 85% terhadap rata-ratanya

